D-GAN: DIVERGENT GENERATIVE ADVERSARIAL APPROACH FOR POSITIVE-UNLABELED LEARNING AND COUNTER-EXAMPLES GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Positive Unlabeled learning task remains an interesting challenge in the context of image analysis. Recent approaches suggest to exploit the GANs (Goodfellow et al., 2014) abilities to answer this problem. In this paper, we propose a new approach named Divergent-GAN (D-GAN). It keeps the light adversarial architecture of the PGAN method, with a better robustness counter the varying images complexity, while simultaneously allowing the same functionalities as the GenPU method, like the generation of relevant counter-examples. However, this is achieved without the need of prior knowledge, nor an onerous architecture and framework. Its functionning is based on the combination between the behaviour principles of Positive Unlabeled learning classification and the adversarial GAN training. Experimental results show that this divergent adversarial framework outperforms the state of the art PU learning in terms of prediction accuracy, training robustness, and its ability to work on both simple and complex real images. Combined with an additional generator, the proposed approach even allows to accomplish noisy labeled learning, and thus opening new application perspectives for GANs architectures.

1 INTRODUCTION

Nowadays, the number of available labeled datasets dedicated to perception application with learning methods has considerably augmented (Russakovsky et al., 2015), (Yu et al., 2015), (Cordts et al., 2016). However, when these methods trained on labeled datasets are applied to real application data, the generalization problem can be intensified. Indeed, the prediction performances of these methods deteriorate when the real data on which they are applied do not follow the same properties as the training dataset. Consequently, the need to use a labeled dataset specialized for the given target application becomes a necessity. Furthermore, another issue susceptible to happen in the context of a real application is the risk to meet samples which does not match any training class. This can be associated to the novelty detection problem. Nonetheless, for such real applications, it can also be easy to get unlabeled data from the real context application. If we consider these unlabeled data as counter-examples of our classes of interest, then this allows a better adaptability of these learning methods to real world applications, without the need to label additional data. Such difficulty can be solved with Positive Unlabeled (PU) learning techniques. (Sansone et al., 2018) enumerates additional application examples which can also be adressed in this way.

The PU learning task consists in learning to distinguish positive samples, considered as the samples of our class of interest, from negative samples considered as the counter-examples of our class of interest, by using only positive and unlabeled samples during the training phase. The difficulty resides in the fact that unlabeled samples can be composed by a fraction π_p of unlabeled positive samples. Several PU learning methods exist, and some of them are adapted for image classification. They can be distinguished in two groups. On the one hand, one-stage methods have been proposed, such as the unbiased PU method (uPU) (Du Plessis et al., 2015), or the non-negative PU method (nnPU) (Kiryo et al., 2017). Both allow good prediction preformances but they require prior knowledge on the training dataset. After all, these one-stage approaches have the advantage to need only one training of the classifier, and they can be combined with some approaches (Jain et al., 2016) which suggest to estimate prior knowledge. On the other hand, two-stages methods include Rank Pruning

method (RP) (Northcutt et al., 2017), which consists in keeping only the samples considered during the training as the most negative among the unlabeled ones. Several consecutive trainings of a binary classifier have to be realized. After each training, only the samples considered as the most negative are kept in order to substitute the unlabeled samples for the next training of the same classifier. RP achieves good prediction performances, even without the need of prior knowledge in opposition to uPU and nnPU. Another advantage of RP is that it can even be applied on noisy labeled datasets, which can be considered as a PU dataset with some mislabeled positive samples included in the positive labeled set. However this second type of methods needs a higher training computational cost and seems limited in their prediction performances on complex RGB image datasets like CIFAR-10.

Moreover, a new subcategory of two-stage methods with benefits of the generative models abilities have been recently proposed, named Positive-GAN (Chiaroni et al., 2018) and GenPU (Hou et al., 2018). They demonstrate some comparable or better prediction results as the previously cited methods. In this case, generated samples replace the unlabeled samples by learning on these latter, and enventually on the positive labeled during the first step. But except for the fact that these two approaches exploits the GANs (Goodfellow et al., 2014) benefits, they are not suitable in the same conditions. On the one hand, GenPU seems the most interesting PU method on simple datasets when we own very few positive labeled samples. However, as nnPU and uPU it still needs prior knowledge, while its training optimization functions need to many hyper-parameters. In addition, its onerous architecture composed of 5 learning models instead of two for the PGAN, entails a high computational cost. But despite these constraints, it allows to generate relevant counter-examples from a PU dataset, which is a very interesting functionality. On the other hand, the Positive-GAN also named PGAN presents a relatively steadier behaviour than RP and better results on the complex RGB image dataset CIFAR-10, in terms of prediction performances, and without prior knowledge for varying π_P fractions. However, it appears to be less suited for relatively simpler datasets such as MNIST. Indeed, the GAN generated samples are considered as negative samples from the discriminator point of view, but this is true only if the generated samples distribution does not match with the real unlabeled samples distribution, while converging close enough to it. In other words, if the GAN works too well, then the PGAN will fall back to the initial PU learning problem. However, it provides a practical framework and it is suited for complex datasets, contrary to previous methods. As far as the classification step is concerned, it only needs a traditional GAN model, such as (Goodfellow et al., 2014) or (Arjovsky et al., 2017), composed of one generator and one discriminator.

Also, we want to emphasize that a PU method which does not need prior knowledge to be effective could potentially be extended to additional various learning real application challenges such as incremental learning (Craye et al., 2018) or (Hadsell et al., 2009). Indeed, in this context the fraction π_P of unlabeled positive samples can vary continuously between each new incremental unlabeled minibatch of samples to process. However, to the best of our knowledge, existing PU learning methods which need prior knowledge implicitly assume that this prior knowledge holds true for each backpropagation iteration. For this reason, we will detail and show experimentally the interest to use a normalization technique which avoids unexpected dependencies between distinct samples of a same minibatch, contrary to Batch-Normalization (Ioffe & Szegedy, 2015).

Table 1: Highlighting of our contribution compared to previous state of the art methods.

Methods	D-GAN (proposed)	PGAN	GenPU	RP	nnPU	uPU
No need of priori knowledge	✓	✓	×	✓	×	×
Practical PU framework	✓	✓	×	✓	✓	✓
Simple PU dataset analysis	✓	×	✓	✓	✓	✓
Complex PU dataset analysis	✓	✓	×	×		
Generation of relevant counter-examples	✓	×	✓			
Noisy Labeled learning	✓	×	×	✓	×	×

As illustrated in Table 1, the proposed Divergent-GAN approach has a new framework which combines the benefits of previous methods. More specifically, it exploits both principles of PU and adversarial training from the discriminator point of view. In this way, the D-GAN gets the GenPU ability to generate relevant counter-examples, but without its onerous framework. Simultaneously, it keeps the light architecture of the PGAN and converges towards its interesting behaviour in the worst case of complex image datasets. Prior knowledge is not needed to outperform the PGAN and RP methods on MNIST and CIFAR-10 datasets in a One vs. Rest mode, nor to be equivalent to the GenPU method on the simple MNIST dataset One vs. One mode with few positive labeled samples. In addition, we discuss the possible extension of this model to the noisy labeled learning task, also

referred to as noisy PU learning, and illustrate it with some qualitative results on a simple noisy labeled dataset. Moreover, we show the D-GAN ability to generate relevant counter-examples on the RBG image dataset celebA converted in a PU dataset. [1] to allow the reproducibility of the method on 64*64*3 image RGB format.

Section 2 describes the proposed approach and its functionning behaviour. Section 3 shows experimental results which demonstrate its abilities from a qualtitative point of view and its learning stability and prediction performances for the PU learning task compared to other PU learning methods. Then, we finish by a conclusion where we also discuss perspectives.

2 METHOD

The proposed approach consists in combining the benefits of the traditional PU learning methods for simple problems with the Positive-GAN performance robustness for complex images. In addition, the Divergent-GAN can enjoy the interesting GenPU functionalities, like the generation of relevant counter examples, but without the need of prior knowledge or additional hyper-parameters. Moreover, this is achieved while keeping a lighter architecture to allow a more practical framework. On another note, the presented reasoning can be extended to noisy labeled learning tasks. In other words, the Divergent-GAN can even work with a noisy Positive Unlabeled dataset. Moreover, some regularization methods like Batch-Normalization, often used to improve GANs training, are reconsidered in the context of this untraditional GAN framework.

2.1 A DIVERGENT APPROACH FROM THE DISCRIMINATOR POINT OF VIEW

Firstly, we remember that the Positive Unlabeled (PU) learning problem consists in trying to distinguish positive samples from negative samples by only owning a PU dataset. We start by detailing the expected PU functionality incorporated in the GAN discriminator. Let $X \in \mathbb{R}^m$ be the input random variable and $y \in \{0, 1\}$ its associated label. X can be a positive x_P, negative x_N or unlabeled x_U sample which respectively follow the distributions $p_P = p(X|y = 0)$, $p_N = p(X|y = 1)$ and $p_U = (1 - \pi_P) \cdot p_N + \pi_P \cdot p_P$, with $\pi_P \in (0, 1)$ the unknown prior which represents the fraction of unlabeled positive samples included in the unlabeled dataset. Let $D : \mathbb{R}^m \to (0, 1)$ be the decision function which is then considered as the discriminator network of our generative model. We have $l(\hat{y}, y)$ such that $l : \mathbb{R} \times \{0, 1\} \to \mathbb{R}$ is the cost function with the predicted output $\hat{y}$ of D for a given sample X and the corresponding label y as inputs. Given the composition of the unlabeled dataset, we develop the following Positive Unlabeled risk $R_{PU}(D)$ such that:

$$\begin{aligned} R_{PU}(D) &= \mathbb{E}_{X\sim p_U}[l(D(X), y = 1)] + \mathbb{E}_{X\sim p_P}[l(D(X), y = 0)] \\ &= (1 - \pi_P) \cdot \mathbb{E}_{X\sim p_N}[l(D(X), y = 1)] + \pi_P \cdot \mathbb{E}_{X\sim p_P}[l(D(X), y = 1)] \\ &+ \mathbb{E}_{X\sim p_P}[l(D(X), y = 0)]. \end{aligned} \tag{1}$$

If we suppose that D does not overfit labeled and unlabeled positive samples, we can deduce that:

$$R_{PU}(D) = R_{PN}(D, \delta), \tag{2}$$

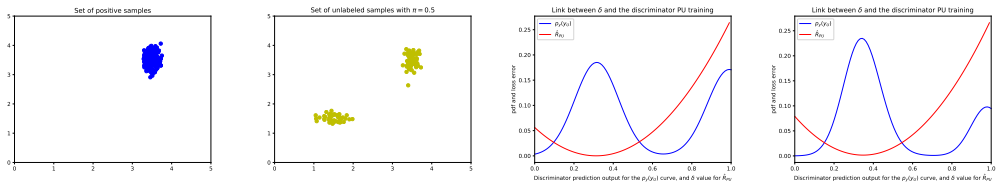
with $R_{PN}(D, \delta)$ the risk of a traditional Positive Negative training influenced by the parameter δ such that:

$$R_{PN}(D, \delta) = (1 - \pi_P) \cdot \mathbb{E}_{X\sim p_N}[l(D(X), y = 1)] + \mathbb{E}_{X\sim p_P}[l(D(X), \delta)], \tag{3}$$

$$\text{with } \mathbb{E}_{X\sim p_P}[l(D(X), \delta)] \Leftrightarrow \pi_P \cdot \mathbb{E}_{X\sim p_P}[l(D(X), y = 1)] + \mathbb{E}_{X\sim p_P}[l(D(X), y = 0)]. \tag{4}$$

The parameter $\delta \in (0, 1)$ represents the mean distribution of the probability density function of the output values predicted by the discriminator D for the positive samples x_P. Therefore, D will output a prediction value $\hat{y}$ close to 0 when the input is a negative sample x_N, and close to δ when the input is a positive sample x_P. Figure 1 demonstrates experimentally this phenomenon when we train with the mean squared error (MSE) loss function a multi-layer perceptron to distinguish a 2D gaussian distribution to another one by only owning a PU dataset. We can observe that the global minimum of the approximated risk $\hat{R}_{PN}(D, \delta)$ corresponds to the global maximum of the distribution of

[1] The corresponding code of this latter experiment will be publicly available (once the anonymous reviewing period will end)



(a) (b) (c) (d)

Figure 1: Link between the equation loss suggested with δ and the distribution of the discriminator output predictions for an unlabeled minibatch. (a) Shows a set of 2D points considered as positive samples. (b) Shows a set of 2D points considered as unlabeled samples. For (c), the loss function l is the MSE metric. Both curves have been normalized to allow a better visualization. $p_Y(\hat{y}_U)$ represents the probability distribution of the predicted output $\hat{y}$ for a minibatch of unlabeled samples, with $\pi_P = 0.5$. $\hat{R}_{PN}(D, \delta)$ represents the risk approximated in function of δ with the equation 3 on a minibatch of positive and negative labeled samples. (d) Same curves as in (c) but by giving in input a concatenation of an unlabeled minibatch with a positive labeled minibatch. We can observe that unlabeled and labeled positive samples provide a unified prediction output distribution.

positive unlabeled samples, which confirms experimentally the equality exposed above. Moreover the distribution of the negative samples is centered on the label associated to the unlabeled samples during the training, which coincides also with the equation formulation 3.

Next, let us consider the decision function as the discriminator D of a generative adversarial network. We propose to incorporate the risk R_{PU} into the minimax value function of the original GAN model (Goodfellow et al., 2014) by adding the loss function term $log[1 - D(x_P)]$, such that we obtain the following divergent value function V_D for the proposed approach:

$$\begin{aligned} \min_G \max_D V_D(D, G) = &\mathbb{E}_{x_U \sim p_U(x_U)}[logD(x_U)] + \mathbb{E}_{z \sim p_z(z)}[log(1 - D(G(z)))] \\ &+ \mathbb{E}_{x \sim p_P(x_P)}[log(1 - D(x_P))], \end{aligned} \tag{5}$$

with

$$\begin{aligned} log[1 - D(x_P)] &= y_P \cdot log[D(x_P)] + (1 - y_P) \cdot log[1 - D(x_P)] \\ &= H(D(x_P), y_P) \end{aligned} \tag{6}$$

with l considered in this case as the binary cross entropy loss function H, and $y_P = 0$ the label associated to the positive samples. z is the input random vector, of the generative model G, which can follow a uniform or normal distribution. Consequently, if we impose the generative model G during the adversarial training to converge towards the distribution of unlabeled samples x_U, it will in fact only converge towards the subdivision of the distribution p_U for which the discriminator output prediction is the closest to the label value associated to unlabeled samples. In other words, by combining a PU loss function with the minimax value function of a GAN, such that the same label is associated to the fake generated samples and to the positive labeled samples, this forces the GAN convergence towards the unlabled distribution while diverging to the positive sample distribution. In this way, the GAN tends to only converge towards the unlabeled negative samples distribution. In summary, with the proposed approach, the representation of the unlabeled counter-examples of the positive class is learned by the generative model thanks to the discriminator point of view guidelines. An implementation of the first step is described in annexe section 5.

Once the D-GAN training is finished, we can start the second step which consists in using a classifier C to distinguish fake generated samples $x_{FN} = G(z)$, which are ideally equivalent to the real negative samples, from real positive labeled samples as illustrated in figure 2. The divergent GAN behaviour is only possible if the discriminator D is able to distinguish positive and negative samples such that we have $supp(p_P(x_P)) \cap supp(p_N(x_N)) \rightarrow \emptyset$, with $supp$ the support function of probability distributions. Nonetheless, in the worst-case scenario where the discriminator is not able to sufficiently encode the complexity of the boundary between positive and negative samples, the D-GAN will then converge towards the behaviour of the P-GAN, which seems to be the best solution in this critic situation. This is demonstrated experimentally in the next section.

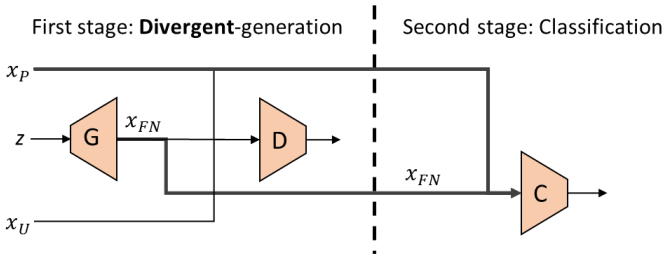


Figure 2: Proposed Positive Unlabeled system: Divergent-GAN Learning model. x_{FN} represents the fake generated samples which are similar to real negative samples x_N. G is the generative model. D is the discriminator. C is the classifier which is used to perform a binary Positive-Negative classification.

2.2 An extension to noisy labeled learning and few labeled data ressources

If a generative model in an adversarial training only converges towards the subdivision of a distribution which is considered as the closest to a given label from the discriminator point of view, then we can extend the proposed approach to the noisy labeled learning task. More precisely, the D-GAN architecture can be modified to include a generative model G_P trained to generate fake positive samples. This is achieved by adding the loss function term $logD(G_P(z))$ in the minimax value function. Furthermore, this extension also allows to manage problems where we only have few labeled positive samples. Indeed, the positive generator G_P will converge towards the distribution considered as the most positive, the closest to zero in our case, which includes also the unlabeled positive samples perceived by the discriminator at the same location that the labeled ones as illustrated in figure 1 (d). We argue that it is indirectly the same phenomenon that allows the GenPU (Hou et al., 2018) to get relatively good accuracy predictions with few labeled positive samples. Because the discriminator task becomes more difficult, the weight α is applied on the unlabeled loss function $log(D(x_U))$ and on the positive one $log(1 - D(x_P))$. It ensures the guidelines that both generators G and G_P have to follow. The boolean variable $noisyLabelsMode$ of the algorithm 1 is set to $True$ to activate this functionality.

2.3 Without discriminator batch normalization

Nowadays, Batch Normalization (BN) (Ioffe & Szegedy, 2015), is considered as a relevant regularization technique commonly used in deep neural networks architectures. Its utility for GANs training has been highlighted by (Radford et al., 2015) for the DCGAN architecture in order to stabilize the adversarial training. Other variants like the Wasserstein-GAN (Arjovsky et al., 2017) or the Loss-Sensitive GAN (Qi, 2017) confirmed its interest. Indeed, as developped in (Ioffe & Szegedy, 2015), BN addresses issues like vanishing, exploding gradient problems, and the risk of getting stuck in a poor local minima. BN suggests to reduce the $internal\ covariate\ shift$ problem of the learning model. In this way, a higher learning rate can be used and it can improve significantly the training speed.

But BN regularizes the model, such that a training sample from a given minibatch is considered in conjunction with other examples of this minibatch. This is the consequence of estimating the mean and variance normalization parameters one time per minibatch, and then applying them on each minibatch sample. It turns out that it does not allow to link labeled positive samples to the unlabeled positive samples and to produce a gap between positive and negative samples, when positive samples x_P and unlabeled samples x_U are not in the same training minibatch. To counter this problem, we could imagine to apply BN on a unified minibatch which contains a fraction of each distribution x_P, x_U and x_F. But the BN effect is influenced a lot by the content of the minibatch on which it is applied. So the fraction π_P of positive samples included in x_U will affect a lot the batchnorm effect.

As illustrated experimentally in figure 5, with BN, we observe that the classifier only learns to distinguish minibatches of positive samples from minibatches of unlabeled samples. On the contrary, without the use of BN, both distributions appear in the unlabeled minibatch from the discriminator

output point of view. In other words, the negative samples can be distinguished from the positive samples without BN.

However, BN benefits in a more traditional training are not negligible, so we suggest to use another type of normalization which could replace BN role in a PU learning task. It turns out that Layer Normalization (LN) (Ba et al., 2016), is a frequently used technique with sequential networks as it can be applied for each sequential sample separately. Indeed, with LN the normalization for a given sample is computed on its resulting output feature map layers and the mean and variance are computed separately for each sample of a minibatch. For this reason, we propose to apply LNs instead of BNs inside our discriminative model D. As we can observe in figure 5, it allows the discriminator to keep the behaviour that interests us which is the ability to distinguish unlabeled negative samples to unlabeled positive samples. Additional qualitative experiments in annexe section 5 show that the LN can substitute BN inside the discriminative model structure to generate acceptable images.

The next section presents experimental results which demonstrate the good performance of the proposed approach.

3 EXPERIMENTAL RESULTS

This section demonstrates experimentally the usability of D-GAN on simple and complex images. We present qualitative results for the counter-example generation, and comparative measurable results for the PU learning classification task.

In our case, the classifier C used for the comparative experiments has two convolutional layers and two prediction outputs, as described in the publication of the PGAN (Chiaroni et al., 2018). It is always trained during 20 epochs. We use the DCGAN (Radford et al., 2015) architecture for all comparative experiments except for celebA and CIFAR-10 datasets where we respectively adapt the LS-GAN and WGAN-GP to the D-GAN framework. The D-GAN uses LN in D instead of BN. However, we continue to use BN in the generator G.

From a qualitative point of view, and on the contrary to the PGAN model, the D-GAN allows to generate samples which are only similar to the counter-examples for varying complex types of data, as illustrated in figures 3, 4 and 7. The method seems to work well on a relatively acceptable size of images to allow real applications. Moreover, in order to allow the proposed approach reproducibility, a D-GAN version which allows to get the results of the figure 3 is available [2] and is based on the implementation code of (Qi, 2017). Our code includes also the method to establish a PU training dataset from a traditional labeled dataset, as suggested by Chiaroni et al. (2018), that we have implemented because we needed it to do comparative results with the PGAN.

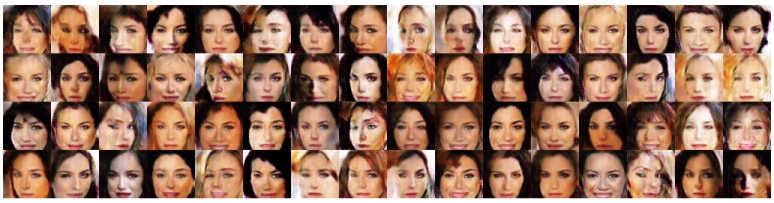
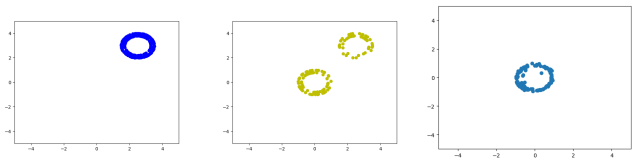


Figure 3: This is a minibatch of fake negative images x_{FN} generated, after 20 epochs iteration, with the D-GAN based on the Loss-Sensitive GAN (Qi, 2017) and by replacing BN with LN. The PU training dataset uses the celebA dataset images with settings $\rho = 0.5$, $\pi_P = 0.3$ and "Male" the positive class chosen arbitrarily. The images size is cropped to 64*64*3. All generated samples are qualitatively relevant fake counter-examples.

[2]The corresponding code of this latter experiment will be publicly available (once the anonymous reviewing period will end).

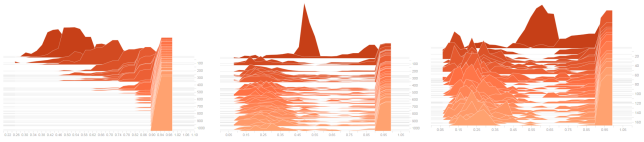


(a) (b) (c)

Figure 4: D-GAN functioning on clusters of 2D points such that D and G have a multilayer-perceptron structure. (a), (b) and (c) are respectively sets of labeled positive, unlabeled, and generated samples. The generated samples only follow the distribution of the counter-examples.

3.1 Normalization effects on the Discriminator

We compare in the figure 5 the ability of D to distinguish the positive from the negative samples distribution when it is trained on a PU dataset with BN, LN layers or without any normalization technique. Smaller is surface intersection between positive and negative samples included in the unlabeled dataset and better it is.



(a) (b) (c)

Figure 5: Normalization effect on the histogram distribution of output values predicted by D on the unlabeled minibatch. (a) is with BN, (b) without any normalization, and (c) with LN. The axe in depth represents the training iterations. The lighter histograms are, the closer training iterations are. Settings are with positive class 8 and negative class 3 of MNIST dataset, with $\rho = 0.5$ and $\pi_P = 0.3$.

We observe that BN does not allow to distinguish positive from negative samples when the discriminator is trained on a PU dataset. On the contrary, LN allows it for the reasons specified in the subsection 2.3.

3.2 Divergent-GAN for Positive Unlabeled learning

As detailed previously in section 2.2, the D-GAN combined with an additional generator can also learn to generate fake positive samples which converge towards the distributon of the unlabeled positive samples and thus allowing an indirect data augmentation. So we use this alternative to do the One vs. One task with few positive labels. We compare the D-GAN accuracy prediction to GenPU on the table 2, by doing the experiment with the D-GAN as possible in the same conditions. In the context of these tests we reused their specified generative model architecture. This allows to get Accuracy performances comparable to GenPU method for one hundred of positive labeled samples after 40 epochs iterations for the D-GAN. However, the GenPU method still presents better results for fewer labeled positive samples, which remains an interesting performance.

By taking into acount the fact that the D-GAN does not need prior knowledge to get these results, we can affirm it works relatively good and even stays globally better than uPU method and close to the nnPU method.

Concerning the One vs. Rest task, the D-GAN globally outperforms the PGAN and RP methods in tems of F1-Score predictions performances on the table 3 both on MNIST and CIFAR-10 datasets, while demonstrating a more robust behaviour against the overfitting problem as illustrated in figure 6.

Table 2: Comparative results of Accuracy prediction performances on MNIST in a One vs. One mode. N_P is the number of positive labeled samples, and N_U is the number of unlabeled samples which mixes all the rest of positive and negative samples.

One vs. One	'3'vs.'5'				'8'vs.'3'			
Dataset: MNIST	D-GAN	GenPU	nnPU	uPU	D-GAN	GenPU	nnPU	uPU
No need of prior knowledge	✓	×	×	×	✓	×	×	×
N_P=100 : N_U=9900	**0.987**	0.983	0.969	0.914	**0.989**	0.982	0.974	0.932
N_P=50 : N_U=9950	0.964	**0.982**	0.966	0.854	0.974	**0.979**	0.965	0.873

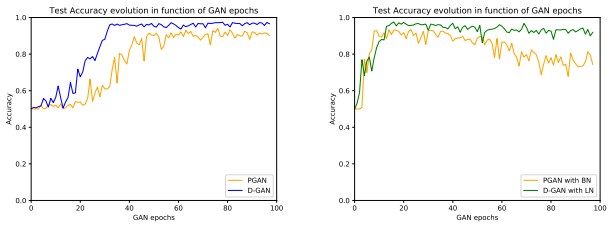
Table 3: Comparative results of average F1-Score prediction performances in a One vs. Rest mode on MNIST and CIFAR-10 with respectively 35 Divergent-GAN epochs on MNIST and 275 Divergent-WGAN-GP epochs on CIFAR-10. Results of PGAN and RP comes from (Chiaroni et al., 2018).

One vs. Rest	ref	$\rho = 0.5, \pi_P = 0.1$			$\rho = 0.5, \pi_P = 0.3$			$\rho = 0.5, \pi_P = 0.5$			$\rho = 0.5, \pi_P = 0.7$		
Datasets	PN	D-GAN	PGAN	RP	D-GAN	PGAN	RP	D-GAN	PGAN	RP	D-GAN	PGAN	RP
AVG_{MNIST}	0.993	**0.99**	0.965	0.967	**0.984**	0.958	0.975	**0.972**	0.946	0.951	0.922	0.875	**0.933**
$AVG_{\text{CIFAR-10}}$	0.680	**0.749**	0.745	0.622	0.727	**0.760**	0.730	**0.749**	0.748	0.716	**0.713**	0.702	0.684

In figure 6(a), without BL or LN, the D-GAN method allows to get a faster and better Accuracy prediction performance than the PGAN when they are trained in exactly the same conditions. In figure 6(b), the D-GAN with LN can follow the rythm of the PGAN training with BN, while keeping a more stable behaviour once the Accuracy progression is finished. The D-GAN does not overfit as the P-GAN on this task.

4 Conclusion

One the one hand, the D-GAN outperforms the PGAN performances for the One vs. Rest challenging task twice on simple and complex datasets. On the other hand it allows the same functionalities as the GenPU method until a certain limit, but without the need of prior knowledge and with a more practical and lighter framework. On the third hand, premices of Noisy labeled learning results allow to plan to extend the proposed approach to a unified generative framework which could output relevant generated labeled samples even from a weak, noisy labeled or incremental dataset with varying fractions of unlabeled positive samples. However, beyond the fact that the proposed approach seems to outperform the following Positive Unlabeled Learning state of the art, we strongly believe that its robustness and prediction performances still have the potential to be improved by taking the best of the PU and representation learning domains. Moreover, some recent promising GAN training approaches like the progressive growing technique (Karras et al., 2018) demonstrated impressive generative performances on large-scaled image datasets, even without the need of BN. This suggests to predict the well functioning of the Divergent-GAN even on higher dimensional data PU learning issues than it has been done until now.



(a) (b)

Figure 6: Classifier test Accuracy evolution in function of the GAN epochs. 8 vs. Rest MNIST task, with $\rho = 0.5$ and $\pi_p = 0.5$. (a) D-GAN and PGAN are trained without normalization layers. (b) D-GAN and PGAN are respectively trained with LN and BN inside the discriminator.

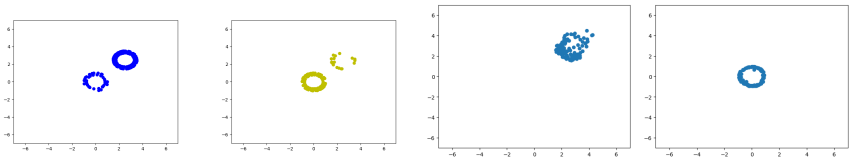
5 ANNEXE

The algorithm 1 is the implementation of the proposed approach, including the option for the noisy labeled learning task.

The table 4 details for each class the average F1-Score results presented in Table 3.

The figure 7 shows the results obtained on clusters of 2D point clouds with the D-GAN, when its noisy labeled mode is activated.

The figure 9 presents the same minibatch generated as in the figure 3, but in a larger size.



(a) (b) (c) (d)

Figure 7: D-GAN Noisy-labeled version functioning on sets of 2D points such that (a), (b), (c) and (d) are respectively sets of real noisy labeled positive (20 percents of noise), real noisy labeled negative (20 percents of noise), fake generated positive and fake generated negative samples. D, G and G_P have a multilayer-perceptron structure. The generated samples only follow the ditribution of the well-labeled samples of the input sets.

Table 4: Comparative results of F1-Score prediction performances on MNIST and CIFAR-10 in a One vs. Rest mode with 20 classifier epochs.

	ref	$\rho = 0.5, \pi_P = 0.1$			$\rho = 0.5, \pi_P = 0.3$			$\rho = 0.5, \pi_P = 0.5$			$\rho = 0.5, \pi_P = 0.7$		
Datasets	PN	D-GAN	PGAN	RP	D-GAN	PGAN	RP	D-GAN	PGAN	RP	D-GAN	PGAN	RP
0	0.997	**0.996**	0.974	0.992	**0.994**	0.973	0.955	0.984	0.973	**0.991**	**0.965**	0.902	0.88
1	0.998	**0.995**	0.971	**0.995**	0.994	0.979	**0.996**	0.988	0.958	**0.994**	0.952	0.863	**0.993**
2	0.99	**0.992**	0.972	0.975	**0.99**	0.959	0.923	**0.98**	0.947	0.936	0.892	0.914	**0.987**
3	0.996	**0.993**	0.963	0.991	0.97	0.953	**0.991**	**0.957**	0.934	0.882	**0.914**	0.885	0.829
4	0.997	**0.989**	0.964	0.972	0.989	0.952	**0.995**	**0.972**	0.945	0.933	0.923	0.914	**0.977**
5	0.993	**0.988**	0.974	0.985	**0.976**	0.95	0.943	**0.969**	0.949	0.91	0.894	0.873	**0.973**
6	0.992	**0.992**	0.962	0.928	0.99	0.959	**0.992**	0.986	0.971	**0.993**	0.961	0.944	**0.99**
7	0.995	**0.982**	0.962	0.947	0.983	0.96	**0.991**	0.969	0.926	**0.988**	0.918	0.737	**0.979**
8	0.995	**0.985**	0.949	0.929	0.981	0.941	**0.982**	**0.966**	0.922	0.941	**0.88**	0.849	0.818
9	0.981	**0.986**	0.959	0.954	0.972	0.956	**0.979**	**0.947**	0.939	0.941	**0.921**	0.865	0.904
AVG_{MNIST}	0.993	**0.99**	0.965	0.967	**0.984**	0.958	0.975	**0.972**	0.946	0.951	0.922	0.875	**0.933**
Plane	0.727	0.759	**0.818**	0.669	0.749	0.784	**0.795**	0.736	**0.758**	0.743	0.715	**0.731**	0.718
Auto	0.78	0.766	**0.801**	0.695	0.749	0.737	**0.829**	0.793	0.789	**0.798**	0.757	0.734	**0.783**
Bird	0.447	0.686	**0.688**	0.56	0.706	**0.744**	0.68	0.692	**0.694**	0.644	0.68	**0.688**	0.542
Cat	0.5	**0.695**	0.658	0.384	0.713	**0.722**	0.651	0.707	**0.718**	0.67	**0.712**	0.69	0.698
Deer	0.698	**0.768**	0.68	0.605	0.681	**0.708**	**0.708**	**0.72**	0.708	0.64	**0.672**	0.633	0.602
Dog	0.567	**0.716**	0.632	0.539	0.68	**0.756**	0.648	0.74	**0.746**	0.733	0.704	0.678	**0.712**
Frog	0.691	0.758	**0.837**	0.666	0.786	0.793	**0.794**	0.78	**0.788**	0.769	0.723	**0.75**	0.749
Horse	0.786	**0.697**	0.693	0.653	0.685	**0.757**	0.723	0.74	0.751	**0.759**	**0.713**	0.675	0.711
Ship	0.832	**0.851**	0.821	0.764	0.784	0.809	**0.831**	**0.797**	0.775	0.785	0.728	0.716	**0.755**
Truck	0.771	0.794	**0.822**	0.685	0.741	**0.786**	0.637	**0.778**	0.754	0.617	**0.732**	0.724	0.564
$AVG_{\text{CIFAR-10}}$	0.680	**0.749**	0.745	0.622	0.727	**0.760**	0.730	**0.749**	0.748	0.716	**0.713**	0.702	0.684

REFERENCES

Martin Arjovsky, Soumith Chintala, and Lon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pp. 214–223, 2017.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Florent Chiaroni, Mohamed-Cherif Rahal, Nicolas Hueber, and Frdric Dufaux. Learning with a generative adversarial network from a positive unlabeled dataset for image classification. In *IEEE International Conference on Image Processing*, 2018.

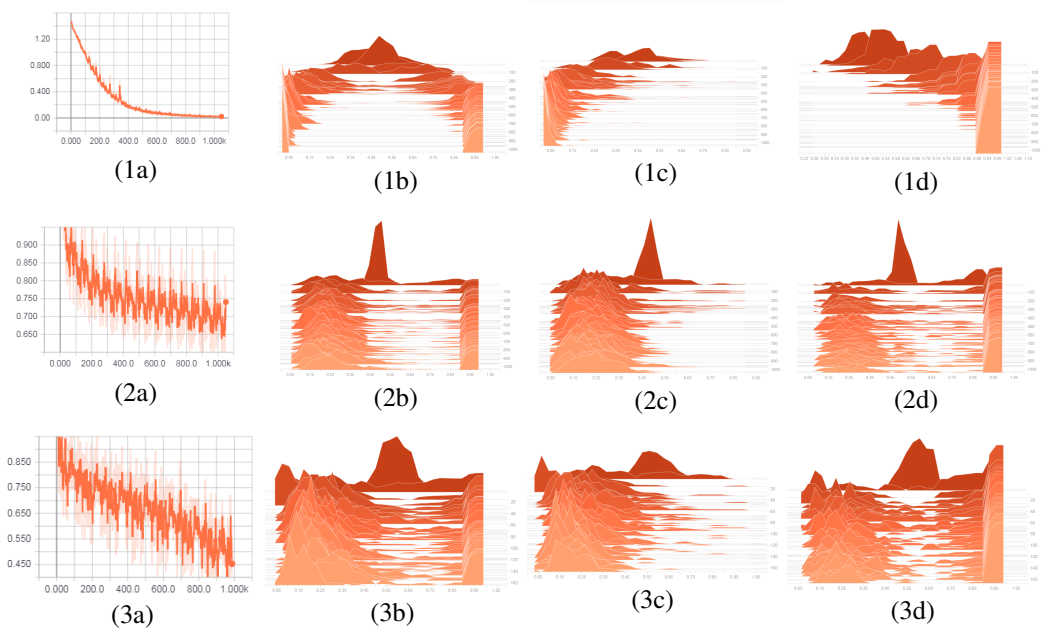


Figure 8: Details of figure 5 Batch Normalization effect on discriminator output predictions on distinct Positive and Unlabeled minibatches. Settings are with positive class 8 and negative class 3 of MNIST dataset, with $\rho = 0.5$ and $\pi_P = 0.3$. The results on the first row are done with BN, without on the second row, and with LN on the third row. The first column (a) is the mean D loss value estimated in function of training iterations. The second column (b) is the histogram distributions of output values predicted by D on the positive and unlabeled minibatches. The third column (c) is the histogram distribution of output values predicted by D on the positive minibatch. The fourth column (d) is the histogram distribution of output values predicted by D on the unlabeled minibatch. The axe in depth represents the training iteration axe on which the clearest histograms gradually correspond to the closest training iterations. Other alternative normalization techniques have been applied to GANs, such as weight normalization (Karras et al., 2018) or the Virtual Batch Normalization (VBN) (Salimans et al., 2016) where they essentially mention the same problem for BN. However, VBN parameters strongly depend to their "reference batch" composition. This is a severe limitation if the fraction of unlabeled positive samples changes during the training, as it can happen for example in a real incremental learning application. Moreover, VBN in (Salimans et al., 2016) is only used in the generator, which is not what we are looking for. For PU learning task, BN is not a problem inside the generator structure.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

C. Craye, D. Filliat, and J. Goudou. Biovision: a biomimetics platform for intrinsically motivated visual saliency learning. *IEEE Transactions on Cognitive and Developmental Systems*, 2018.

Marthinus Du Plessis, Gang Niu, and Masashi Sugiyama. Convex formulation for learning from positive and unlabeled data. In *International Conference on Machine Learning*, pp. 1386–1394, 2015.

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

Algorithm 1 Minibatch stochastic gradient descent training of the Divergent-GAN

Initialize $noisyLabelsMode = false$.
Initialize $\alpha = 2$.
for number of training iterations **do**
  for k steps **do**
    Sample minibatch of m noise samples $\{z^{(1)}, ..., z^{(m)}\}$ from noise prior $p_G(z)$.
    Sample minibatch of m unlabeled examples $\{x_U^{(1)}, ..., x_U^{(m)}\}$ from data distribution p_U.
    Sample minibatch of m positive labeled examples $\{x_P^{(1)}, ..., x_P^{(m)}\}$ from data distribution p_P.

    Update the discriminator D by descending its stochastic gradient:
    if $noisyLabelsMode$ is $false$ **then**

$$\nabla_{\theta_D} \frac{1}{m} \sum_{i=0}^{m} logD(x_U^{(i)}) + log[1 - D(G(z^{(i)}))] + log[1 - D(x_p^{(i)})]$$

    else

$$\nabla_{\theta_D} \frac{1}{m} \sum_{i=0}^{m} \alpha \cdot \big(logD(x_U^{(i)}) + log[1 - D(x_p^{(i)})]\big) + log[1 - D(G(z^{(i)}))] + logD(G_P(z^{(i)}))$$

    end if
  end for
  Sample minibatch of m noise samples $\{z^{(1)}, ..., z^{(m)}\}$ from noise prior $p_G(z)$.
  Update the generator G by descending its stochastic gradient:

$$\nabla_{\theta_G} \frac{1}{m} \sum_{i=0}^{m} log[1 - D(G(z^{(i)}))]$$

  if $noisyLabelsMode$ is $true$ **then**
    Update the generator G_P by descending its stochastic gradient:

$$\nabla_{\theta_{G_P}} \frac{1}{m} \sum_{i=0}^{m} logD(G_P(z^{(i)}))$$

  end if
end for
The gradient-based updates can use any standard gradient-based learning rule. We use Adam in our experiments.

Raia Hadsell, Pierre Sermanet, Jan Ben, Ayse Erkan, Marco Scoffier, Koray Kavukcuoglu, Urs Muller, and Yann LeCun. Learning long-range vision for autonomous off-road driving. *Journal of Field Robotics*, 26(2):120–144, 2009.

Ming Hou, Qibin Zhao, Chao Li, and Brahim Chaib-draa. A generative adversarial framework for positive-unlabeled classification. *arXiv preprint arXiv:1711.08054*, 2018.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pp. 448–456, 2015.

Shantanu Jain, Martha White, and Predrag Radivojac. Estimating the class prior and posterior from noisy positives and unlabeled data. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 29*, pp. 2693–2701. Curran Associates, Inc., 2016.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

Ryuichi Kiryo, Gang Niu, Marthinus C du Plessis, and Masashi Sugiyama. Positive-Unlabeled Learning with Non-Negative Risk Estimator. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 1675–1685. Curran Associates, Inc., 2017.

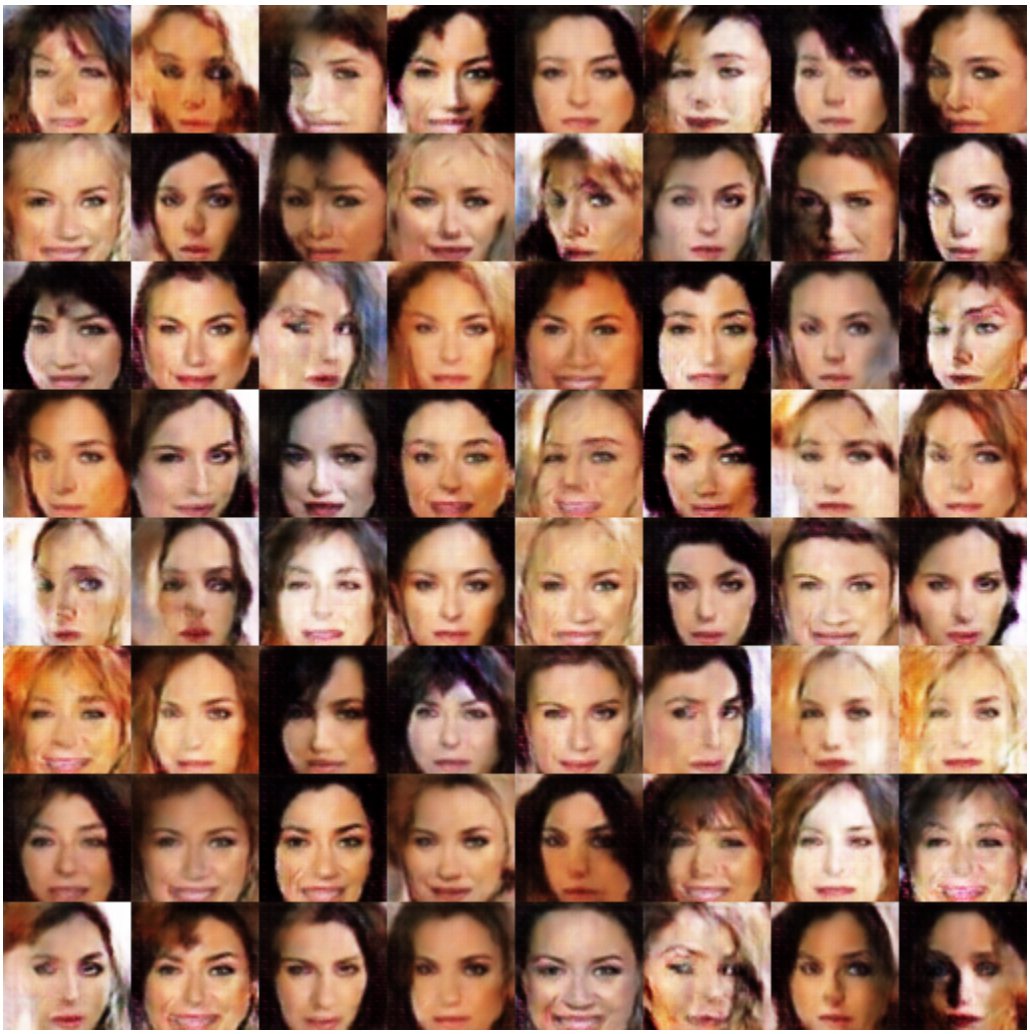


Figure 9: LSGAN-GP generated minibatch with divegent loss on celebA converted in a PU dataset, with $\rho = 0.5$ and $\pi_P = 0.3$ after 20 epochs.

Curtis G. Northcutt, Tailin Wu, and Isaac L. Chuang. Learning with Confident Examples: Rank Pruning for Robust Classification with Noisy Labels. *arXiv preprint arXiv:1705.01936*, 2017.

Guo-Jun Qi. Loss-sensitive generative adversarial networks on lipschitz densities. *arXiv preprint arXiv:1701.06264*, 2017.

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

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

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

Emanuele Sansone, Francesco GB De Natale, and Zhi-Hua Zhou. Efficient training for positive unlabeled learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. LSUN: construction of a large-scale image dataset using deep learning with humans in the loop. *CoRR*, abs/1506.03365, 2015. URL http://arxiv.org/abs/1506.03365.