International Journal of Computer Vision (2020) 128:2665–2683
https://doi.org/10.1007/s11263-020-01348-5

RoCGAN: Robust Conditional GAN

Grigorios G. Chrysos[1] · **Jean Kossaifi**[1,2] · **Stefanos Zafeiriou**[1]

Received: 15 May 2019 / Accepted: 16 June 2020 / Published online: 14 July 2020

Abstract
Conditional image generation lies at the heart of computer vision and conditional generative adversarial networks (cGAN) have recently become the method of choice for this task, owing to their superior performance. The focus so far has largely been on performance improvement, with little effort in making cGANs more robust to noise. However, the regression (of the generator) might lead to arbitrarily large errors in the output, which makes cGANs unreliable for real-world applications. In this work, we introduce a novel conditional GAN model, called *RoCGAN*, which leverages structure in the target space of the model to address the issue. Specifically, we augment the generator with an unsupervised pathway, which promotes the outputs of the generator to span the target manifold, even in the presence of intense noise. We prove that RoCGAN share similar theoretical properties as GAN and establish with both synthetic and real data the merits of our model. We perform a thorough experimental validation on large scale datasets for natural scenes and faces and observe that our model outperforms existing cGAN architectures by a large margin. We also empirically demonstrate the performance of our approach in the face of two types of noise (adversarial and Bernoulli).

Keywords Conditional GAN · Unsupervised learning · Autoencoder · Robust regression · Super-resolution · Adversarial attacks · Cross-noise experiments

1 Introduction

Image-to-image translation and more generally conditional image generation lie at the heart of computer vision. Conditional generative adversarial networks (cGAN) (Mirza and Osindero 2014) have become a dominant approach in the field, e.g. in dense[1] regression (Isola et al. 2017; Pathak et al. 2016; Ledig et al. 2017; Bousmalis et al. 2016; Liu et al. 2017; Miyato and Koyama 2018; Yu et al. 2018; Tulyakov et al. 2018). The major focus so far has been on improving the performance; we advocate instead that improving the generalization performance, e.g. as measured under intense noise and test-time perturbations, is a significant topic with a host of applications, e.g. facial analysis (Georgopoulos et al. 2018). If we aim to utilize cGAN or similar methods as a production technology, they need to have performance guarantees even under large amount of noise. To that end, we study the robustness of conditional GAN under noise.

Conditional Generative Adversarial Networks consist of two modules, namely a generator and a discriminator. The role of the generator role is to map the source signal, e.g. prior information in the form of an image or text, to the target signal. This mapping is completed in two steps: the source signal is embedded into a low-dimensional, latent subspace, which is then mapped to the target subspace. The generator

Communicated by Jun-Yan Zhu, Hongsheng Li, Eli Shechtman, Ming-Yu Liu, Jan Kautz, Antonio Torralba.

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s11263-020-01348-5) contains supplementary material, which is available to authorized users.

✉ Grigorios G. Chrysos
g.chrysos@imperial.ac.uk

Jean Kossaifi
jean.kossaifi@gmail.com

Stefanos Zafeiriou
s.zafeiriou@imperial.ac.uk

[1] Department of Computing, Imperial College London, 180 Queen's Gate, London SW7 2AZ, UK

[2] NVIDIA, Santa Clara, USA

[1] The output includes at least as many dimensions as the input, e.g. super-resolution, or text-to-image translation. We cast conditional image generation as a dense regression task; all the outcomes in this work can be applied to any dense regression task.

is implemented with convolutional or fully connected layers, which are not invariant to (additive) noise. Thus, an input signal that includes (small) additive noise might be mapped arbitrarily off the target manifold (Vidal et al. 2017). In other words, cGAN do not constrain the output to lie in the target manifold which makes them liable to any input perturbation.

A notable line of research, that tackles sensitivity to noise, consists in complementing supervision with an unsupervised learning module. The unsupervised module forms a new pathway that is trained on either the same, or different data samples. The unsupervised pathway enables the network to explore the structure that is not present in the labelled training set, while implicitly constraining the output. The unsupervised module is only required during the training stage, i.e. it is removed during inference. In Rasmus et al. (2015) and Zhang et al. (2016) the authors augment the original bottom up (encoder) network with an additional top-down (decoder) module. The autoencoder, i.e. the bottom-up and the top-down modules combined, forms an auxiliary task to the original classification. However, in contrast to classification studied in Rasmus et al. (2015) and Zhang et al. (2016), in dense regression both bottom-up and top-down modules exist by default, therefore augmenting with an unsupervised module is not trivially extended.

Motivated by the combination of supervised and unsupervised modules, we propose a novel conditional GAN model which implicitly constrains the latent subspace. We coin this new model 'robust conditional GAN' (*RoCGAN*). The motivation behind RoCGAN is to take advantage of the structure in the target space of the model. We learn this structure with an unsupervised module which is included along with our supervised pathway. Specifically, we replace the original generator, i.e. encoder–decoder, with a two pathway module (Fig. 1). Similarly to the cGAN generator, the first pathway performs regression while the second is an autoencoder in the target domain (unsupervised pathway). The two pathways share a similar network structure, i.e. each one includes an encoder–decoder network. The weights of the two decoders are shared to force the latent representations of the two pathways to be semantically similar. Intuitively, this can be thought of as constraining the output of our dense regression to span the target subspace. The unsupervised pathway enables the utilization of all the samples in the target domain even in the absence of a corresponding input sample. During inference, the unsupervised pathway is no longer required, therefore the testing complexity remains the same as in cGAN.

In the following sections, we introduce our novel RoCGAN and study their theoretical/experimental properties (Sect. 2). We prove that RoCGAN share similar theoretical properties with the original GAN, i.e. convergence and optimal discriminator (Sect. 2.5). An experiment with synthetic data is designed to visualize the target subspaces and assess our intuition (Sect. 2.6). We experimentally scrutinize the sensitivity of the hyper-parameters and evaluate our model in the face of intense noise (Sect. 3). Moreover, thorough experimentation with both images from natural scenes and human faces is conducted in different tasks to evaluate the model. The experimental results demonstrate that RoCGAN outperform *consistently* the baseline cGAN in all cases.

Our contributions are summarized as following:

- We introduce RoCGAN that leverage structure of the target space and promote robustness in conditional image generation and dense regression tasks.
- We scrutinize the model's performance under the effect of noise and adversarial perturbations. This robustness analysis had previously not been studied in the context of conditional GAN.
- A thorough experimental analysis for different tasks is conducted. We outline how RoCGAN performs with lateral connections from encoder to decoder. The source code is made freely available for the community[2].

Our preliminary work in Chrysos et al. (2019b) shares the same underlying idea, however this version is significantly extended. Initially, all the experiments have been conducted from scratch based on the new Chainer (Tokui et al. 2015) implementation[2]. The task of super-resolution is introduced in this version, while the noise and adversarial perturbations are categorized and extended, e.g. iterative attack case. Lastly, the manuscript is significantly modified; the experimental section is written from scratch, while other parts like related work or method section are extended substantially.

In this section, we introduce the related literature on conditional GAN and the lines of research related to our work.

Adversarial attacks (Szegedy et al. 2014; Yuan et al. 2017; Samangouei et al. 2018) is an emerging line of research that correlates with our goal. Adversarial attacks are mostly applied to classification tasks; the core idea is that perturbing input samples with a small amount of noise, often imperceptible to the human eye, can lead to severe classification errors. The adversarial attacks are an active field of study with diverse clustering of the methods (Kurakin et al. 2018), e.g. single/multi-step attack, targeted/non-targeted, white/black box. Several techniques 'defend' against adversarial perturbations. A recent example is the Fortified networks of Lamb et al. (2018) which uses Denoising Autoencoders (Vincent et al. 2008) to ensure that the input samples do not fall off the target manifold. Kumar et al. (2017) estimate the tangent space to the target manifold and use that to insert invariances to the discriminator for classification purposes. Even though RoCGAN share similarities with those methods, the

[2] https://github.com/grigorisg9gr/rocgan.

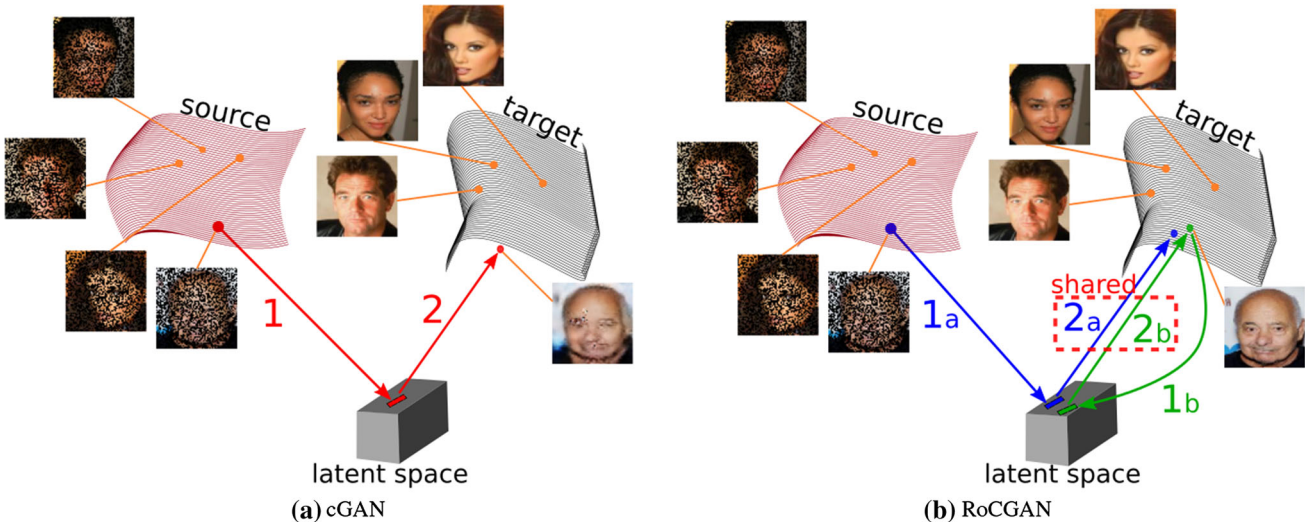


Fig. 1 The mapping process of the generator of the baseline cGAN (**a**) and our model (**b**). **a** The source signal is embedded into a low-dimensional, latent subspace, which is then mapped to the target subspace. The lack of constraints might result in outcomes that are arbitrarily off the target manifold. **b** On the other hand, in RoCGAN, steps 1b and 2b learn an autoencoder in the target manifold and by sharing the weights of the decoder, we restrict the output of the regression (step 2a). All figures in this work are best viewed in color

scope is different since (a) the output of our method is high-dimensional[3] and (b) adversarial examples are not extended to dense regression.[4]

Except for the study of adversarial attacks, combining supervised and unsupervised learning has been used for enhancing the classification performance. In the Ladder network Rasmus et al. (2015) the authors adapt the bottom-up network by adding a decoder and lateral connections between the encoder (original bottom-up network) and the decoder. During training they utilize the augmented network as two pathways: (i) labelled input samples are fed to the initial bottom-up module, (ii) input samples are corrupted with noise and fed to the encoder–decoder with the lateral connections. The latter pathway is an autoencoder; the idea is that it can strengthen the resilience of the network to samples outside the input manifold, while it improves the classification performance.

The effect of noise in the source or target distributions has been the topic of several works. Lehtinen et al. (2018) demonstrate that zero-mean noise in the target distribution does not deteriorate the training, while it might even lead to an improved generalization. The seminal AmbientGAN of Bora et al. (2018) introduces a method to learn from partial or noisy data. They use a measurement function f to simulate the corruption in the output of the generator; they prove that the generator will learn the clean target distribution. The differences with our work are twofold: (a) we do not have access to the corruption function, (b) we do have a prior signal to condition the generator. The works of Li et al. (2019) and Pajot et al. (2019) extend the AmbientGAN with additional cases. Kaneko et al. (2019) and Thekumparampil et al. (2018) study cGAN when the labels are discrete, categorical distributions; they include a noise transition model to clean the noisy labels. Kaneko and Harada (2019) extend the idea to image-to-image translation, i.e. when in addition to the conditional source image, there is a categorical, noisy label. The two main differences from our work are that: (a) we do not have categorical labels, (b) we want to constrain the output of the generator to lie in the target space. A common difference between the aforementioned works and ours is that they do not assess the robustness in the face of adversarial perturbations. Gondim-Ribeiro et al. (2018) conduct a study with adversarial perturbations in auto-encoders and conclude that auto-encoders are well-equipped for such attacks. Kos et al. (2018) propose three adversarial attacks tailored for VAE (Kingma and Welling 2014) and VAE-GAN. Arnab et al. (2018) perform the first large-scale evaluation of adversarial attacks on semantic segmentation models.

Our core goal consists in constraining the model's output. Aside from deep learning approaches, such constraints in manifolds were typically tackled with component analysis. Canonical correlation analysis (Hotelling 1936) has been extensively used for finding common subspaces that maximally correlate the data (Panagakis et al. 2016). The recent work of Murdock et al. (2018) combines the expressiveness

[3] In the classification tasks studied, e.g. the popular ImageNet (Deng et al. 2009), there are up to a thousand classes. On the other hand, our output includes tens or hundreds of thousands of dimensions.

[4] The robustness in our case refers to being resilient to changes in the distribution of the labels (label shift) and training set (covariance shift) (Wang et al. 2017).

of neural networks with the theoretical guarantees of classic component analysis.

1.1 Conditional GAN

Conditional signal generation leverages a conditioning label, e.g. a prior shape (Tran et al. 2019) or an embedded representation (Mirza and Osindero 2014), to produce the target signal. In this work, we focus on the latter setting, i.e. we assume a dense regression task with the conditioning label being an image.

Conditional image generation is a popular task in computer vision, dominated by approaches similar to the original cGAN paper (Mirza and Osindero 2014). The improvements to the original cGAN can be divided into three categories: changes in the (a) architecture of the generator (b) in the architecture of the discriminator, (c) regularization and/or loss terms. The resulting cGAN architectures and their variants have successfully been applied to a host of different tasks, e.g. inpainting (Iizuka et al. 2017; Yu et al. 2018), super-resolution (Ledig et al. 2017). In this paper, our work focuses on improving any cGAN model; we refer to the reader to more targeted applications for a thorough review of specific applications, e.g. super-resolution (Agustsson and Timofte 2017) or inpainting (Wu et al. 2017).

The majority of the architectures in the generator follow the influential work of Isola et al. (2017), widely known as 'pix2pix', that includes lateral skip connections between the encoder and the decoder of the generator. Similarly to lateral connections, residual blocks are often utilized (Ledig et al. 2017; Chrysos et al. 2019a). An additional engineering improvement is to include multiscale generation introduced by Yang et al. (2017). Coarse-to-fine architectures often emerge by training more generators, e.g. in Huang et al. (2017) and Ma et al. (2017) they utilize one generator for the global structure and one for the fine-grained result.

The discriminator in Mirza and Osindero (2014) accepts a generated signal and the corresponding target signal. Isola et al. (2017) make two core modifications in the discriminator (applicable to image-to-image translations): (a) it accepts pairs of source/gt and source/model output images, (b) the discriminator extracts patches instead of the whole image. Miyato and Koyama (2018) replace the inputs to the discriminator with a dot product of the source/gt and source/model output images. In Iizuka et al. (2017), they include two discriminators, one for the global structure and one for the local patches (block inpainting task).

The goal of the aforementioned improvements is to improve the performance or stabilize the training; none of these techniques' aim is to make cGAN more robust to noise. Therefore, our work is perpendicular to all such architecture changes and can be combined with any of the aforementioned architectures.

On the other hand, adding regularization terms in the loss function can impose stronger supervision, thus restricting the output. A variety of additional loss terms have been proposed for regularizing cGAN. The feature matching loss (Salimans et al. 2016) was proposed for stabilizing the training of the discriminator; it measures the discrepancy of the representations (in some layer) of the discriminator. The motivation lies in matching the low-dimensional distributions created by the discriminator layers. Isola et al. (2017) propose a content loss (implemented as ℓ_1 loss) for measuring the per pixel discrepancy of the generated versus the target signal. The perceptual loss is used in Ledig et al. (2017) and Johnson et al. (2016) instead of a per pixel loss. The perceptual loss denotes the difference between the representations[5] of the target and the generated signal. Frequently, task-specific losses are utilized, such as identity preservation or symmetry loss in Huang et al. (2017).

The aforementioned regularization terms provide implicit supervision in the generator's output through similarity with the target signal. However, this supervision does not restrict the generated signals to lie in the target manifold.

2 Method

In this section, we elucidate our proposed RoCGAN. In the following paragraphs, we develop the problem statement (Sect. 2.1), we review the original conditional GAN model (Sect. 2.2), and introduce RoCGAN (Sect. 2.3). Sequentially, we study a special case of generators, i.e. the generators that include lateral skip connections from the encoder to the decoder, and we pose the modifications required (Sect. 2.4). In Sect. 2.5, we prove that RoCGAN share the same properties as the original GAN (Goodfellow et al. 2014) and in Sect. 2.6 the intuition behind the model is assessed with synthetic data.

2.1 Problem Statement

The task of conditional signal generation is posed as generating signals given an input label[6] $\boldsymbol{s}$. We assume the label $\boldsymbol{s} \in \boldsymbol{S}$, where $\boldsymbol{S}$ is the domain of labels, follows a different distribution from the target signals $\boldsymbol{y} \in \boldsymbol{Y}$, where $\boldsymbol{Y}$ is the domain of target signals. Also, we frequently want to include some stochasticity in the mapping; we include a latent variable $\boldsymbol{z} \in \boldsymbol{Z}$ where $\boldsymbol{Z}$ is a known distribution, e.g. Gaussian.

[5] Typically those representations are extracted from a pretrained network, e.g. VGG19.

[6] In this work, we will interchangeably refer to this as the input/conditioning label or source signal.

Mathematically, if $\boldsymbol{G}$ denotes the mapping we want to learn, then:

$$\boldsymbol{G} : \boldsymbol{S} \times \boldsymbol{Z} \rightarrow \boldsymbol{Y} \tag{1}$$

To learn $\boldsymbol{G}$, we assume we have access to a database of N pairs $D = \{(\boldsymbol{s}^{(1)}, \boldsymbol{y}^{(1)}), \ldots, (\boldsymbol{s}^{(n)}, \boldsymbol{y}^{(n)}), \ldots, (\boldsymbol{s}^{(N)}, \boldsymbol{y}^{(N)})\}$ with $n \in [1, N]$. In the following paragraphs we drop the index, i.e. we denote $\boldsymbol{s}^{(n)}$ as $\boldsymbol{s}$, to avoid cluttering the notation.

Conditional GAN, which we develop below, have been dominating in the literature for learning such mappings $\boldsymbol{G}$. However, our interest lies in studying the case that during inference time the source signal is $\boldsymbol{s} + \boldsymbol{f}(\boldsymbol{s}, \boldsymbol{G})$ instead of $\boldsymbol{s}$, i.e. there is some unwanted noise in our source signal. We argue that such noise is of both theoretical and practical value for commercial applications.

Notation A bold letter represents a vector/tensor; a plain letter designates a scalar number. Unless explicitly mentioned otherwise $||\cdot||$ will declare an ℓ_1 norm. The symbols $\mathcal{L}_*$ define loss terms, while λ_* denote regularization hyperparameters optimized on the validation set. For a matrix $\mathbf{M}$, diag($\mathbf{M}$) denotes its diagonal elements.

2.2 Conditional GAN

GAN consist of a generator and a discriminator module commonly optimized with alternating gradient descent. The generator's goal is to model the target distribution p_d, while the discriminator's to discern the samples synthesized by the generator and the target (ground-truth) distributions. More precisely, the generator samples $\boldsymbol{z}$ from a prior distribution p_z, e.g. uniform, and maps that to a sample; the discriminator $\boldsymbol{D}$ tries to distinguish between the synthesized sample and one sample from p_d.

The idea behind conditional GAN (cGAN) (Mirza and Osindero 2014) is to provide some additional labels to the generator. The generator $\boldsymbol{G}$ typically takes the form of an encoder–decoder network, where the encoder projects the label into a low-dimensional latent subspace and the decoder performs the opposite mapping, i.e. from low-dimensional to high-dimensional subspace. In other words, the generator performs the regression from the source to the target signal.

The core loss of cGAN is the adversarial loss, which determines the alternating role of the generator and the discriminator:

$$\begin{aligned}\mathcal{L}_{adv} = &\mathbb{E}_{\boldsymbol{s},\boldsymbol{y}\sim p_d(\boldsymbol{s},\boldsymbol{y})}[\log \boldsymbol{D}(\boldsymbol{y}|\boldsymbol{s})] \\ &+ \mathbb{E}_{\boldsymbol{s}\sim p_d(\boldsymbol{s}), z\sim p_z(z)}[\log(1 - \boldsymbol{D}(\boldsymbol{G}(\boldsymbol{s}, \boldsymbol{z})|\boldsymbol{s}))]\end{aligned} \tag{2}$$

The loss is optimized through the following min-max problem:

$$\begin{aligned}\min_{\boldsymbol{w}_G}\max_{\boldsymbol{w}_D} \mathcal{L}_{adv} = &\min_{\boldsymbol{w}_G}\max_{\boldsymbol{w}_D} \mathbb{E}_{\boldsymbol{s},\boldsymbol{y}\sim p_d(\boldsymbol{s},\boldsymbol{y})}[\log \boldsymbol{D}(\boldsymbol{y}|\boldsymbol{s}, \boldsymbol{w}_D)] \\ &+ \mathbb{E}_{\boldsymbol{s}\sim p_d(\boldsymbol{s}), z\sim p_z(z)}[\log(1 - \boldsymbol{D}(\boldsymbol{G}(\boldsymbol{s}, \boldsymbol{z}|\boldsymbol{w}_G)|\boldsymbol{s}, \boldsymbol{w}_D))]\end{aligned}$$

where $\boldsymbol{w}_G$, $\boldsymbol{w}_D$ denote the generator's and the discriminator's parameters respectively. To simplify the notation, we drop the dependencies on the parameters and the noise $\boldsymbol{z}$ in the rest of the paper. In our experiments, we use a discriminator that is not conditioned on the input, i.e. $\boldsymbol{D}(\boldsymbol{y})$; we include a related ablation study in Sect. 3.4.3.

Aside of the adversarial loss, cGAN models include auxiliary losses, e.g. task-specific ℓ_1 reconstruction or regularization terms for discriminator. Those losses do not affect the core model nor its adaptation to RoCGAN; we symbolize with $\mathcal{L}_{cGAN}$ the total loss function.

2.3 RoCGAN

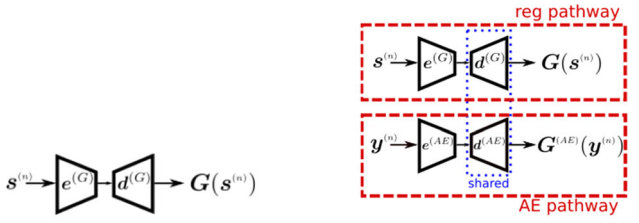
Our main goal is to improve robustness to noise in dense regression tasks. To that end, we introduce our model that leverages structure in the target space of the model to enhance the generator's regression. Our model shares the same structure as cGAN, i.e. it consists of a generator that performs the regression and a discriminator that separates the synthesized from the target signal. We achieve our goal by constructing a generator that includes two pathways.

The generator of RoCGAN includes two pathways instead of the single pathway of the original cGAN. The first pathway, referred as *reg pathway* henceforth, performs a similar regression as its counterpart in cGAN; it accepts a sample from the source domain and maps it to the target domain. We introduce an additional unsupervised pathway, named *AE pathway*. AE pathway works as an autoencoder in the target domain. Both pathways consist of similar encoder–decoder networks.[7] By sharing the weights of their decoders, we promote the regression outputs to span the target manifold and not induce arbitrarily large errors. A schematic of the generator is illustrated in Fig. 2. The discriminator can remain the same as the cGAN: it accepts the reg pathway's output along with the corresponding target sample as input.

To simplify the notation below, the superscript 'AE' abbreviates modules of the AE pathway and 'G' modules of the reg pathway.We denote $\boldsymbol{G}(\boldsymbol{s}) = \boldsymbol{d}^{(G)}(\boldsymbol{e}^{(G)}(\boldsymbol{s}))$ the output of the reg pathway and $\boldsymbol{G}^{(AE)}(\boldsymbol{y}) = \boldsymbol{d}^{(AE)}(\boldsymbol{e}^{(AE)}(\boldsymbol{y}))$ the output of the AE pathway; $\boldsymbol{e}, \boldsymbol{d}$ symbolize the encoder and decoder of a pathway respectively.

The unsupervised module (autoencoder in the target domain) contributes the following loss term:

[7] In principle the encoders' architectures might differ, e.g. when the two domains differ in dimensionality.



(a) cGAN (b) RoCGAN

Fig. 2 Schematic of the generator of **a** cGAN versus **b** our proposed RoCGAN. The single pathway of the original model is replaced with two pathways

$$\mathcal{L}_{AE} = \mathbb{E}_{\boldsymbol{y}\sim p_d(\boldsymbol{y})}[f_d^{AE}(\boldsymbol{y}, \boldsymbol{G}^{(AE)}(\boldsymbol{y}))] \tag{3}$$

where f_d^{AE} denotes a function to measure the divergence.[8]

Despite sharing the weights of the decoders, we cannot ensure that the latent representations of the two pathways span the same subspace. To further reduce the distance of the two representations in the latent space, we introduce the latent loss term $\mathcal{L}_{lat}$. This term minimizes the distance between the encoders' outputs, i.e. the two representations are spatially close (in the subspace spanned by the encoders). The latent loss term is:

$$\mathcal{L}_{lat} = \mathbb{E}_{\boldsymbol{s},\boldsymbol{y}\sim p_d(\boldsymbol{s},\boldsymbol{y})}[f_d^{lat}(\boldsymbol{e}^{(G)}(\boldsymbol{s}), \boldsymbol{e}^{(AE)}(\boldsymbol{y}))] \tag{4}$$

where f_d^{lat} can be any divergence function. In practice, for both $\mathcal{L}_{lat}$ and $\mathcal{L}_{AE}$ we employ ordinary loss functions, e.g. ℓ_1 or ℓ_2 norms. As a future step we intend to replace the latent loss term $\mathcal{L}_{lat}$ with a kernel-based method (Gretton et al. 2007) or a learnable metric for matching the distributions (Ma et al. 2018).

The final loss function of RoCGAN combines the loss terms of the original cGAN $\mathcal{L}_{cGAN}$ with the additional two terms for the AE pathway:

$$\mathcal{L}_{RoCGAN} = \mathcal{L}_{cGAN} + \lambda_{ae} \cdot \mathcal{L}_{AE} + \lambda_l \cdot \mathcal{L}_{lat} \tag{5}$$

2.4 RoCGAN with Skip Connections

The RoCGAN model of Sect. 2.3 describes a family of networks and not a predefined set of layers. A special case of RoCGAN emerges when skip connections from the encoder to the decoder are included. In this section, skip connections refer only to the case of lateral skip connections from the encoder to the decoder. We study below the modifications required for this case.

Skip connections are frequently used as they enable deeper layers to capture more abstract representations without the need of memorizing all the information. The shortcut connection allows a low-level representation from an encoder layer to be propagated directly to a decoder layer without passing through the long path, i.e. the network without the lateral skip connections. An autoencoder (AE) with such a skip connection can achieve close to zero reconstruction error by simply propagating the representation through the shortcut. This shatters the signal in the long path (Rasmus et al. 2015), which is an unwanted behavior.

To achieve training the long path, we explore a number of regularization methods. Our first approach in our original work was to include a regularization loss term. In this work, we propose an additional regularization technique for the skip case.

In the first approach, we implicitly tackle the issue by maximizing the variance captured by the longer path representations. We add a loss term that penalizes the correlations in the representations (of a layer) and thus implicitly encourage the representations to capture diverse and useful information. We implement the decov loss (Cogswell et al. 2016):

$$\mathcal{L}_{decov} = \frac{1}{2}\left(||\boldsymbol{C}||_F^2 - ||\text{diag}(\boldsymbol{C})||_2^2\right) \tag{6}$$

where $\boldsymbol{C}$ is the covariance matrix of the layer's representations. The loss is minimized when the covariance matrix is diagonal, i.e. it imposes a cost to minimize the covariance of hidden units without restricting the diagonal elements that include the variance of the hidden representations.

A similar loss is explored by Valpola (2015), where the decorrelation loss is applied in every layer. Their loss term has stronger constraints: (i) it favors an identity covariance matrix but also (ii) penalizes the smaller eigenvalues of the covariance more. We have not explored this alternative loss term, as the decov loss worked in our case without the additional assumptions of the Valpola (2015).

In this work, we consider an alternative regularization technique. The approach is motivated by Rasmus et al. (2015) who include noise in the lateral skip connections. We do include zero-mean Gaussian noise in the shortcut connection, i.e. the representation of the encoder is modified by some additive Gaussian noise when skipped to the decoder. In our experimentation, both approaches can lead to improved results, we prefer to use the latter in the experiments.

2.5 Theoretical Analysis

In the next few paragraphs, we prove that RoCGAN share the properties of the original GAN (Goodfellow et al. 2014). Even though the derivations follow similar steps as the original GAN, but are added to make the paper self-contained.

[8] The $\mathcal{L}_{AE}$ can also leverage unpaired samples in the target domain. That is, if we have M samples $\{\boldsymbol{y}_U^{(1)}, \ldots, \boldsymbol{y}_U^{(m)}, \ldots, \boldsymbol{y}_U^{(M)}\}$ available, we can use them to improve the AE pathway.

We derive the optimal discriminator and then compute the optimal value of $\mathcal{L}_{adv}(\boldsymbol{G}, \boldsymbol{D})$.

Proposition 1 *For a fixed generator $\boldsymbol{G}$ (reg pathway), the optimal discriminator is:*

$$\boldsymbol{D}^* = \frac{p_d(\boldsymbol{s}, \boldsymbol{y})}{p_d(\boldsymbol{s}, \boldsymbol{y}) + p_g(\boldsymbol{s}, \boldsymbol{y})} \tag{7}$$

where p_g is the model (generator) distribution.

Proof Since the generator is fixed, the goal of the discriminator is to maximize the $\mathcal{L}_{adv}$ where:

$$\begin{aligned}\mathcal{L}_{adv}(\boldsymbol{G}, \boldsymbol{D}) &= \int_{\boldsymbol{y}}\int_{\boldsymbol{s}} p_d(\boldsymbol{y}, \boldsymbol{s}) \log \boldsymbol{D}(\boldsymbol{y}|\boldsymbol{s}) \boldsymbol{dyds} \\ &\quad + \int_{\boldsymbol{s}}\int_{\boldsymbol{z}} p_d(\boldsymbol{s}) p_z(\boldsymbol{z}) \log(1 - \boldsymbol{D}(\boldsymbol{G}(\boldsymbol{s}, \boldsymbol{z})|\boldsymbol{s})) \boldsymbol{dsdz} \\ &= \int_{\boldsymbol{y}}\int_{\boldsymbol{s}} p_d(\boldsymbol{s}, \boldsymbol{y}) \log \boldsymbol{D}(\boldsymbol{y}|\boldsymbol{s}) \boldsymbol{dy} \\ &\quad + p_g(\boldsymbol{s}, \boldsymbol{y}) \log(1 - \boldsymbol{D}(\boldsymbol{y}|\boldsymbol{s})) \boldsymbol{dyds}\end{aligned} \tag{8}$$

To maximize the $\mathcal{L}_{adv}$, we need to optimize the integrand above. We note that with respect to $\boldsymbol{D}$ the integrand has the form $f(y) = a \cdot log(y) + b \cdot log(1 - y)$. The function f for $a, b \in (0, 1)$ as in our case, obtains a global maximum in $\frac{a}{a+b}$, so:

$$\begin{aligned}\mathcal{L}_{adv}(\boldsymbol{G}, \boldsymbol{D}) \leq &\int_{\boldsymbol{y}}\int_{\boldsymbol{s}} p_d(\boldsymbol{s}, \boldsymbol{y}) \log \boldsymbol{D}^*(\boldsymbol{y}|\boldsymbol{s}) \boldsymbol{dy} \\ &+ p_g(\boldsymbol{s}, \boldsymbol{y}) \log(1 - \boldsymbol{D}^*(\boldsymbol{y}|\boldsymbol{s})) \boldsymbol{dyds}\end{aligned} \tag{9}$$

with

$$\boldsymbol{D}^* = \frac{p_d(\boldsymbol{s}, \boldsymbol{y})}{p_d(\boldsymbol{s}, \boldsymbol{y}) + p_g(\boldsymbol{s}, \boldsymbol{y})} \tag{10}$$

thus $\mathcal{L}_{adv}$ obtains the maximum with $\boldsymbol{D}^*$. □

Proposition 2 *Given the optimal discriminator $\boldsymbol{D}^*$ the global minimum of $\mathcal{L}_{adv}$ is reached if and only if $p_g = p_d$, i.e. when the model (generator) distribution matches the data distribution.*

Proof From Proposition 1, we have found the optimal discriminator as $\boldsymbol{D}^*$, i.e. the $\arg\max_{\boldsymbol{D}} \mathcal{L}_{adv}$. If we replace the optimal value we obtain:

$$\begin{aligned}&\max_{\boldsymbol{D}} \mathcal{L}_{adv}(\boldsymbol{G}, \boldsymbol{D}) \\ &= \int_{\boldsymbol{y}}\int_{\boldsymbol{s}} p_d(\boldsymbol{s}, \boldsymbol{y}) \log \boldsymbol{D}(\boldsymbol{y}|\boldsymbol{s}) \boldsymbol{dy} \\ &\quad + p_g(\boldsymbol{s}, \boldsymbol{y}) \log(1 - \boldsymbol{D}(\boldsymbol{y}|\boldsymbol{s})) \boldsymbol{dyds} \\ &= \int_{\boldsymbol{y}}\int_{\boldsymbol{s}} p_d(\boldsymbol{s}, \boldsymbol{y}) \log \left(\frac{p_d(\boldsymbol{s}, \boldsymbol{y})}{p_d(\boldsymbol{s}, \boldsymbol{y}) + p_g(\boldsymbol{s}, \boldsymbol{y})} \right) \\ &\quad + p_g(\boldsymbol{s}, \boldsymbol{y}) \log \left(1 - \frac{p_d(\boldsymbol{s}, \boldsymbol{y})}{p_d(\boldsymbol{s}, \boldsymbol{y}) + p_g(\boldsymbol{s}, \boldsymbol{y})} \right) \boldsymbol{dyds} \\ &= \int_{\boldsymbol{y}}\int_{\boldsymbol{s}} p_d(\boldsymbol{s}, \boldsymbol{y}) \log \left(\frac{p_d(\boldsymbol{s}, \boldsymbol{y})}{p_d(\boldsymbol{s}, \boldsymbol{y}) + p_g(\boldsymbol{s}, \boldsymbol{y})} \right) \\ &\quad + p_g(\boldsymbol{s}, \boldsymbol{y}) \log \left(\frac{p_g(\boldsymbol{s}, \boldsymbol{y})}{p_d(\boldsymbol{s}, \boldsymbol{y}) + p_g(\boldsymbol{s}, \boldsymbol{y})} \right) \boldsymbol{dyds}\end{aligned} \tag{11}$$

We add and subtract log(2) from both terms, which after few math operations provides:

$$\begin{aligned}\max_{\boldsymbol{D}} \mathcal{L}_{adv}(\boldsymbol{G}, \boldsymbol{D}) = &-2 \cdot \log(2) + KL\left(p_d || \frac{p_d + p_g}{2} \right) \\ &+ KL\left(p_g || \frac{p_d + p_g}{2} \right)\end{aligned}$$

where in the last row KL symbolizes the Kullback–Leibler divergence. The latter one can be rewritten more conveniently with the help of the Jensen–Shannon (JSD) divergence as

$$\max_{\boldsymbol{D}} \mathcal{L}_{adv}(\boldsymbol{G}, \boldsymbol{D}) = -\log(4) + 2 \cdot JSD(p_d || p_g) \tag{12}$$

The Jensen–Shannon divergence is non-negative and obtains the zero value only if $p_d = p_g$. Equivalently, the last equation has a global minimum (under the constraint that the discriminator is optimal) when $p_d = p_g$. □

2.6 Experiment on Synthetic Data

We design an experiment on synthetic data to explore the differences between the original generator and our two pathway generator. Specifically, we design a network where each encoder/decoder consists of two fully connected layers; each layer followed by a RELU. We optimize the generators only, to avoid adding extra learned parameters.

The inputs/outputs of this network span a low-dimensional space, which depends on two independent variables $x, y \in [-1, 1]$. We've experimented with several arbitrary functions in the input and output vectors and they perform in a similar way. We exhibit here the case with input vector $[x, y, e^{2x}]$ and output vector $[x + 2y + 4, e^x + 1, x + y + 3, x + 2]$. The reg pathway accepts the three inputs, projects it into a two-dimensional space and the decoder maps it to the target four-dimensional space.

We train the baseline and the autoencoder modules separately and use their pre-trained weights to initialize the two pathway network. The loss function of the two pathway network consists of the $\mathcal{L}_{lat}$ (Eq. 4) and ℓ_2 content losses in the two pathways. The networks are trained either till convergence or till 100,000 iterations (batch size 128) are completed.

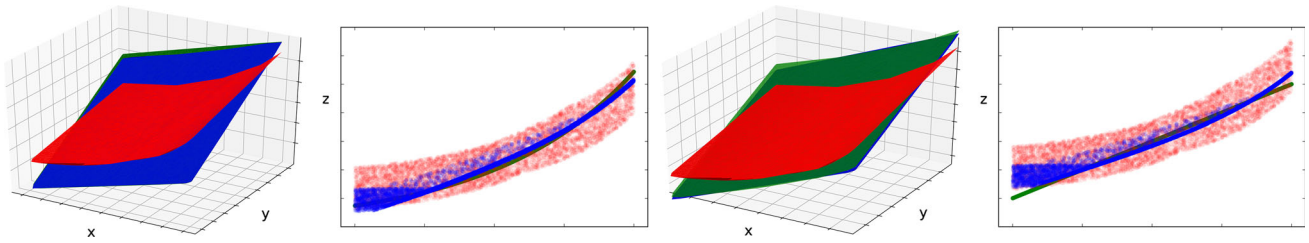


Fig. 3 Qualitative results in the synthetic experiment of Sect. 2.6. Each plot corresponds to the respective manifolds in the output vector; the first and third depend on both x, y (xyz plot), while the rest on x (xz plot). The green color visualizes the target manifold, the red the baseline and the blue ours. Even though the two models include the same parameters during inference, the baseline does not approximate the target manifold as well as our method (Color figure online)

During testing, 6400 new points are sampled and the overlaid results are depicted in Fig. 3; the individual figures for each output can be found in the supplementary. The ℓ_1 errors for the two cases are: 9843 for the baseline and 1520 for the two pathway generator. We notice that the two pathway generator approximates the target manifold better with the same number of parameters during inference.

3 Experiments

In the following paragraphs we initially design and explain the noise models (Sect. 3.1), we review the implementation details (Sect. 3.2) and the experimental setup (Sect. 3.3). Sequentially, we conduct an ablation study and evaluate our model on real-worlds datasets, including natural scenes and human faces.

3.1 Noise Models

In this work, we explore two different types of noise with multiple variants tested in each type. Those two types are Bernoulli noise and adversarial noise.

Bernoulli noise For an input $\boldsymbol{s}$, the noise model is represented by a Bernoulli function $\Phi_v(\boldsymbol{s}, \theta)$. Specifically, we have

$$\Phi_v(\boldsymbol{s}, \theta)_{i,j} = \begin{cases} v & \text{with probability } \theta \\ \boldsymbol{s}_{i,j} & \text{with probability } 1-\theta \end{cases} \tag{13}$$

To provide a practical example, assume that $v = 0$ and $\theta = 0.5$, then an image $\boldsymbol{s}$ has half of its pixels converted to black, which is known as sparse inpainting.

Adversarial examples Apart from testing in the face of additional Bernoulli noise, we explore adversarial attacks in the context of dense regression. Recent works, e.g. Szegedy et al. (2014), Yuan et al. (2017), Samangouei et al. (2018) and Madry et al. (2018), explore the robustness of (deep) classifiers.

Contrary to classification case, there has not been much investigation of adversarial attacks in the context of image-to-image translation or any other dense regression task. However, since an adversarial example perturbs the source signal, dense regression tasks can be vulnerable to such modifications. We conduct a thorough investigation of this phenomenon by attacking our model with three adversarial attacks for dense regression. We introduce the adversarial attacks in the following paragraphs.

The first, and most ubiquitous attack is the fast gradient sign method (FGSM), introduced by Goodfellow et al. (2015). It is the simplest attack and the basis for several variants. In addition, the authors of Dou et al. (2018) mathematically prove the efficacy of this attack in the classification case. Let us define the auxiliary function:

$$\boldsymbol{u}(\boldsymbol{s}) = \boldsymbol{s} + \epsilon \text{sign}\left(\nabla_{\boldsymbol{s}} \mathcal{L}(\boldsymbol{s}, \boldsymbol{y})\right) \tag{14}$$

with $\mathcal{L}(\boldsymbol{s}, \boldsymbol{y}) = ||\boldsymbol{y} - \boldsymbol{G}(\boldsymbol{s})||_1$. Then, each source signal $\boldsymbol{s}$ is modified as:

$$\tilde{\boldsymbol{s}} = \boldsymbol{s} + \boldsymbol{\eta} \tag{15}$$

The perturbation $\boldsymbol{\eta}$ is defined as:

$$\boldsymbol{\eta} = \boldsymbol{u}(\boldsymbol{s}) \tag{16}$$

with ϵ a hyper-parameter, $\boldsymbol{y}$ the target signal and $\mathcal{L}$ an appropriate loss function.

However, to make the perturbation stronger, we iterate the gradient computation. The iterative FGSM (IFGSM) method of Dou et al. (2018)[9] is:

$$\tilde{\boldsymbol{s}}_{(k)} = Clip\{\boldsymbol{u}(\tilde{\boldsymbol{s}}_{(k-1)})\} \tag{17}$$

where k is the kth iteration, $\tilde{\boldsymbol{s}}_{(0)} = \boldsymbol{s}$ and $Clip$ function restricts the outputs in the source signal range.

[9] This method is also known as basic iterative method (BIM) Kurakin et al. (2016).

The second attack that is selected is the projected gradient descent method (PGD) of Madry et al. (2018). PGD is an iterative method which given the source signal $\tilde{\boldsymbol{s}}_{(0)} = \boldsymbol{s}$, it modifies it as:

$$\tilde{\boldsymbol{s}}_{(k)} = Clip\{\Pi_{\boldsymbol{s}}\boldsymbol{u}(\tilde{\boldsymbol{s}}_{(k-1)})\} \tag{18}$$

Robustness to this attack typically implies robustness to all first order methods (Madry et al. 2018), making it a particularly interesting case study.

The third adversarial method is the latent attack of Kos et al. (2018). The loss in this attack is computed in the latent space, i.e. the output of the encoder $\boldsymbol{e}^{(G)}(\boldsymbol{s})$.

In the following sections, we model the (I)FGSM attacks with the tuple (k, ϵ) that declare the total iterative steps and the ϵ hyper-parameter value respectively. In the Bernoulli case, we use three cases of v, i.e. $v = 0$ corresponding to black pixels, $v = 1$ corresponding to white pixels and channel-wise $v = 0$. We abbreviate the three cases with a triplet $(\theta_{v=0}, \theta_{v=1}, \theta_{v=0,channel})$ denoting the θ probability in each case. For instance, the triplet (50, 0, 0) denotes Bernoulli noise with $v = 0$ and probability 50%. Unless explicitly mentioned otherwise, the default adversarial attack below is the IFGSM.

3.2 Implementation Details

Conditional GAN model Several cGAN models have been proposed (see Sect. 1.1). In our experiments, we employ a simple cGAN model based on the best (experimental) practices so far Isola et al. (2017), Salimans et al. (2016) and Zhu et al. (2017).

The works of Salimans et al. (2016) and Isola et al. (2017) demonstrate that auxiliary loss terms, i.e. feature matching and content loss, improve the final outcome, hence we consider those as part of the baseline cGAN. The feature matching loss[10] is:

$$\mathcal{L}_f = \mathbb{E}_{\boldsymbol{s},\boldsymbol{y}\sim p_d(\boldsymbol{s},\boldsymbol{y})}||\pi(\boldsymbol{G}(\boldsymbol{s})) - \pi(\boldsymbol{y})|| \tag{19}$$

where $\pi()$ extracts the features from the penultimate layer of the discriminator.

The final loss function for the cGAN is the following:

$$\mathcal{L}_{cGAN} = \mathcal{L}_{adv} + \lambda_c \cdot \underbrace{\mathbb{E}_{\boldsymbol{s},\boldsymbol{y}\sim p_d(\boldsymbol{s},\boldsymbol{y})}[||\boldsymbol{G}(\boldsymbol{s}) - \boldsymbol{y}||]}_{content-loss} + \lambda_\pi \cdot \mathcal{L}_f \tag{20}$$

where λ_c, λ_π are hyper-parameters to balance the loss terms.

RoCGAN model To fairly compare against the aforementioned cGAN model, we make only the following three adaptations: (i) we duplicate the encoder/decoder (for the new AE pathway); (ii) we share the decoder's weights in the two pathways; (iii) we augment the loss function with the additional loss terms. We emphasize that this is only performed for experimental validation; in practice the encoder of the AE pathway can have a different structure or new task-specific loss terms can be introduced; we have made no effort to optimize further RoCGAN. We use ℓ_1 loss for both the $\mathcal{L}_{lat}$ and $\mathcal{L}_{AE}$.

Training details A 'layer' refers to a block of three units: a convolutional unit with a 4×4 kernel size, followed by Leaky RELU and batch normalization (Ioffe and Szegedy 2015). The hyper-parameters introduced by our model are: $\lambda_l = 1$, $\lambda_{ae} = 100$. The values of the common hyper-parameters, e.g. λ_c, λ_π, are the same between the cGAN/RoCGAN. A mild data augmentation technique is utilized for training cGAN/RoCGAN: The training images are reshaped to 75×75 and random patches of 64×64 are fed into the network. Each training image is horizontally flipped with probability 0.5; no other augmentation is used. A constant learning rate of $2 \cdot 10^{-4}$ (same as in Isola et al. 2017) is used for $3 \cdot 10^5$ iterations with a batch size of 64. During training, we run validation every 10^4 iterations and export the best model, which is used for testing. The discriminator consists of 3 convolutional layers followed by a fully-connected layer. The input to the discriminator is either the output of the generator or the respective target image, i.e. we do not condition the discriminator on the source image.

Our workhorse for testing is a network denoted '5layer', because each encoder and decoder consists of 5 layers. In the following experiments 'Baseline-5layer' represents the cGAN '5layer' case, while ours is indicated as 'Ours-5layer'. In the skip case, we add a skip connection from the output of the third layer of the encoder to the respective decoder layer; we add a '-skip' in the respective method name.

We train an adversarial autoencoder (AAE) (Makhzani et al. 2015) as an established method capable of learning compressed representations as an upper performance bound baseline. Each module of the AAE shares the same architecture as its cGAN counterpart, while the AAE is trained with images in the target space. The target images are used as the input to the AAE and its output, i.e. the reconstruction, is used for the evaluation. In our experimental setting, AAE can be thought of as an upper performance limit of RoCGAN/cGAN for a given capacity (number of parameters).

The task selected for our testing is super-resolution by $4\times$. That is, we downsample an image 4 times; we upsample it with bilinear interpolation and use this interpolated as the corrupted image. In the supplementary, we include an experiment with sparse inpainting.

[10] Referred to as projection loss in Chrysos et al. (2019a).

3.3 Experimental Setup

Datasets In addition to validating our model on synthetic data, we utilize a variety of real-world datasets:

- *MS-Celeb* (Guo 2016) is introduced for large scale face recognition. It contains approximately 10 million facial images from 1 million celebrities. The dataset was collected semi-automatically, while the noise was not manually removed from the training images. We export 3 million samples for training and use 100 thousand images for validation.
- *CelebFaces attributes dataset (Celeb-A)* (Liu et al. 2015) consists a popular benchmark for large-scale face attribute classification. Each image is annotated with 40 binary attributes. Celeb-A is used in conjunction with MS-Celeb in this work, where the latter is used for training and the former is used for testing. All the 202,500 samples of Celeb-A are used for testing. This combination is the main focus for our experiments; specifically it is used in Sects. 3.4, 3.5, 3.6 and 3.8.
- *300 Videos in the Wild (300VW)* (Shen et al. 2015) is a benchmark for face tracking; it includes a sparse set of points annotated per frame. It includes three categories of videos with increasing difficulty; in this work we use as testset the most challenging category (categ 3) that includes over 27,000 frames. We use 300VW in Sect. 3.7 for assessing the performance of RoCGAN in video datasets.
- *ImageNet* (Deng et al. 2009) is a large image database with 1000 different objects. An average of over five hundred images per objects exist. In the experiment for natural scenes, we utilize the training set of Imagenet which consists of 1, 2 million images and its testset that includes 98 thousand images (Sect. 3.5).

The two categories of images, i.e. faces and natural scenes, are extensively used in computer vision and machine learning both for their commercial value as well as for their online availability. For the experiments with faces, Ms-Celeb consists the training set, while for the natural scenes ImageNet.

Error metrics In the comparisons of RoCGAN against cGAN the following metrics are used:

- *Structural similarity (SSIM)* (Wang et al. 2004): A metric used to quantify the perceived image quality of an image. We use it to compare every output image with respect to the reference (ground-truth) image; it ranges from [0, 1] with higher values demonstrating better quality.
- *Frechet inception distance (FID)* Heusel et al. (2017): A measure for the quality of the generated images, frequently used with GAN. It extracts second order information from a pretrained classifier[11] applied to the images. FID assumes that the two distributions p_1 and p_2 are multivariate Gaussian, i.e. $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{C}_1)$ and $\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{C}_2)$. Then:

$$FID(p_1, p_2) = ||\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2||_2^2 + Tr(\boldsymbol{C}_1 + \boldsymbol{C}_2 - 2(\boldsymbol{C}_1\boldsymbol{C}_2)^{\frac{1}{2}}) \tag{21}$$

In our work p_1 is the distribution of the ground-truth images, while p_2 is the distribution of the generated images from each method. FID is lower bounded (by 0) in the case that p_2 matches p_1; a lower FID score translates to the distributions being 'closer'. We compute the FID score using the Inception network (in Chainer).

3.4 Ablation Study

In the following paragraphs we conduct an ablation study to assess RoCGAN in different cases; specifically we evaluate the sensitivity in a hyper-parameter range and different initialization options. We also summarize different options for loss functions and other architecture-related choices.

Unless mentioned otherwise, '5layer' network is used; the task selected is face super-resolution while SSIM is reported as a metric in this part. The options selected in the ablation study are used in the following experiments and comparisons against cGAN.

3.4.1 Initialization of RoCGAN

We conduct an experiment to evaluate different initialization options for RoCGAN. The motivation for the different initializations is to assess the necessity of the pretrained models as used in Chrysos et al. (2019b). The options are:

- Random initialization for all modules.
- Initializing the $\boldsymbol{e}^{(AE)}$ to the pretrained weights of the respective AAE encoder and the rest modules from the pretrained cGAN.
- Initializing only the unsupervised pathway from the respective pretrained generator of AAE. The rest modules are initialized randomly.

The results in Table 1 demonstrate that the initializations are not crucial for the final performance, however the second option performs slightly worse. We postulate that the pretrained cGAN makes RoCGAN get stuck 'near' the cGAN optimum. In the remaining experiments, we use the third option, i.e. we initialize the unsupervised pathway from the

[11] Typically the features from the last layer of the pretrained Inception CNN are used.

Table 1 Quantitative results evaluating the different initialization options (Sect. 3.4.1)

Initialization	RI	PRETR	AE
SSIM	0.830	0.812	0.830
FID	58.2	51.2	57.4

The abbreviations 'RI', 'PRETR', 'AE' stand for the three options of (i) random initialization, (ii) pretrained models, (iii) only pretrained AE pathway. Note that all RI and AE initializations are equivalent in terms of SSIM, while the PRETR is worse. Therefore, we can select either RI or AE for initializing RoCGAN

Table 2 Validation of λ_l hyper-parameter in the '5layer' network

λ_l	0.1	1	5	10	20	50	100
SSIM	0.825	0.830	0.830	0.829	0.829	0.828	0.828
FID	58.9	57.4	55.5	56.1	56.2	56.4	56.1

The final SSIM values do not vary much for λ_l in a wide range, which indicates that our model is robust to λ_l choices

respective AAE generator while the rest modules are initialized randomly.

3.4.2 Hyper-Parameter Range

Our model introduces two new loss terms, i.e. $\mathcal{L}_{lat}$ and $\mathcal{L}_{AE}$, that need to be validated. Below, we scrutinize one hyper-parameter every time, while we keep the rest in their selected value. During our experimentation, we observed that the optimal values of these hyper-parameters might differ per case/network, however unless we mention it explicitly in an experiment *the hyper-parameters remain the same as aforementioned*.

The search space for each term is decided from its theoretical properties and our intuition. In more details, λ_{ae} would have a value at most equal to λ_c.[12] The latent loss encourages the two pathways' latent representations to be similar, however since the final evaluation is performed in the pixel space, we postulate that a value smaller than λ_c is appropriate.

In Table 2, different values for the λ_l are presented. The optimal values emerge in the interval [1, 10), however even for the rest choices the SSIM values are similar. In our experimentation, RoCGAN is more resilient to changes in λ_l than other hyper-parameters.

Different values of λ_{ae} are considered in Table 3. RoCGAN are robust to a wide range of values and both the visual and the quantitative results remain similar. In the following experiments we use $\lambda_{ae} = \lambda_c = 100$ because of the semantic similarity with the content loss; further improvements can be obtained by the best validation values.

[12] To fairly compare with baseline cGAN, we use the same value as in Isola et al. (2017).

Table 3 Validation of λ_{ae} values (hyper-parameter choices) in the '5layer' network

λ_{ae}	1	5	10	50	100	150	200
SSIM	0.834	0.834	0.834	0.832	0.830	0.829	0.828
FID	52.8	54.2	53.5	58.3	57.4	59.5	59.8

The network remains robust for a wide range of values of the hyper-parameter λ_{ae}. The best performance is obtained for lower values of λ_{ae}, i.e. $\lambda_{ae} < 50$, however in our evaluation we use $\lambda_{ae} = \lambda_c = 100$ for the semantic meaning. For further improvements one of the rest values or even further search might result in better hyper-parameter values

Table 4 Quantitative results on the discriminator variants (see Sect. 3.4.3)

Discriminator options	Default	Concat	Proj
SSIM	0.830	0.829	0.827
FID	57.4	59.2	60.1

The 'Concat' abbreviates the concatenation of Isola et al. (2017), while 'Proj' abbreviates the projective discriminator of Miyato and Koyama (2018). All three discriminators result in a similar performance, with the projective discriminator resulting in a marginal deterioration in the score. However, we believe that for larger networks, there might be indeed difference in the performance

3.4.3 Robustness on Discriminator Variants

Since the advent of cGAN, several discriminator architectures have been used. In the original paper, the discriminator accepts as input only the output of the generator or a sample from the target distribution. By contrast, Isola et al. (2017), propose to instead concatenate the source and the target images. Miyato and Koyama (2018) argue that instead of concatenation, the inner product of the source and the target image should be computed.

We assess the robustness of RoCGAN under these different discriminators. As a reminder, we consider the discriminator of Mirza and Osindero (2014) as the default; to implement the variants of Isola et al. (2017) and Miyato and Koyama (2018), we do not change the number of depth of the layers, but only perform the respective concatenation, projection respectively.

In Table 4 the evaluation demonstrates that all three discriminators perform similarly. There is a marginal performance drop in the case of the projective discriminator, but this could be mitigated with a stronger generator for example. This experiment demonstrates that the proposed RoCGAN is not tied to a single discriminator, but rather can work with a number of discriminator architectures.

3.4.4 Other Training Options

We evaluate two more options for training our model: (a) whether the improvement can be obtained without batch nor-

Table 5 Quantitative results evaluating training options (Sect. 3.4.4)

Training options	Default	ℓ_2	No BN
SSIM	0.830	0.831	0.830
FID	57.4	58.6	55.0

The two options (along withe the 'Default') are (a) to use an ℓ_2 loss for $\mathcal{L}_{lat}$ and (b) to remove batch normalization from the generator pathways. In both cases the performance remains the same

Table 6 Quantitative comparison of cGAN/RoCGAN (Sect. 3.5)

Experiment	Faces		Scenes	
Method	*SSIM*	*FID*	*SSIM*	*FID*
Baseline-5layer	0.791	67.7	0.539	156.1
Ours-5layer	0.830	57.4	0.552	128.9
AAE	0.903	29.0	0.723	68.0

In both scenes and faces datasets RoCGAN verifies our intuition and outperforms the baseline

Table 7 Quantitative comparison of cGAN/RoCGAN for the case of skip connections (Sect. 3.5)

Metric Method	SSIM	FID
Baseline-5layer-skip	0.843	50.0
Ours-5layer-skip	0.857	47.3

The task is face super-resolution and the results are similar to the networks without skip connections

malization, (b) a different latent loss function (ℓ_2). In Table 5 we add the two options along with the default options from above. The results indicate that (i) batch normalization does not seem to contribute in RoCGAN's performance in this network, (ii) our choice of ℓ_1 can be replaced from another function with similar results. In the rest of the experiments, we use batch normalization and ℓ_1 for $\mathcal{L}_{lat}$.

3.5 Testing on Static Images

Our first evaluation against baseline cGAN is on testing without any additional noise (other than the implicit biases of the datasets). The task for both the faces and the scenes is super-resolution in the respective domain. The training images are from Ms-Celeb and ImageNet respectively, while the testing images from Celeb-A and ImageNet testset. The numerical results in Table 6 dictate that in both cases and with both metrics, RoCGAN outperform cGAN. We also experiment with the '5layer-skip' networks to assess the performance in the skip case. The results in Table 7 illustrate similar behavior to the previous case, i.e. our model outperforms the baseline.

3.6 Testing Under Additional Noise

We conduct a dedicated experiment to evaluate the resilience of the models to noise. The idea is to artificially corrupt the source signal $\boldsymbol{s}$ with the noise models of Sect. 3.1, i.e. feed as input $\boldsymbol{s} + \boldsymbol{f}(\boldsymbol{s}, \boldsymbol{G})$ for some corruption function $\boldsymbol{f}$.

We use the '5layer' networks in the face super-resolution task and corrupt them with (a) adversarial and (b) Bernoulli noise.

Bernoulli noise As a reminder the noise in this experiment is used exclusively during testing. All three cases of $(1, 0, 0), (0, 1, 0), (0, 0, 1)$ are assessed[13], along with mixed cases. The quantitative results for Bernoulli noise are reported Table 8. Our model is consistently better with a relative performance gain of up to 9.9%. Indicative visual results are depicted in Fig. 4.

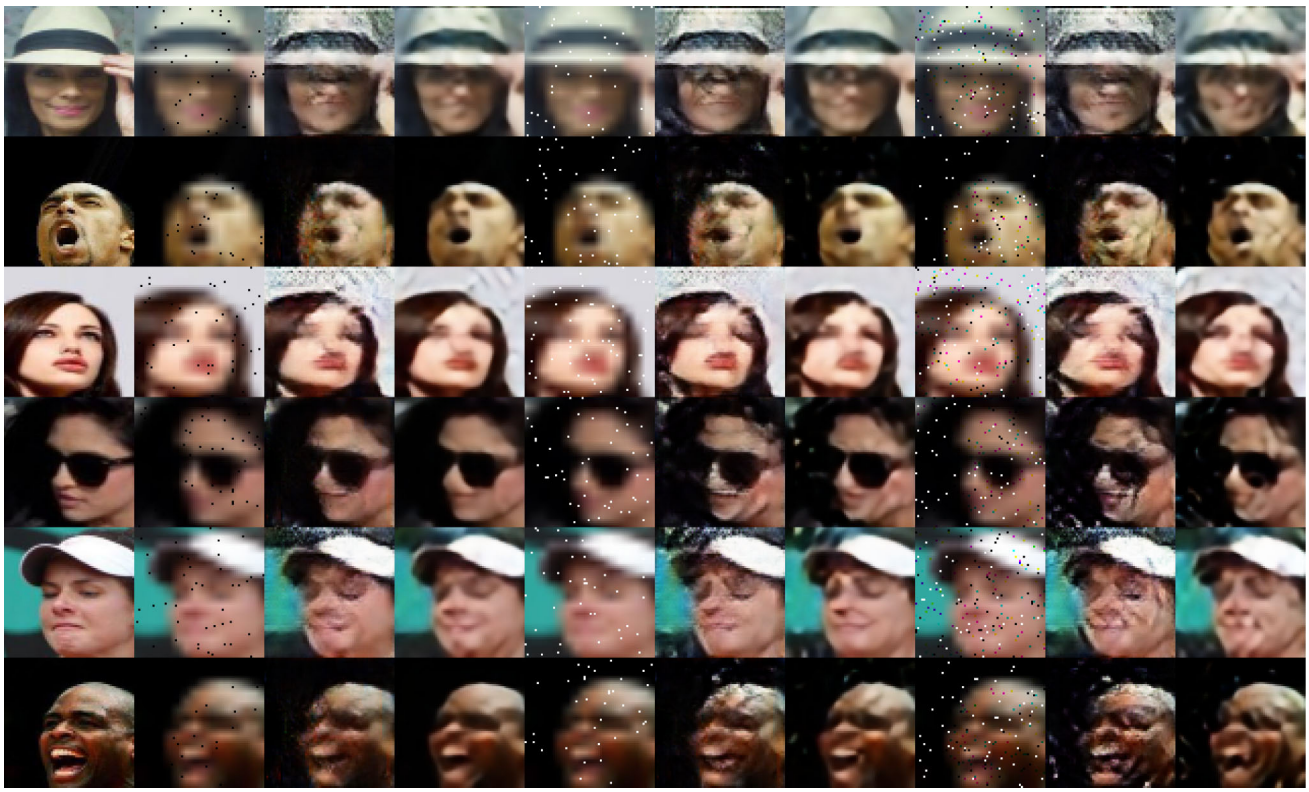
Adversarial noise The performance under the three different adversarial attacks is assessed. For *IFGSM*, we initially start with a small value of ϵ, i.e. $\epsilon = 0.01$, and progressively increase either the steps or the hyper-parameter's value. As expected, the results in Table 9 highlight that increasing values of either the steps or ϵ deteriorate the performance of the networks. However, the performance of cGAN decline with a faster pace when compared to our proposed RoCGAN. The relative performance difference (in SSIM) is 4.9% in the original testing, while it progressively grows up to 24.3% in the $(1, 0.1)$ noise. The effect of the steps in IFGSM is further explored in Fig. 5. We fix $\epsilon = 0.01$ and study the evolution in performance as we vary number of steps. Note that the curve of cGAN is much steeper than that of RoCGAN as the number of steps increase. Beyond 10 steps, the performance of cGAN drops below 0.5 and can essentially be considered as noise. We perform the same experiment with the PGD attack; the effect of the increasing steps are visualized in Fig. 6. We note that after 10 steps there is substantial difference between the two models. This difference is maintained and increased if we increase the steps to 30. We also compare the two models under the latent attack in Fig. 7. For 1 or 2 iteration of the latent attack, the curves are similar to the previous two, however for more steps the curves become steeper than in previous attacks, while the performance gap grows faster in this attack. The efficiency of the three attacks differs when it comes to the number of steps required, with the latent attack being the most successful. Remarkably though, all three attacks have similar effects in the two models, i.e. the performance gap increases as the number of steps increase. By implementing three adversarial attacks, we illustrate that empirically the proposed model is more robust in the face of noise against the baseline.

[13] As a reminder (a, b, c) means that with probability $a\%$ a pixel is converted to black; with probability $b\%$ converted to white and with probability $c\%$ converted channel-wise to black.

Table 8 Quantitative evaluation of the '5layer' network under Bernoulli noise (face super-resolution; Sect. 3.6)

Noise type	Bernoulli											
Method	(1, 0, 0)		(5, 0, 0)		(0, 1, 0)		(0, 0, 1)		(0, 0, 5)		(1, 1, 1)	
	SSIM	FID	SSIM	FID	SSIM	FID	SSIM	FID	SSIM	FID	SSIM	FID
Baseline-5layer	0.756	83.8	0.646	155.5	0.709	125.5	0.768	90.9	0.692	132.2	0.658	173.4
Ours-5layer	0.800	71.3	0.709	119.7	0.767	102.0	0.812	74.0	0.752	108.9	0.723	144.2

RoCGAN exhibit improved performance when compared to the baseline in every case; this intensifies as the noise increases



(a) GT **(b)** Corr-(1,0,0) **(c)** cGAN-(1,0,0) **(d)** RoCGAN-(1,0,0) **(e)** Corr-(0,1,0) **(f)** cGAN-(0,1,0) **(g)** RoCGAN-(0,1,0) **(h)** Corr-(1,1,1) **(i)** cGAN-(1,1,1) **(j)** RoCGAN-(1,1,1)

Fig. 4 Visual results depicting Bernoulli noise. Similarly to Fig. 10 different samples are visualized per row. The corrupted images are visualized in the original size to make the additional noise more visible. The compared methods have to perform denoising in addition to the translation they are trained on

Table 9 Quantitative evaluation of the '5layer' network under adversarial noise (face super-resolution; Sect. 3.6)

Noise type	No noise		Adversarial									
Method			(1, 0.01)		(2, 0.01)		(5, 0.01)		(1, 0.05)		(1, 0.10)	
	SSIM	FID	SSIM	FID	SSIM	FID	SSIM	FID	SSIM	FID	SSIM	FID
Baseline-5layer	0.791	67.7	0.785	70.8	0.773	76.4	0.705	97.2	0.679	107.8	0.555	190.3
Ours-5layer	0.830	57.4	0.828	58.8	0.822	61.0	0.800	69.3	0.781	74.5	0.690	101.8
AAE	0.903	29.0	0.902	28.8	0.901	28.6	0.891	28.0	0.890	28.0	0.862	27.6

The no-noise refers to original testing, while the rest columns from left to right include progressively increasing amount of noise. It is noticeable that the difference in performance between cGAN and RoCGAN is increasing in both metrics

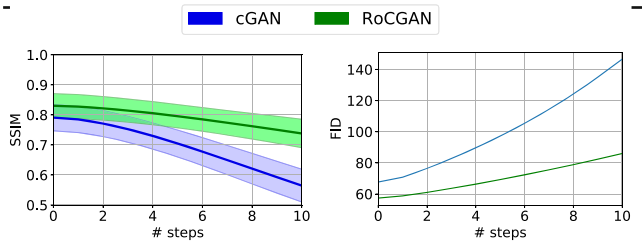


Fig. 5 Performance of cGAN/RoCGAN with respect to the number of steps in the IFGSM noise (mean SSIM on the left, FID score on the right). We emphasize that a higher SSIM (or a lower FID) indicates better performance. The number of steps vary from 1 to 10, while the highlighted region denotes the variance (left). The cGANmodel exhibits steeper curve over RoCGAN

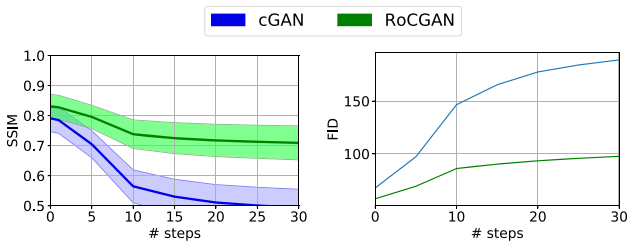


Fig. 6 Performance of cGAN/RoCGAN with respect to the number of steps in the PGD noise (mean SSIM on the left, FID score on the right). In contrast to the IFGSM noise, we plot every 5 steps, since more steps are required in this case. Similarly to the IFGSM in Fig. 5, the baseline (cGAN) exhibits steeper curve over RoCGAN

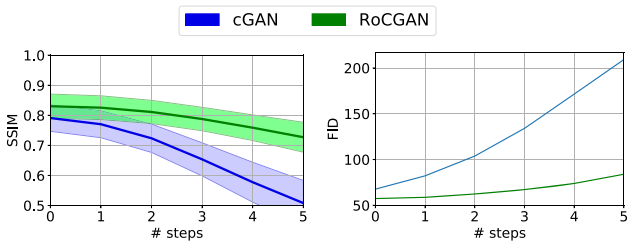


Fig. 7 Performance of cGAN/RoCGAN with respect to the number of steps in the latent attack (mean SSIM on the left, FID score on the right). The performance drop of cGAN model is steeper than the RoCGAN, however notice that this attack is more successful

To further analyze the differences between the two models, we create a histogram plot based on the SSIM values. The interval of [0.5, 0.95] that the SSIM values lie is divided in 20 bins, while the vertical axis depicts the frequency of each bin. A histogram with values concentrated to the right (towards 1) signifies superior performance. The histograms comparing '5layer' cGAN/RoCGAN under IFGSM (adversarial noise) are plotted in Fig. 8 (respectively for the Bernoulli noise, the histograms are in Fig. 9). We note that there is an increasing difference between the original histogram (no noise) and the increasing steps of IFGSM, e.g. Fig. 8a versus Fig. 8d. The same difference is observed as ϵ increases; in the extreme case of $\epsilon = 0.1$ there is only minor overlap between the two methods. In Fig. 10, qualitative results demonstrating the adversarial noise are depicted.

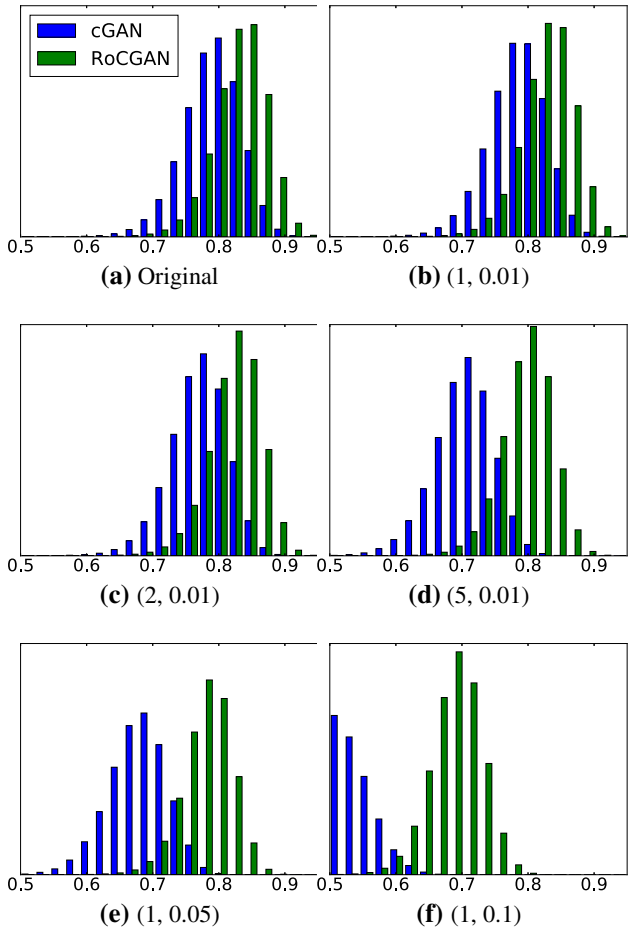


Fig. 8 Histogram plots for the SSIM under adversarial noise (Sect. 3.6). The two distributions differ in the original testing, however the difference increases dramatically for more intense noise

3.7 Testing on a Video Sequence

Aside of the experiment with the static testset, we use the 300VW (Shen et al. 2015) video dataset to assess RoCGAN. The videos include non-linear corruptions, e.g. compression, blurriness, rapid motion; such corruptions make a video dataset the perfect testbed for our evaluation.

In Table 10 we add the results of the experiment.[14] The performance of cGAN is slightly worse than the related experiment in Celeb-A, while RoCGAN's performance remains similar to the static case. The difference in the performance

[14] The most challenging Category3 is selected for the experiment; the other two categories include almost semi-frontal videos as mentioned in Chrysos and Zafeiriou (2017).

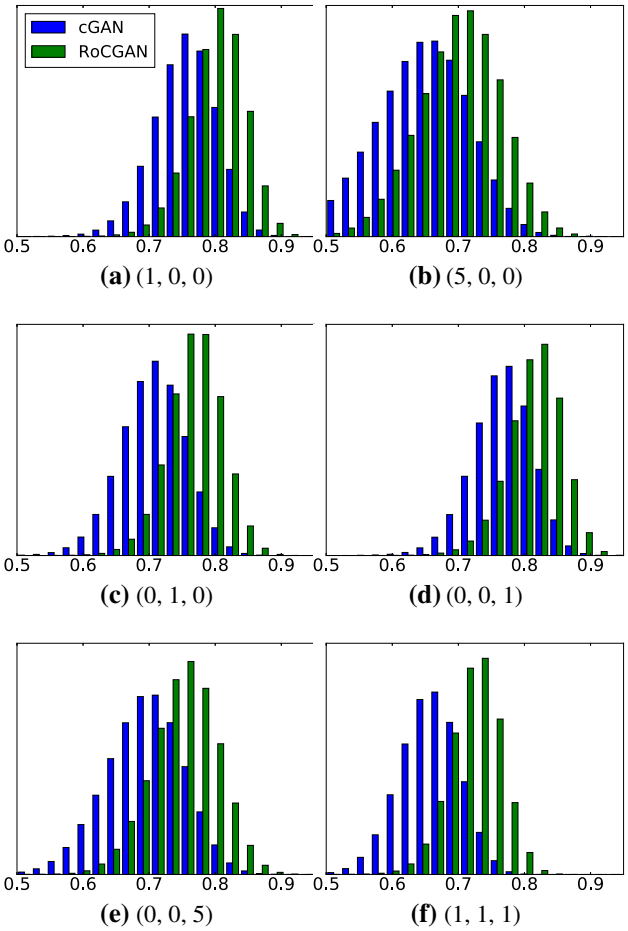


Fig. 9 Histogram plots for the SSIM under Bernoulli noise (Sect. 3.6)

increases in the additional noise cases. The FID performance differs from the respective static experiment, since the mean and covariance for the empirical target distribution are extracted from Celeb-A in both cases. We provide a video of the results in https://youtu.be/RvoW4AYnzQU.

3.8 Cross-Noise Experiments

A reasonable question is whether data augmentation can be used to make the model robust. In our particular setup, we scrutinize this assumption below: we augment the training samples with noise and assess the testing performance. Specifically, we scrutinize the performance of cGAN/RoCGAN with cross-noise experiments, i.e. we train with one type of noise and test with a different type of noise. For a fair comparison with the aforementioned experiments, we keep the same architectures as above, i.e. the '5layer' network, while the task is face super-resolution.

The first experiment is conducted by training with Bernoulli noise; while during testing, adversarial perturbations (IFGSM) are used. The Bernoulli noise (during training) is (5, 0, 0); the variants (10, 0, 0) and $(\theta, 0, 0)$ with θ uniformly sampled in each iteration from [0, 10] were tried but resulted in similar outcomes. The effect of IFGSM for different steps is plotted in Fig. 11; both models exhibit a small improvement with respect to their counterparts trained without noise in Sect. 3.6. Nevertheless, the RoCGAN outperform substantially the cGAN baseline in the face of increasing IFGSM steps.

An additional experiment is conducted with a completely new type of noise, Gaussian noise, i.e. a type of noise that has not been used previously in any of our models. Each training sample is perturbed with additive Gaussian noise. In every iteration a dense noise mask is sampled online from $\mathcal{N}(0, 10)$ (for pixels in the [0, 255] range). The perturbed input for each method is $\boldsymbol{s} + \mathcal{N}(0, 10)$; see Fig. 12 for a visual illustration. The results when trained with adversarial noise (IFGSM) are visualized in Fig. 13, while the comparison with both Bernoulli and adversarial noise is reported in Table 11. The patterns of previous sections (e.g. Sect. 3.6) emerge under Bernoulli noise, i.e. the more intense the noise the larger the performance gap. For instance, the original difference of 0.041 is converted into a difference of 0.069 with 1% white pixels; this intensifies to 0.073 under the (1, 1, 1) case. The performance of both methods improves when trained with Gaussian noise in under both Bernoulli and adversarial noise during testing. However, the performance gap between the baseline and our model remains similar when we increase the number of steps (IFGSM); see Fig. 13.

4 Conclusion

In this work we study the robustness of conditional GANs in the face of noise. Despite their notorious sensitivity to noise, the topic has so far been relatively understudied. In this paper, we introduced the robust conditional gan (RoCGAN) model, a new conditional GAN capable of leveraging unsupervised data to learn better latent representations. RoCGAN modify the generator into a two-pathway generator. The first pathway (*reg pathway*), performs the

(a) GT **(b)** Corr **(c)** cGAN **(d)** RoCGAN **(e)** cGAN-(5,0.01) **(f)** RoCGAN-(5,0.01) **(g)** cGAN-(1,0.05) **(h)** RoCGAN-(1,0.05)

Fig. 10 Visual results for testing with adversarial noise (IFGSM). The columns correspond to **a** the target images, **b** the original corrupted (i.e. downsampled) images, **c**, **d** the outputs of the no-noise (i.e. images of **b**), **e-h** pairs of cGAN/RoCGAN outputs with adversarial noise (see Sect. 3.1 for the encoding). It is noticeable that as the noise increases cGAN outputs deteriorate fast in contrast to their RoCGAN outputs. Notice the ample differences for intense noise; for instance, in columns **(e)** versus **(f)** where cGAN includes unnatural lines in all cases

Table 10 Quantitative results for the video sequence testing (Sect. 3.7)

Method	*Noise type*											
	No noise		Bernoulli				Adversarial					
			(0, 0, 1)		(1, 1, 1)		(2, 0.01)		(1, 0.05)		(2, 0.05)	
	SSIM	FID	SSIM	FID	SSIM	FID	SSIM	FID	SSIM	FID	SSIM	FID
Baseline-5layer	0.785	192.1	0.770	189.8	0.627	175.7	0.768	189.2	0.676	159.3	0.546	167.9
Ours-5layer	0.848	191.0	0.839	188.1	0.722	150.0	0.843	180.2	0.800	153.1	0.727	155.7

The relative gain (of RoCGAN in SSIM) in the video sequence is 8% (original testset), while it grows up to 33% (intense noise)

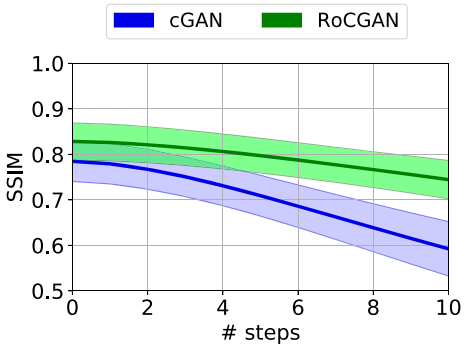


Fig. 11 Performance of a cGAN/RoCGAN (mean SSIM) trained with Bernoulli noise. The *x*-axis depicts an increasing number of iterations of the IFGSM from 1 to 10. The highlighted region in each curve denotes the variance

(a) GT **(b)** Corr **(c)** GNoise **(d)** Corr+GNoise

Fig. 12 Visual example of the training with Gaussian noise (see Sect. 3.8). The ground-truth image is downsampled for the 'Corr' version; Gaussian noise ('GNoise') is sampled and added to the corrupted image; the 'Corr+GNoise' consists the training image for each method

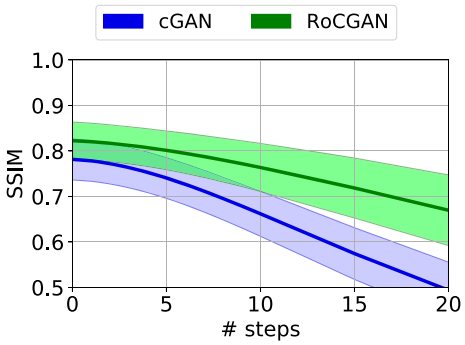


Fig. 13 Performance of cGAN/RoCGAN (mean SSIM) when trained with Gaussian noise. Both models are more robust when trained with Gaussian noise; it requires 15 adversarial steps instead of 10 to achieve the same degradation. Nevertheless, the same pattern with increasing performance gap emerges in the Gaussian noise

regression from the source to the target domain. The new, added pathway (*AE pathway*) is an autoencoder in the target domain. By adding weight sharing between the two decoders, we implicitly constrain the reg pathway to output signals that span the target manifold. We prove that our model shares similar convergence properties with generative adversarial networks. We demonstrated through large scale experiments on images, for both natural scenes and faces, that RoCGAN outperform existing, state-of-the-art conditional GAN models, especially in the face of intense noise. Our model can be used with any form of data and has successfully been applied to sparse inpainting/denoising in Chrysos et al. (2019b) as well as super-resolution. We hope that our work can pave the way towards more robust conditional GANs. Going forward, we aim to study how to merge different types of noise and how to achieve foolproof robustness in a dense regression setting. Additionally, we aim to study how to combine the polynomial networks (Chrysos et al. 2020) with RoCGAN.

Acknowledgements We would like to thank Markos Georgopoulos for our fruitful conversations during the preparation of this work. GG Chrysos would like to thank Amazon web services for the cloud credits. The work of Grigorios Chrysos was partially funded by an Imperial College DTA. The work of Stefanos Zafeiriou was partially funded by the EPSRC Fellowship DEFORM: Large Scale Shape Analysis of Deformable Models of Humans (EP/S010203/1) and a Google Faculty Award.

Table 11 Quantitative evaluation (mean SSIM) of the '5layer' network when trained with Gaussian noise (Sect. 3.8)

Method	*Noise type*								
	No noise	Bernoulli				Adversarial			
		(1, 0, 0)	(0, 1, 0)	(0, 0, 1)	(1, 1, 1)	(1, 0.01)	(2, 0.01)	(5, 0.01)	(10, 0.01)
Baseline-5layer	0.782	0.760	0.713	0.772	0.676	0.778	0.772	0.746	0.662
Ours-5layer	0.823	0.803	0.782	0.815	0.749	0.820	0.817	0.801	0.764

The initial difference of 0.041 is converted into a difference of 0.069 with 1% white pixels; that is RoCGAN increases the performance gap under unseen noise. The same trend is observed in the adversarial (IFGSM) noise

References

Agustsson, E., & Timofte, R. (2017). Ntire 2017 challenge on single image super-resolution: Dataset and study. In *IEEE proceedings of international conference on computer vision and pattern recognition workshops* (CVPR'W) (Vol. 3, p. 2).

Arnab, A., Miksik, O., & Torr, P. H. (2018). On the robustness of semantic segmentation models to adversarial attacks. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)* (pp. 888–897).

Bora, A., Price, E., & Dimakis, A. G. (2018). Ambientgan: Generative models from lossy measurements. *International Conference on Learning Representations (ICLR)*, *2*, 5.

Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., & Erhan, D. (2016). Domain separation networks. In *Advances in neural information processing systems (NIPS)* (pp. 343–351).

Chrysos, G. G., Kossaifi, J., & Zafeiriou, S. (2019b). Robust conditional generative adversarial networks. In *International conference on learning representations (ICLR)*.

Chrysos, G., Moschoglou, S., Bouritsas, G., Panagakis, Y., Deng, J., & Zafeiriou, S. (2020). π−nets: Deep polynomial neural networks. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)*.

Chrysos, G., Favaro, P., & Zafeiriou, S. (2019a). Motion deblurring of faces. *International Journal of Computer Vision (IJCV)*, *127*(6–7), 801–823.

Chrysos, G. G., & Zafeiriou, S. (2017). Pd2t: Person-specific detection, deformable tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, *40*(11), 2555–2568.

Cogswell, M., Ahmed, F., Girshick, R., Zitnick, L., & Batra, D. (2016). Reducing overfitting in deep networks by decorrelating representations. In *International conference on learning representations (ICLR)*.

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)* (pp. 248–255).

Dou, Z., Osher, S. J., & Wang, B. (2018). Mathematical analysis of adversarial attacks. arXiv preprint arXiv:1811.06492.

Georgopoulos, M., Panagakis, Y., & Pantic, M. (2018). Modelling of facial aging and kinship: A survey. *Image and Vision Computing*, *80*, 58–79.

Gondim-Ribeiro, G., Tabacof, P., & Valle, E. (2018). Adversarial attacks on variational autoencoders. arXiv preprint arXiv:1806.04646.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial nets. In *Advances in neural information processing systems (NIPS)*.

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples (2014). In *International conference on learning representations (ICLR)*.

Gretton, A., Borgwardt, K. M., Rasch, M., Schölkopf, B., & Smola, A. J. (2007). A kernel method for the two-sample-problem. In *Advances in neural information processing systems (NIPS)* (pp. 513–520).

Guo, Y., et al. (2016). Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *Proceedings of European conference on computer vision (ECCV)* (pp. 87–102).

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in neural information processing systems (NIPS)* (pp. 6626–6637).

Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, *28*(3/4), 321–377.

Huang, R., Zhang, S., Li, T., He, R., et al. (2017). Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. In *IEEE proceedings of international conference on computer vision (ICCV)*.

Iizuka, S., Simo-Serra, E., & Ishikawa, H. (2017). Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, *36*(4), 107.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning (ICML)*.

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)*.

Johnson, J., Alahi, A., & Fei-Fei, L. (2016). Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of European conference on computer vision (ECCV)* (pp. 694–711).

Kaneko, T., & Harada, T. (2019). Label-noise robust multi-domain image-to-image translation. arXiv preprint arXiv:1905.02185.

Kaneko, T., Ushiku, Y., & Harada, T. (2019). Label-noise robust generative adversarial networks. In *IEEE Proceedings of international conference on computer vision and pattern recognition (CVPR)* (pp. 2467–2476).

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational bayes. In *International conference on learning representations (ICLR)*.

Kos, J., Fischer, I., & Song, D. (2018). Adversarial examples for generative models. In *IEEE security and privacy workshops (SPW)* (pp. 36–42).

Kumar, A., Sattigeri, P., & Fletcher, T. (2017). Semi-supervised learning with GANs: Manifold invariance with improved inference. In *Advances in neural information processing systems (NIPS)* (pp. 5534–5544).

Kurakin, A., Goodfellow, I., & Bengio, S. (2016). Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533.

Kurakin, A., Goodfellow, I., Bengio, S., Dong, Y., Liao, F., Liang, M., et al. (2018). Adversarial attacks and defences competition. arXiv preprint arXiv:1804.00097.

Lamb, A., Binas, J., Goyal, A., Serdyuk, D., Subramanian, S., Mitliagkas, I., et al. (2018). Fortified networks: Improving the robustness of deep networks by modeling the manifold of hidden representations. arXiv preprint arXiv:1804.02485.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., et al. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)*.

Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., et al. (2018). Noise2noise: Learning image restoration without clean data. In *International conference on machine learning (ICML)*.

Li, S. C. X., Jiang, B., & Marlin, B. (2019). MisGAN: Learning from incomplete data with generative adversarial networks. In *International conference on learning representations (ICLR)*.

Liu, M. Y., Breuel, T., & Kautz, J. (2017). Unsupervised image-to-image translation networks. In *Advances in neural information processing systems (NIPS)* (pp. 700–708).

Liu, Z., Luo, P., Wang, X., & Tang, X. (2015). Deep learning face attributes in the wild. In *IEEE proceedings of international conference on computer vision (ICCV)* (pp. 3730–3738).

Ma, L., Jia, X., Sun, Q., Schiele, B., Tuytelaars, T., & Van Gool, L. (2017). Pose guided person image generation. In *Advances in neural information processing systems (NIPS)* (pp. 406–416).

Ma, L., Sun, Q., Georgoulis, S., Van Gool, L., Schiele, B., & Fritz, M. (2018). Disentangled person image generation. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)* (pp. 99–108).

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In *International conference on learning representations (ICLR)*.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., & Frey, B. (2015). Adversarial autoencoders. arXiv preprint arXiv:1511.05644.

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.

Miyato, T., & Koyama, M. (2018). cGANS with projection discriminator. In *International conference on learning representations (ICLR)*.

Murdock, C., Chang, M. F., & Lucey, S. (2018). Deep component analysis via alternating direction neural networks. arXiv preprint arXiv:1803.06407.

Pajot, A., de Bezenac, E., & Gallinari, P. (2019). Unsupervised adversarial image reconstruction. In *International conference on learning representations (ICLR)*.

Panagakis, Y., Nicolaou, M. A., Zafeiriou, S., & Pantic, M. (2016). Robust correlated and individual component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, *38*(8), 1665–1678.

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., & Efros, A. A. (2016). Context encoders: Feature learning by inpainting. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)* (pp. 2536–2544).

Rasmus, A., Berglund, M., Honkala, M., Valpola, H., & Raiko, T. (2015). Semi-supervised learning with ladder networks. In *Advances in neural information processing systems (NIPS)* (pp. 3546–3554).

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training GANs. In *Advances in neural information processing systems (NIPS)* (pp. 2234–2242).

Samangouei, P., Kabkab, M., & Chellappa, R. (2018). Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *International conference on learning representations (ICLR)*.

Shen, J., Zafeiriou, S., Chrysos, G., Kossaifi, J., Tzimiropoulos, G., & Pantic, M. (2015). The first facial landmark tracking in-the-wild challenge: Benchmark and results. In *IEEE proceedings of international conference on computer vision, 300 videos in the wild (300-VW): Facial landmark tracking in-the-wild challenge & workshop (ICCV-W)*.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., et al. (2014). Intriguing properties of neural networks. In *International conference on learning representations (ICLR)*.

Thekumparampil, K. K., Khetan, A., Lin, Z., & Oh, S. (2018). Robustness of conditional GANs to noisy labels. In *Advances in neural information processing systems (NIPS)* (pp. 10271–10282).

Tokui, S., Oono, K., Hido, S., & Clayton, J. (2015). Chainer: A next-generation open source framework for deep learning. In *Proceedings of workshop on machine learning systems (LearningSys) in the twenty-ninth annual conference on neural information processing systems (NIPS)* (Vol. 5, pp. 1–6).

Tran, L., Kossaifi, J., Panagakis, Y., & Pantic, M. (2019). Disentangling geometry and appearance with regularised geometry-aware generative adversarial networks. *IJCV*, *127*(6–7), 824–844.

Tulyakov, S., Liu, M. Y., Yang, X., & Kautz, J. (2018). Mocogan: Decomposing motion and content for video generation. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)*.

Valpola, H. (2015). From neural PCA to deep unsupervised learning. In Advances in independent component analysis and learning machines (pp. 143–171).

Vidal, R., Bruna, J., Giryes, R., & Soatto, S. (2017). Mathematics of deep learning. arXiv preprint arXiv:1712.04741.

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. A. (2008). Extracting and composing robust features with denoising autoencoders. In *International conference on machine learning (ICML)* (pp. 1096–1103).

Wang, Z., Merel, J. S., Reed, S. E., de Freitas, N., Wayne, G., & Heess, N. (2017). Robust imitation of diverse behaviors. In *Advances in neural information processing systems (NIPS)* (pp. 5320–5329).

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions in Image Processing (TIP)*, *13*(4), 600–612.

Wu, X., Xu, K., & Hall, P. (2017). A survey of image synthesis and editing with generative adversarial networks. *Tsinghua Science and Technology*, *22*(6), 660–674.

Yang, C., Lu, X., Lin, Z., Shechtman, E., Wang, O., & Li, H. (2017). High-resolution image inpainting using multi-scale neural patch synthesis. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)*.

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., & Huang, T. S. (2018). Generative image inpainting with contextual attention. In *IEEE Proceedings of international conference on computer vision and pattern recognition (CVPR)*.

Yuan, X., He, P., Zhu, Q., Bhat, R. R., & Li, X. (2017). Adversarial examples: Attacks and defenses for deep learning. arXiv preprint arXiv:1712.07107.

Zhang, Y., Lee, K., & Lee, H. (2016). Augmenting supervised neural networks with unsupervised objectives for large-scale image classification. In *International conference on machine learning (ICML)* (pp. 612–621).

Zhu, J. Y., Zhang, R., Pathak, D., Darrell, T., Efros, A. A., Wang, O., et al. (2017). Toward multimodal image-to-image translation. In *Advances in neural information processing systems (NIPS)* (pp. 465–476).