Robust Conditional Generative Adversarial Networks

Grigorios Chrysos† and Jean Kossaifi† and and Stefanos Zafeiriou†

†*Imperial College London*

Motivation

- Is conditional image generation robust to noise?
- Can we leverage unlabelled data to improve conditional image generation?

Background

- *Regression* is a statistical process for estimating the relationship between the independent (input) variables and the dependent (output) variables.
- Denoting the input domain as $\boldsymbol{S}$ and the output domain $\boldsymbol{Y}$, the regression $\boldsymbol{G}$ is the mapping $\boldsymbol{G} : \boldsymbol{S} \rightarrow \boldsymbol{Y}$.
- For an input signal $\boldsymbol{s}^{(n)} \in \boldsymbol{S}$, layer-wise method parameters $\boldsymbol{W}^i$ and element-wise non-linearity ϕ^i, the regression (with L layers) is expressed as:

$$\boldsymbol{y}^{(n)} = \phi^L(\boldsymbol{W}^L \cdot \ldots \cdot \phi^1(\boldsymbol{W}^1 \cdot \boldsymbol{s}^{(n)})) \quad (1)$$

- In this work, we study regression in the context of conditional image generation. The function $\boldsymbol{G}$ is learned as the generator of a conditional GAN [2, 4].

Conditional GAN

- Conditional GAN [4] consists of two modules, a generator and a discriminator.
- The generator includes an encoder and a decoder that perform the mapping from the input to the output signal. This is visually depicted as:

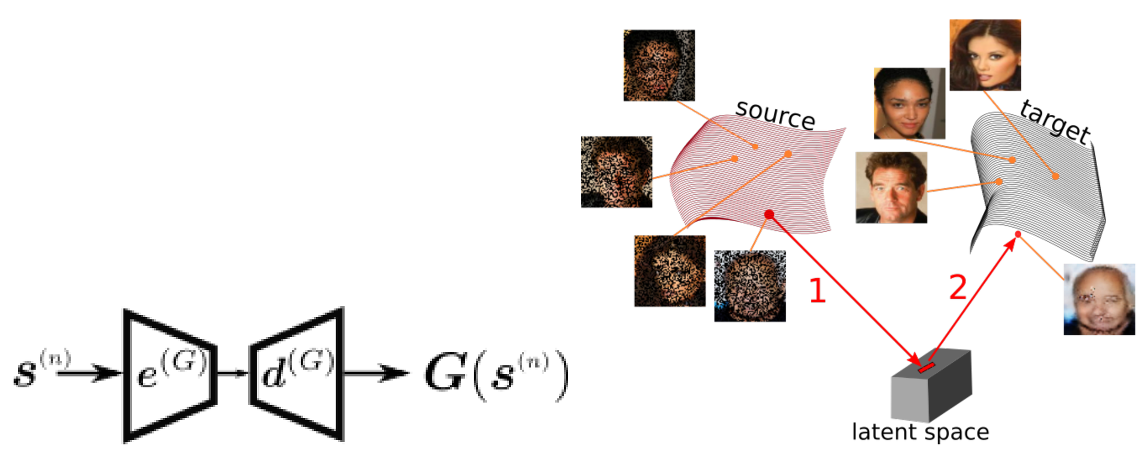

Fig. 1: cGAN generator (left), regression on data space (right).

- Adversarial loss term [2]:

$$\mathcal{L}_{adv} = \log \boldsymbol{D}(\boldsymbol{y}^{(n)}|\boldsymbol{s}^{(n)}) + \log(1 - \boldsymbol{D}(\boldsymbol{G}(\boldsymbol{s}^{(n)})|\boldsymbol{s}^{(n)})) \quad (2)$$

where $\boldsymbol{D}$ denotes the discriminator.

- Frequently, additional (regularization) loss terms are considered. In our work, we include an ℓ_1 content loss [3] and the feature matching loss [5]. The two aforementioned losses are abbreviated as $\mathcal{L}_c$ and $\mathcal{L}_f$. The objective function is formulated, with hyper-parameters λ_* as:

$$\mathcal{L}_{cGAN} = \mathcal{L}_{adv} + \mathcal{L}_{adv} + \lambda_c \cdot \mathcal{L}_c + \lambda_\pi \cdot \mathcal{L}_f \quad (3)$$

Contribution

We propose a new model, coined *RoCGAN*. Our model includes an augmented generator with two pathways.

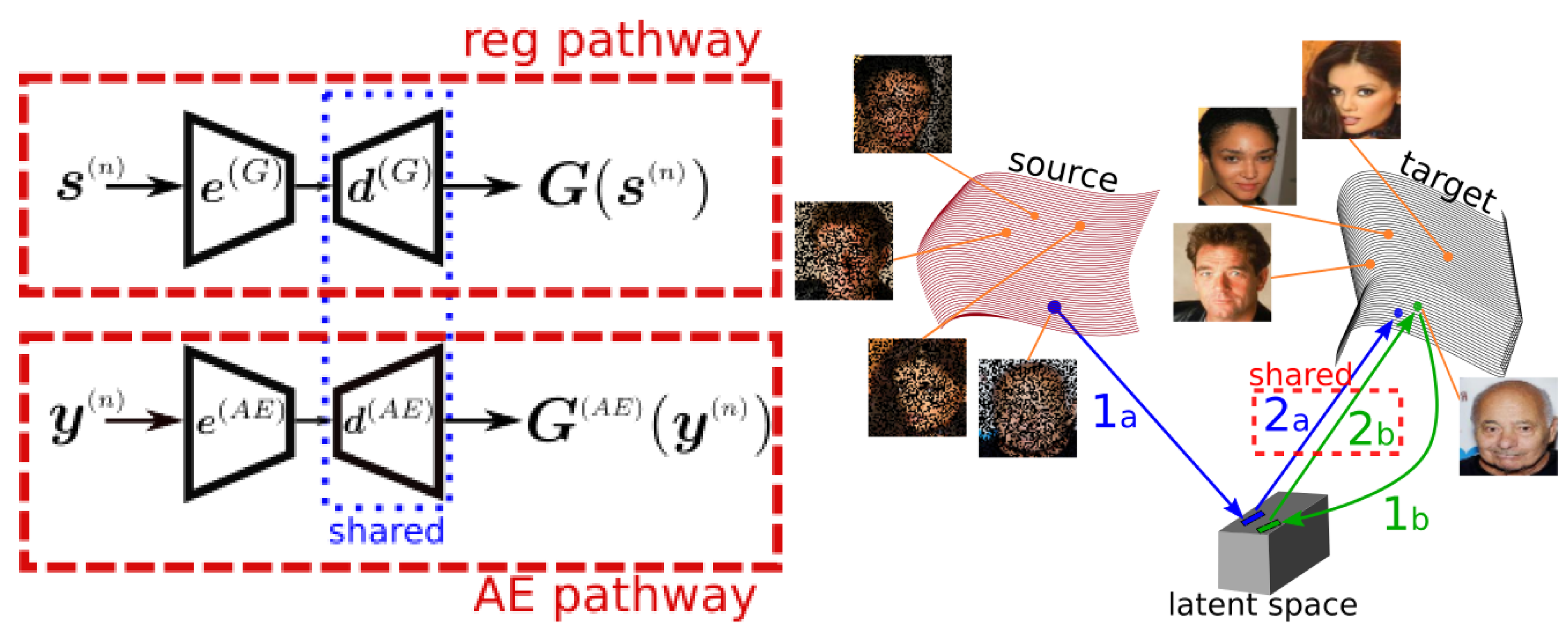

Fig. 2: RoCGAN generator (left), regression on data space (right).

Any cGAN can be modified to RoCGAN format with the following three modifications:

1. Add the new pathway, coined AE pathway. This works as an autoencoder in the output space.
2. Share the decoders' weights in the two pathways.
3. Add the regularization losses for the pathways.

The new loss terms are:

- A content loss in the AE pathway, $\mathcal{L}_{AE}$; we use a loss similar to $\mathcal{L}_c$.
- A loss in the latent representations, $\mathcal{L}_{lat}$; we use an ℓ_1 loss in the encoders' outputs.

The final objective function is:

$$\mathcal{L}_{RoCGAN} = \mathcal{L}_{cGAN} + \lambda_{ae} \cdot \mathcal{L}_{AE} + \lambda_l \cdot \mathcal{L}_{lat} \quad (4)$$

Noise models

To assess the robustness to noise, the following two noise models are used:

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

$$\Phi_v(\boldsymbol{s}, \theta)_{i,j} = \begin{cases} v & \text{with probability } \theta \\ \boldsymbol{s}_{i,j} & \text{with probability } 1 - \theta \end{cases} \quad (5)$$

We use two values for v: $v = 0$, which results in some masked (black) pixels, and channel-wise $v = 0$, which results in pixels masked per channel.

- **Adversarial perturbations**: We extend the FGSM method of [1] for regression:

$$\boldsymbol{u}(\boldsymbol{s}) = \boldsymbol{s} + \epsilon \operatorname{sign}(\nabla_{\boldsymbol{s}} \mathcal{L}(\boldsymbol{s}, \boldsymbol{y})) \quad (6)$$

where ϵ is a hyper-parameter; we use $\boldsymbol{u}(\boldsymbol{s})$ as the perturbed image.

Experiments

Experimental setup:

- The tasks of sparse inpainting (Bernoulli noise $\theta = 0.50$) and denoising ($\theta = 0.25$) are used for training.
- Two different generator architectures are employed. The first one has 4 convolutional layers in each encoder and decoder; the second 5 layers. They are named '4l' and '5l' respectively. We implement cGAN ('Basel') and RoCGAN ('Ours') for each network.
- The quantitative metric is SSIM, which ranges from [0,1] with higher values indicating closer representation to the target image.

Evaluating with same noise as training + additional Bernoulli noise:

Method	Faces				Natural Scenes			
	Denoising		Sparse Inpaint.		Denoising		Sparse Inpaint.	
	25%	35%	50%	75%	25%	35%	50%	75%
Basel-4l	0.803	0.765	0.801	0.701	0.628	0.599	0.639	0.542
Ours-4l	0.834	0.821	0.804	0.708	0.668	0.654	0.648	0.548

In all the experiments RoCGAN *consistently* outperform cGAN.

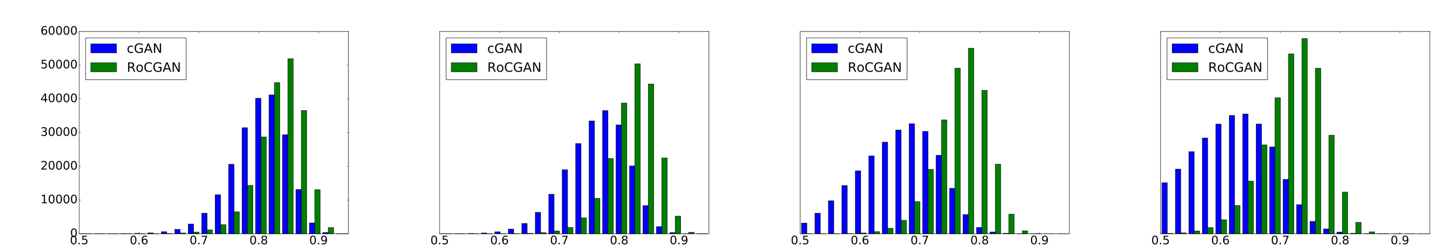

Fig. 3: Histograms of SSIM values for increasing Bernoulli noise. The histogram more to the right is closer to the gt distribution (see paper for further details).

Fig. 4: Qualitative results on faces. Per row: ground-truth images, corrupted images, cGAN outputs, RoCGAN outputs.

References

[1] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples (2014)". In: *ICLR*. 2015.

[2] Ian Goodfellow et al. "Generative adversarial nets". In: *NIPS*. 2014.

[3] Phillip Isola et al. "Image-to-image translation with conditional adversarial networks". In: *CVPR*. 2017.

[4] Mehdi Mirza and Simon Osindero. "Conditional generative adversarial nets". In: *arXiv:1411.1784* (2014).

[5] Tim Salimans et al. "Improved techniques for training gans". In: *NIPS*. 2016, pp. 2234–2242.