In the name of God

Sequence Generative Adversarial Nets (SeqGAN)
And
Objective-Reinforced Generative AdversarialNetworks (ORGAN)

Hossein abbasi

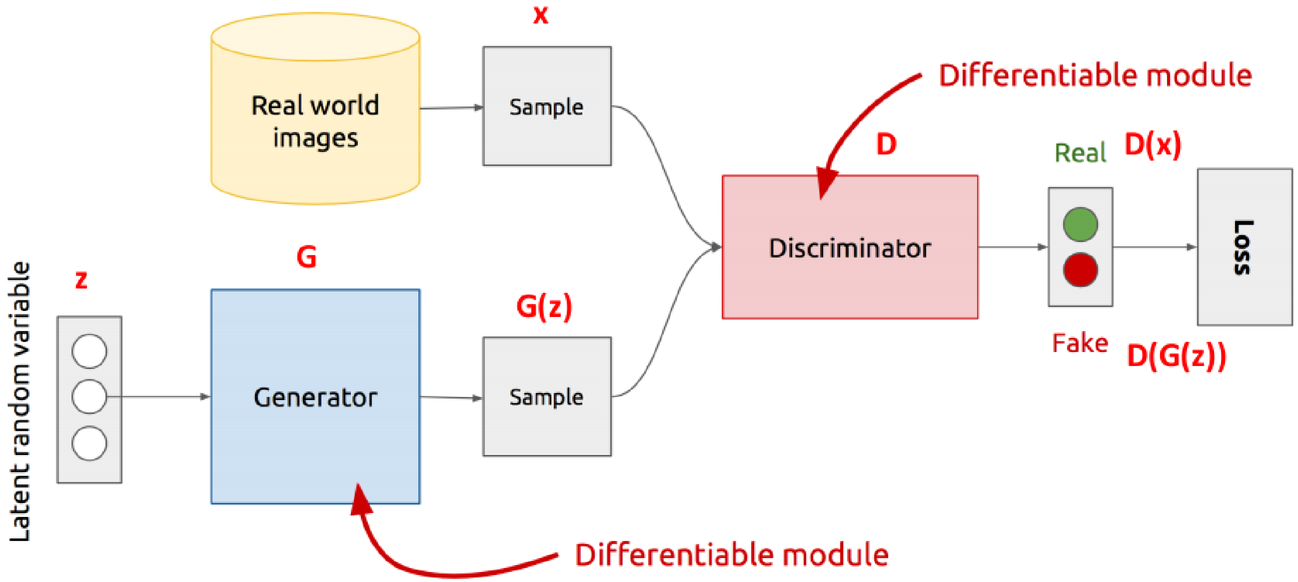
Contents

GANs

Genrative
- Learn a generative Model

Adversarial
- Trained in an adverarial setting

Networks
- Use Deep Neural Networks



What Generative Models?

We've only seen discriminative models so far

- Given an image X, predict a label Y
- Estimates P(Y|X)

Discriminative models have several key limitations

- Can't model P(X), i.e. the probability of seeing a certain image
- Thus, can't sample from P(X), i.e. can't generate new images

Generative models (in general) cope with all of above

- Can model P(X)
- Can generate new images

$$\min_G \max_D V(D, G)$$

$$V(D,G) = \boxed{\mathbb{E}_{x\sim p(x)}[\log D(x)]} + \boxed{\mathbb{E}_{z\sim q(z)}[\log(1 - D(G(z)))]}$$

$$\boxed{D^* = \arg\max_D V(D,G)} \quad \boxed{G^* = \arg\min_G V(D,G)}$$

In this formulation, Discriminator's strategy was $D(x) \rightarrow 1, \;\; D(G(z)) \rightarrow 0$

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

Discriminator updates

for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right].$$

end for

Generator updates

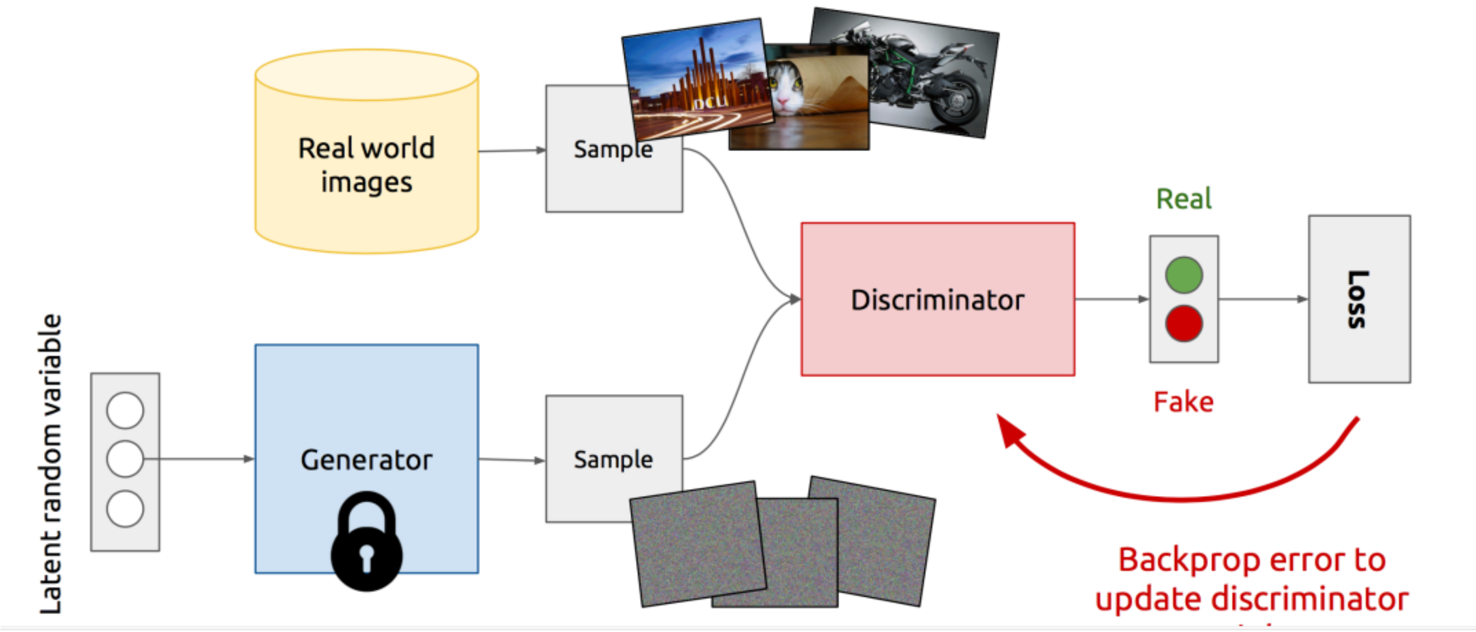
- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

Training GANs

Training Discriminator

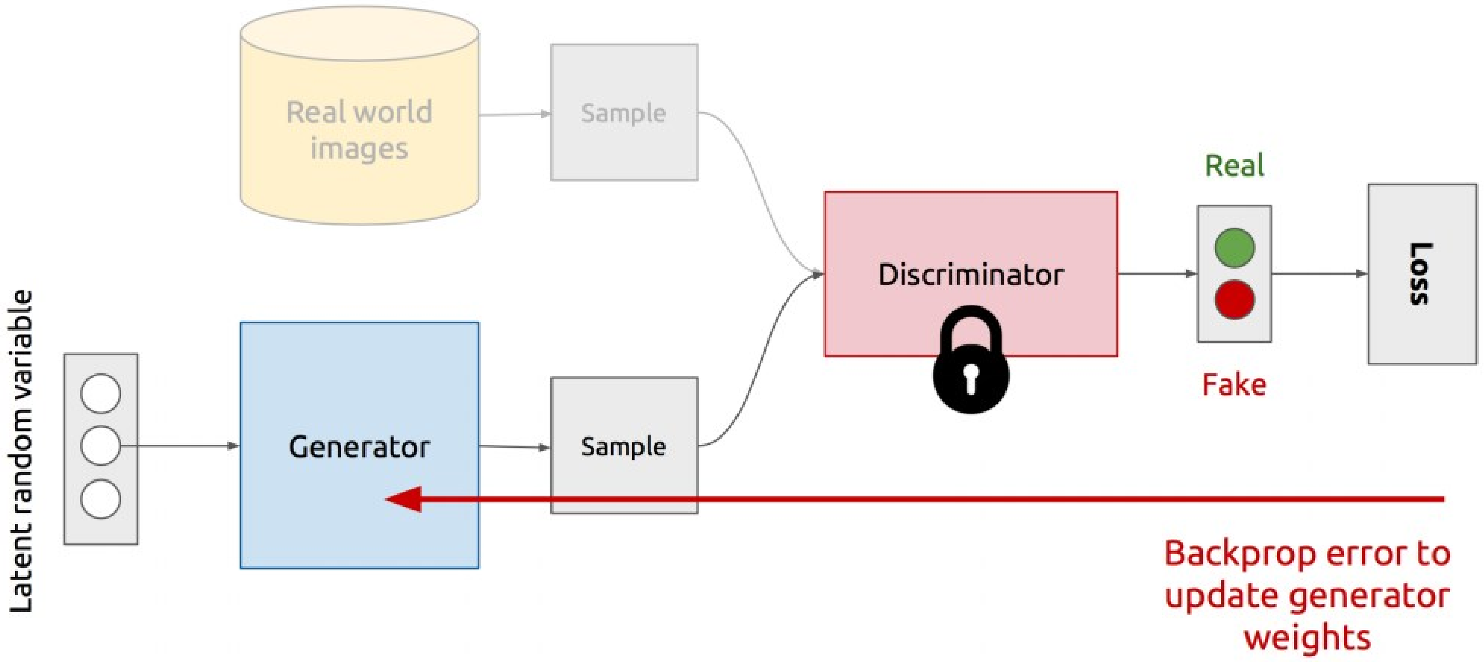
$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right].$$



Training GANs

Training Generator

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$



Applications

- Image-to-Image Translation
- Text-to-Image Synthesis
- Face Aging

…

Image-to-Image Translation

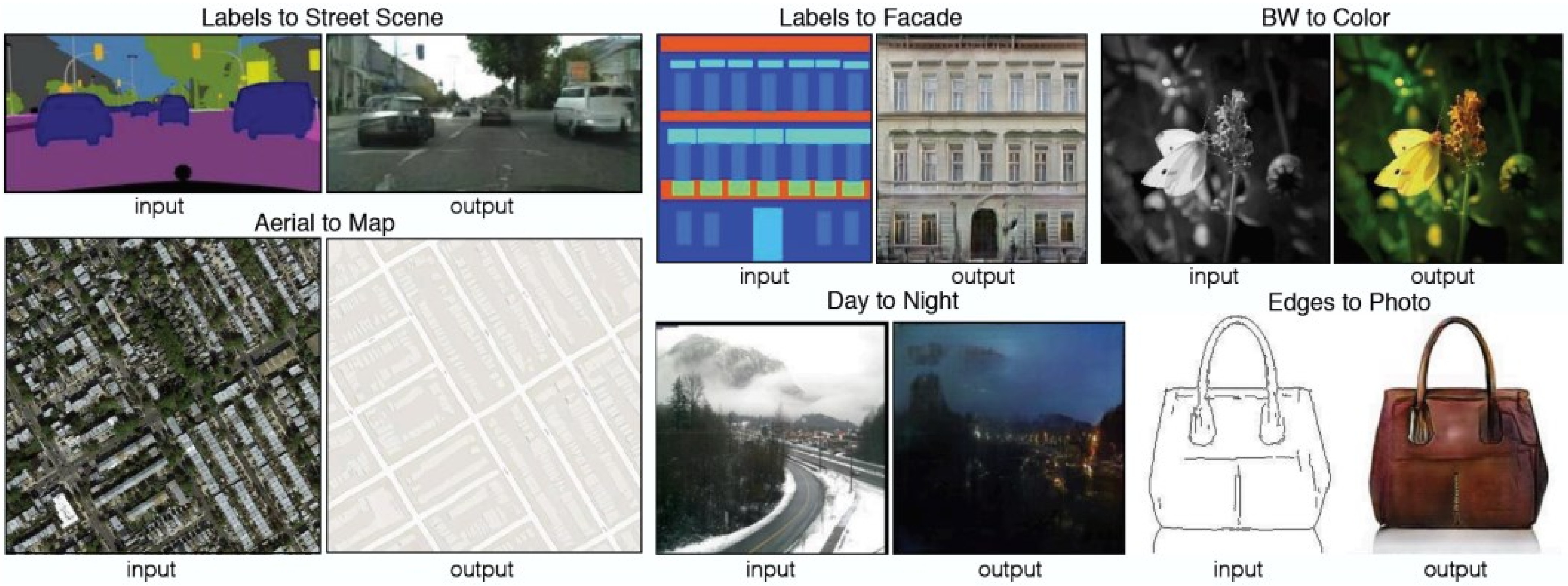


Figure 1 in the original paper.

[Link to an interactive demo of this paper]

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. "**Image-to-image translation with conditional adversarial networks**". arXiv preprint arXiv:1611.07004. (2016).

Text-to-Image Synthesis

Figure 1 in the original paper.

Text-to-Image Synthesis

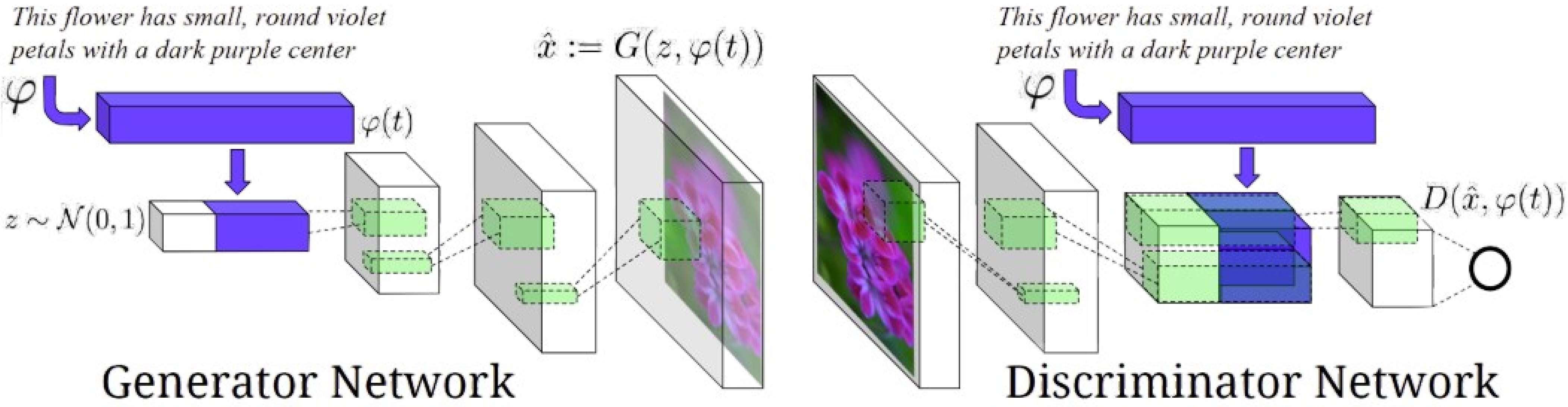


Figure 2 in the original paper.

Positive Example:

Real Image, Right Text

Negative Examples:

Real Image, Wrong Text

Fake Image, Right Text

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., & Lee, H. "**Generative adversarial text to image synthesis**". ICML (2016).

Face Aging with Conditional GANs

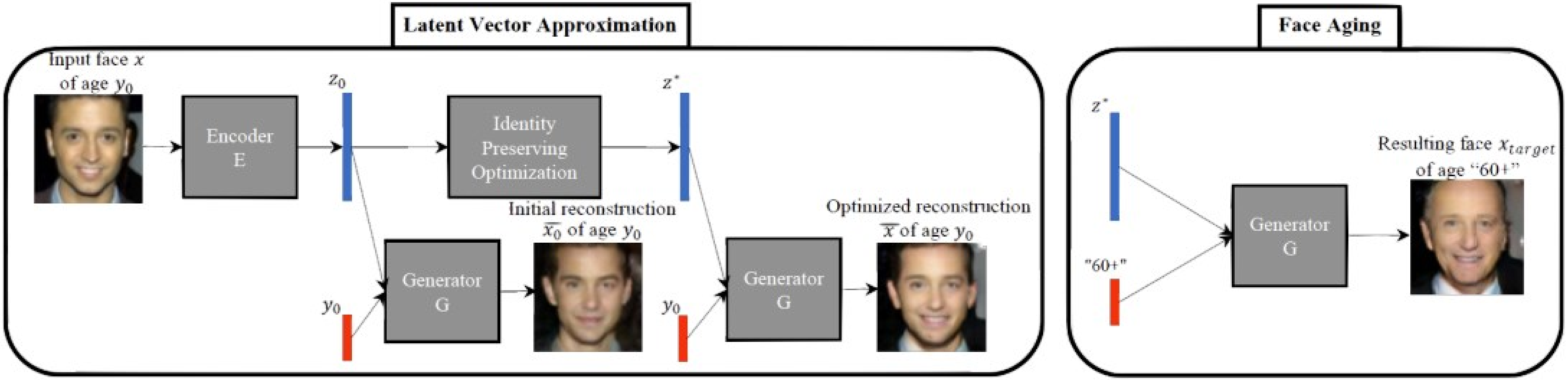
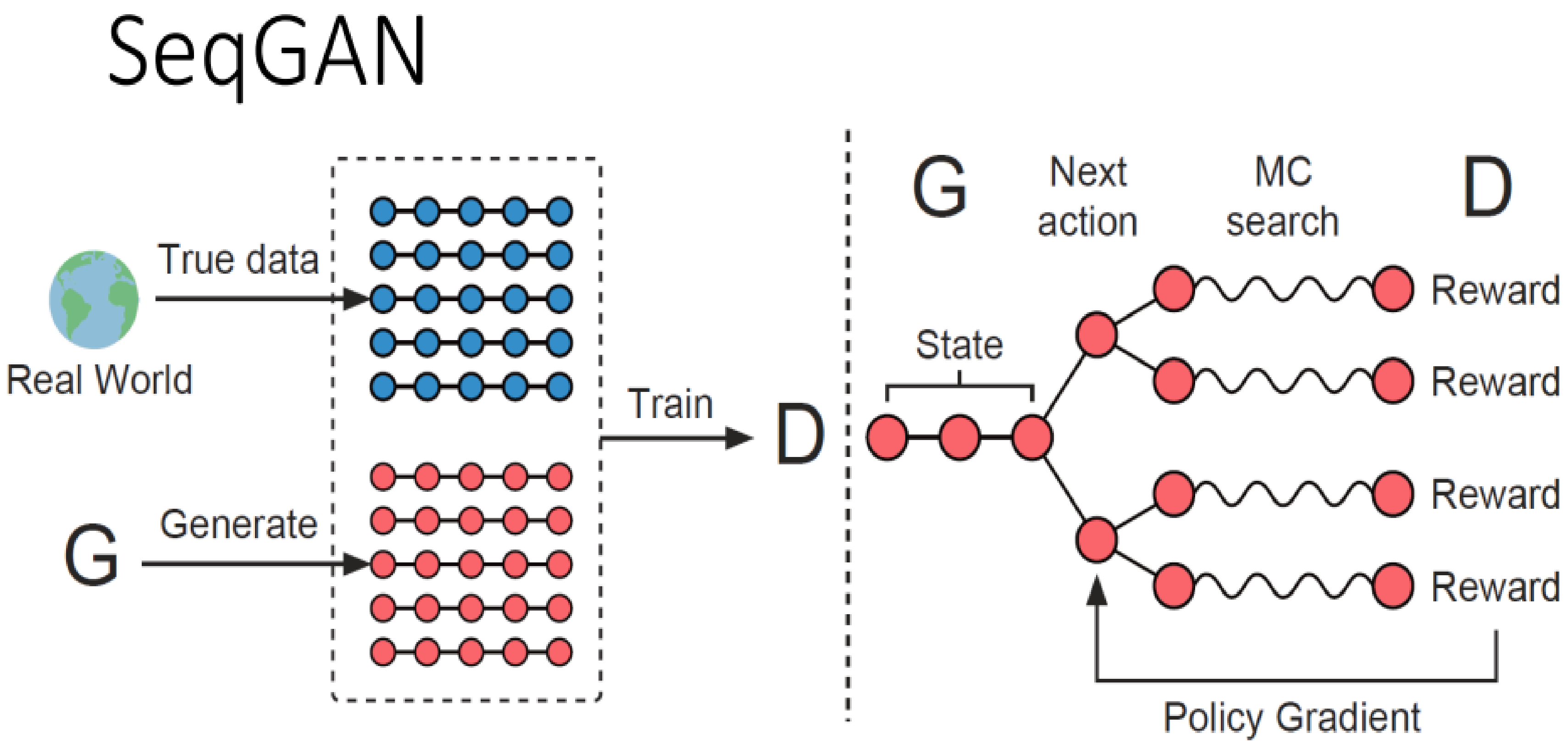


Figure 1 in the original paper.

Antipov, G., Baccouche, M., & Dugelay, J. L. (2017). "**Face Aging With Conditional Generative Adversarial Networks**". arXiv preprint arXiv:1702.01983.

Sequence Generative Adversarial Nets (SeqGAN)



Algorithm 1 Sequence Generative Adversarial Nets

Require: generator policy G_θ; roll-out policy G_β; discriminator D_ϕ; a sequence dataset $\mathcal{S} = \{X_{1:T}\}$

Initialize G_θ, D_ϕ with random weights θ, ϕ.
Pre-train G_θ using MLE on $\mathcal{S}$
$\beta \leftarrow \theta$
Generate negative samples using G_θ for training D_ϕ
Pre-train D_ϕ via minimizing the cross entropy
repeat
    for g-steps **do**
        Generate a sequence $Y_{1:T} = (y_1, \dots, y_T) \sim G_\theta$
        for t in $1 : T$ **do**
            Compute $Q(a = y_t; s = Y_{1:t-1})$ by Eq. (4)
        end for
        Update generator parameters via policy gradient Eq. (8)
    end for
    for d-steps **do**
        Use current G_θ to generate negative examples and combine with given positive examples $\mathcal{S}$
        Train discriminator D_ϕ for k epochs by Eq. (5)
    end for
    $\beta \leftarrow \theta$
until SeqGAN converges

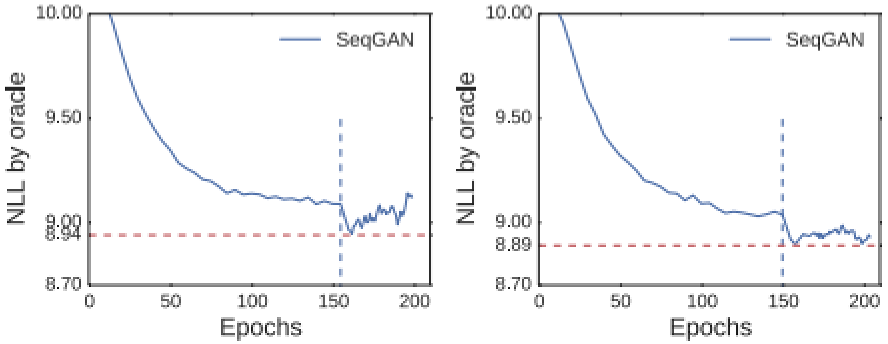
The Discriminative Model for Sequences

Deep discriminative models such as deep neural network (DNN) (Vesely et al ` . 2013),
convolutional neural network (CNN) (Kim 2014)
recurrent convolutional neural network (RCNN) (Lai et al. 2015)

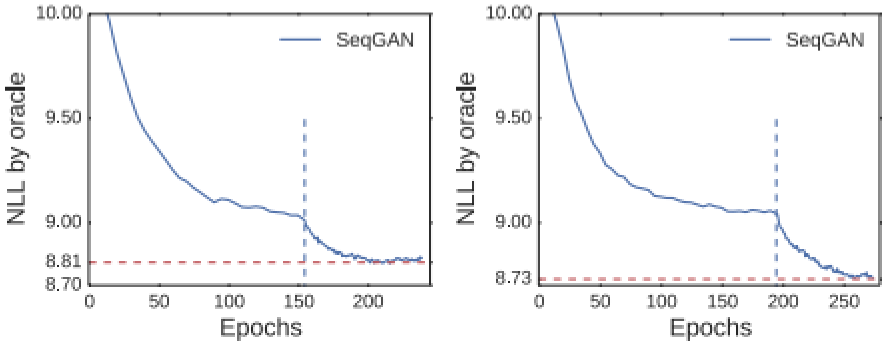
The Generative Model for Sequences

recurrent neural networks (RNNs)

specifically, the g-steps, d-steps and k parameters in Algorithm



(a) g-steps=100, d-steps=1, k=10 (b) g-steps=30, d-steps=1, k=30



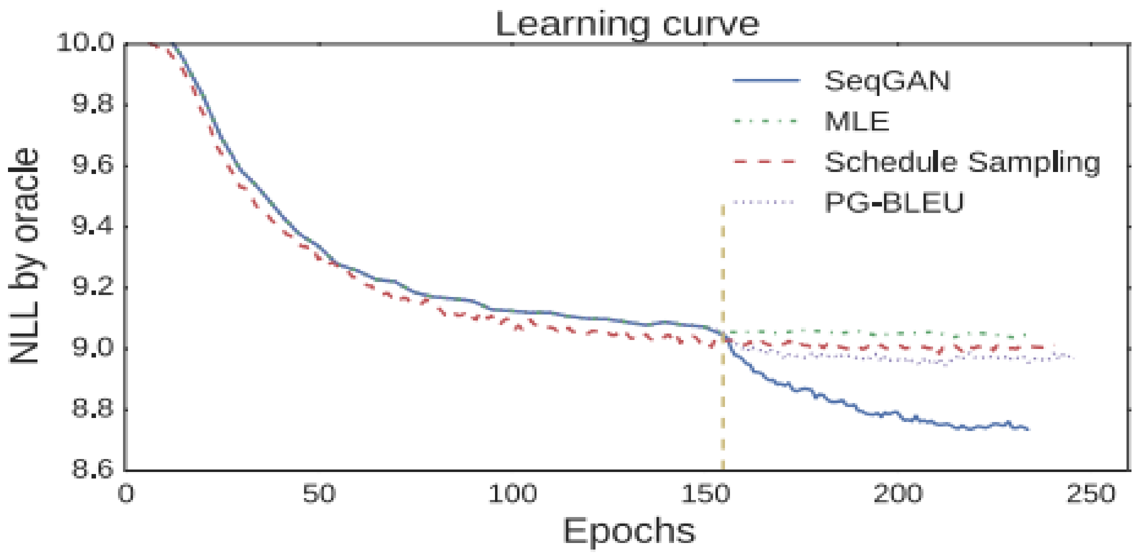
(c) g-steps=1, d-steps=1, k=10 (d) g-steps=1, d-steps=5, k=3

Application

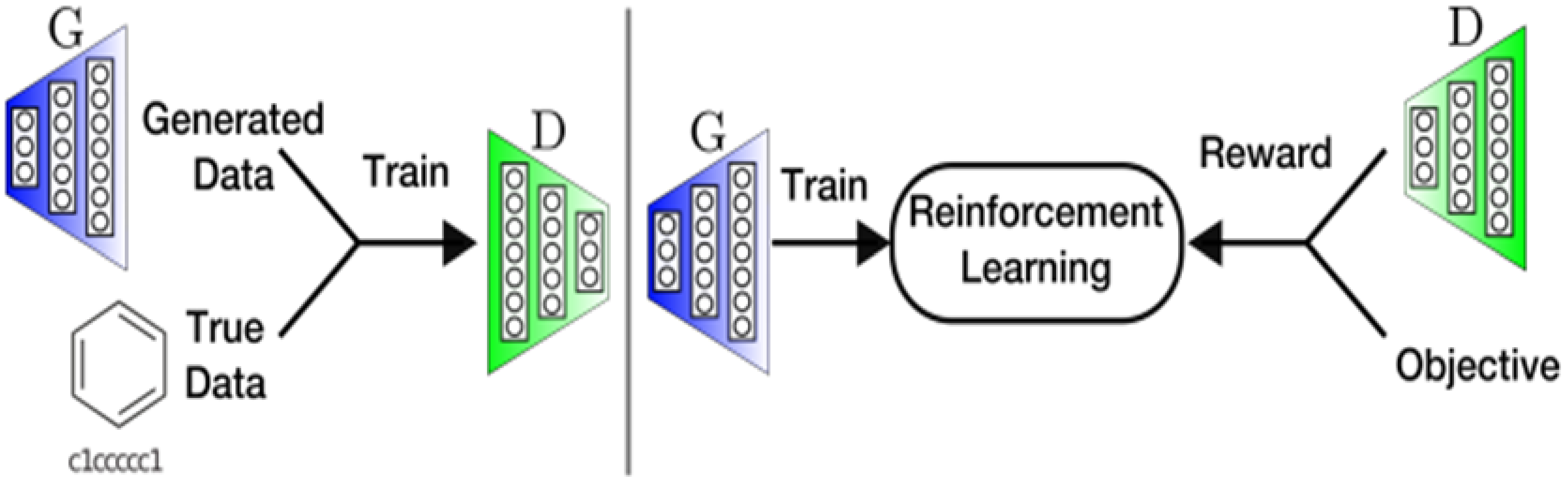
Poems
Spoken language
Generation of music

Table 1: Sequence generation performance comparison. The p-value is between SeqGAN and the baseline from T-test.

Algorithm	Random	MLE	SS	PG-BLEU	SeqGAN
NLL	10.310	9.038	8.985	8.946	**8.736**
p-value	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$	



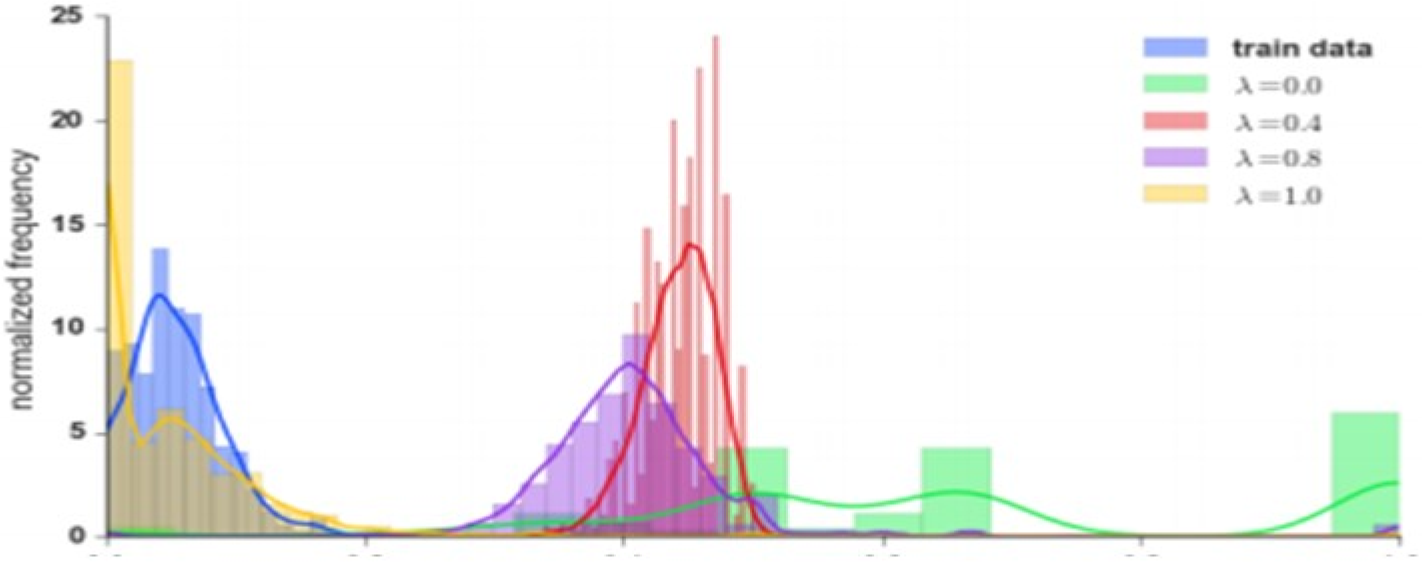
Objective-Reinforced Generative AdversarialNetworks (ORGAN)



Algorithm 1: Objective-Reinforced Generative Adversarial Networks (ORGAN)

require : Generator G_θ; Discriminator D_ϕ; dataset $S = \{Y_{1:T}\}$; objective $O : Y \to (0, 1)$
Initialize G_θ, D_ϕ with random weights θ, ϕ;
Pre-train G_θ using MLE on S;
Generate negative samples using G_θ for training D_ϕ;
Pre-train D_ϕ by minimizing the cross entropy;
repeat
  for *g-steps* **do**
    Generate a sequence $Y_{1:T} = (y_1, ..., y_T) \sim G_\theta$;
    for *t in 1 : T* **do**
      Compute $Q(s = Y_{1:t-1}, a = y_t)$ by Eq. 4 using D_ϕ and O to calculate rewards per Eq. 6
    end
    Update generator parameters θ via policy gradient by Eq 5.;
  end
  for *d-steps* **do**
    Use G_θ to generate negative examples and combine with given positive examples S;
    Train discriminator D_ϕ for k epochs by Eq. 1;
  end
until *model converges*;

RL Parameter



Results of ORGAN optimizing different metrics

Objective	Algorithm	Tonality	Melodicity	Ratio of Steps	Diversity
Tonality	Naive RL	0.629	0.640	0.023	0.086
	ORGAN ($\lambda = 0.6$)	**0.449**	0.452	0.010	**0.274**
	SeqGAN	0.048	0.101	0.317	0.377
	MLE	0.055	0.097	0.280	0.560
Melodicity	Naive RL	0.467	0.629	0.062	0.238
	ORGAN ($\lambda = 0.6$)	0.594	**0.679**	0.033	**0.386**
	SeqGAN	0.057	0.103	0.304	0.535
	MLE	0.055	0.097	0.280	0.560
Ratio of Steps	Naive RL	0.001	0.003	0.901	0.278
	ORGAN ($\lambda = 0.8$)	0.010	0.030	**0.771**	**0.551**
	SeqGAN	0.044	0.084	0.302	0.648
	MLE	0.055	0.097	0.280	0.560

Table 3: Results of ORGAN optimizing different metrics. Each measure was taken over a set of 6400 generated songs. Bolded entries represent the optimized metric as well as the diversity achieved by ORGAN in each experiment.

Result

Finally, future work should extend ORGANs to work with data that is not sequential, such as images.

This requires framing the GAN setup as a reinforcement learning problem in order to add an arbitrary (not necessarily differentiable) objective function.

Reference

1. Yu, Lantao, et al. Seqgan: sequence generative adversarial nets with policy gradient. arXiv preprint arXiv:1609.05473 (2016).
2. Gabriel L. Guimaraes, Benjamin Sanchez-Lengeling, Pedro Luis Cunha Farias, Alán Aspuru-Guzik: Objective-Reinforced Generative Adversarial Networks (ORGAN) for Sequence Generation Models. arXiv:1705.10843v1 [stat.ML] 30 May 2017.
3. Dag Inge Helgøy: Markus LundA Continuous Approach to Controllable Text Generation using Generative Adversarial Network.: Master of Science in Computer Science: June 2017.

THE END

With thanks for your attention