Generative models

Soumith Chintala

Facebook AI Research

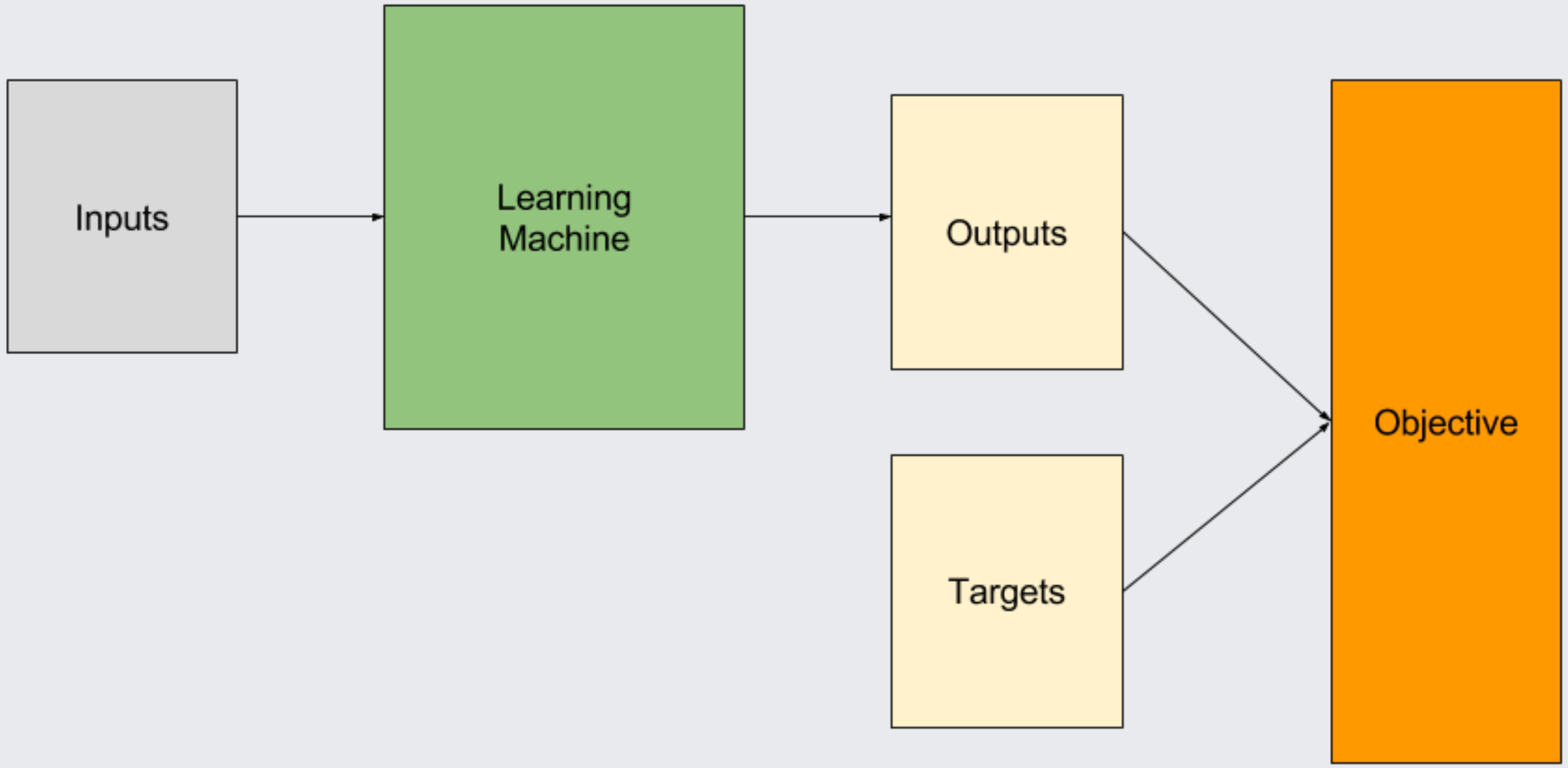
Unfinished GAN business

- Evaluation metrics for GANs
- Regularizing GANs
- GANs as forward models
- GANs in other domains

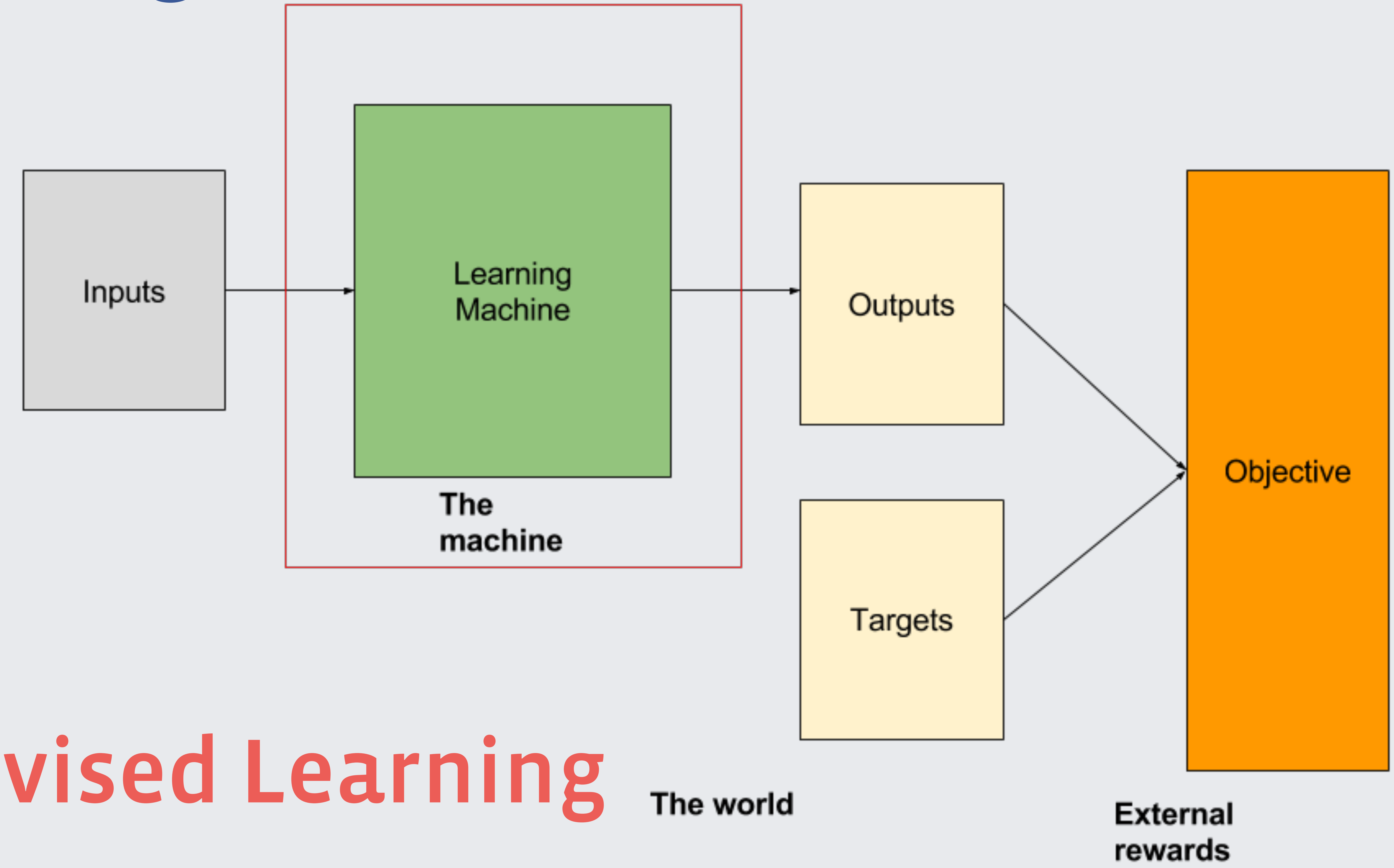
Evaluation / validation

- Human Evaluations
- Semi-supervised Results
- Inception score
  - Improved Techniques for Training GANs: https://arxiv.org/abs/1606.03498
- Frechet Inception Distance

Reusing Representations

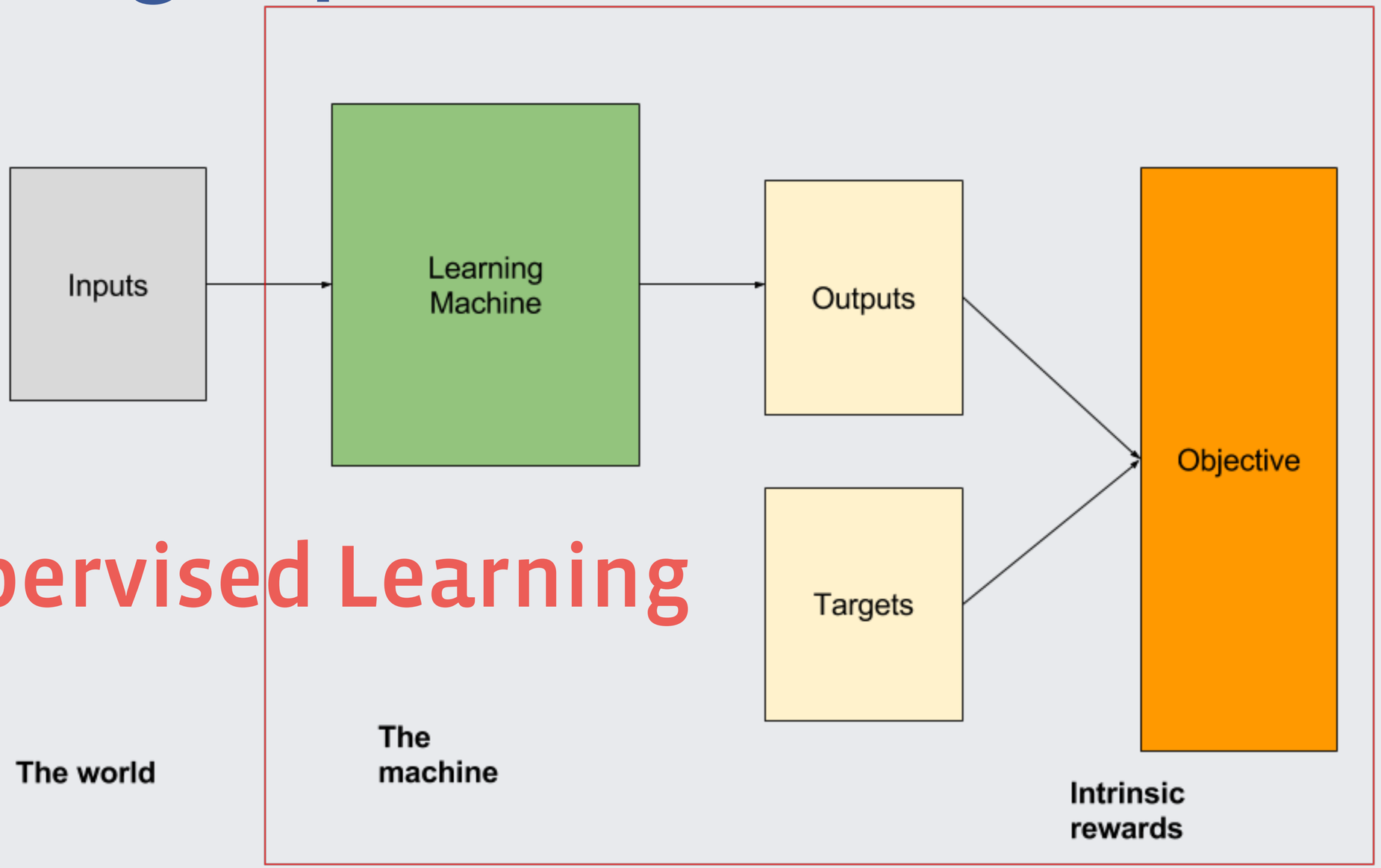


Reusing Representations



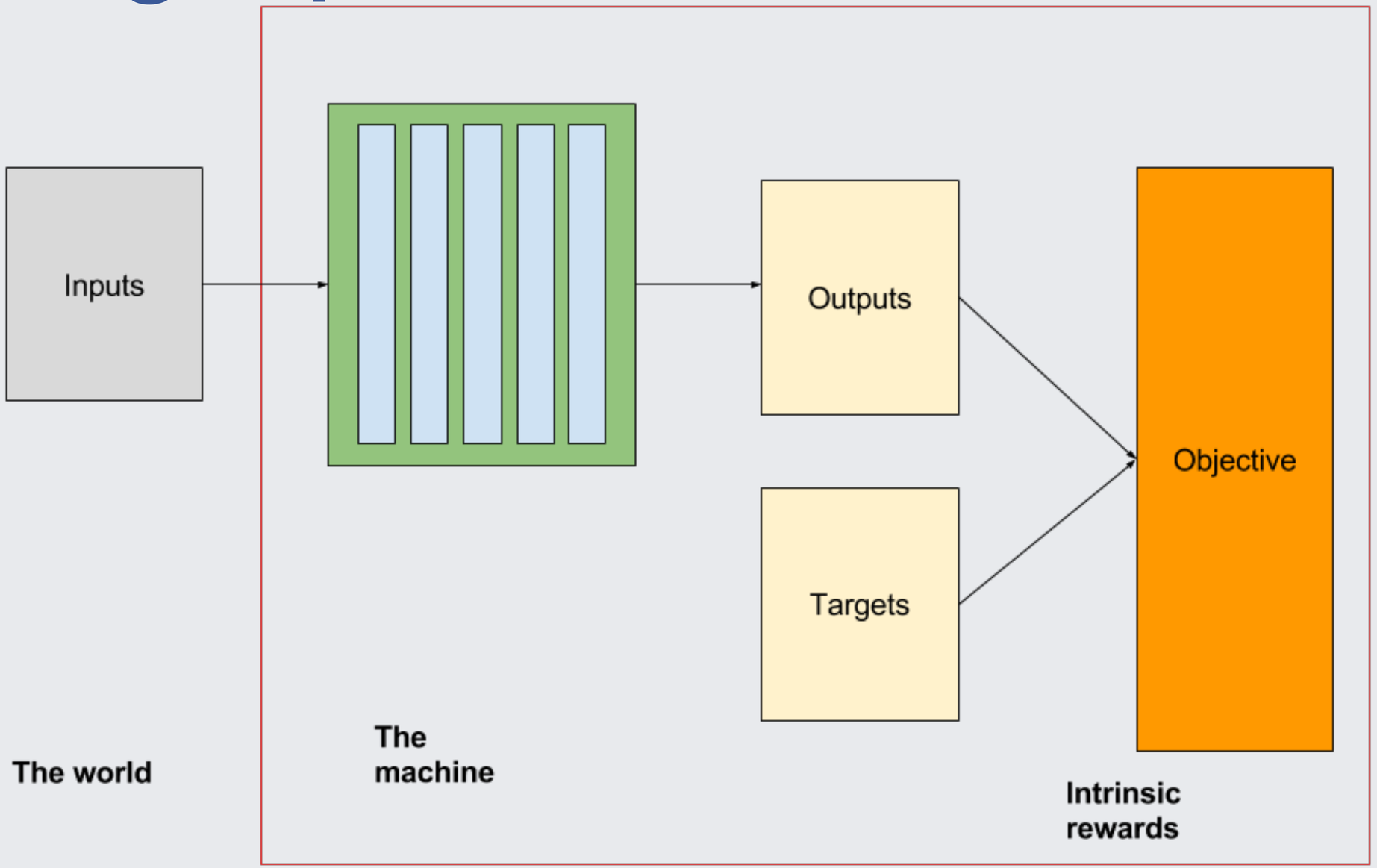
Supervised Learning

Reusing Representations

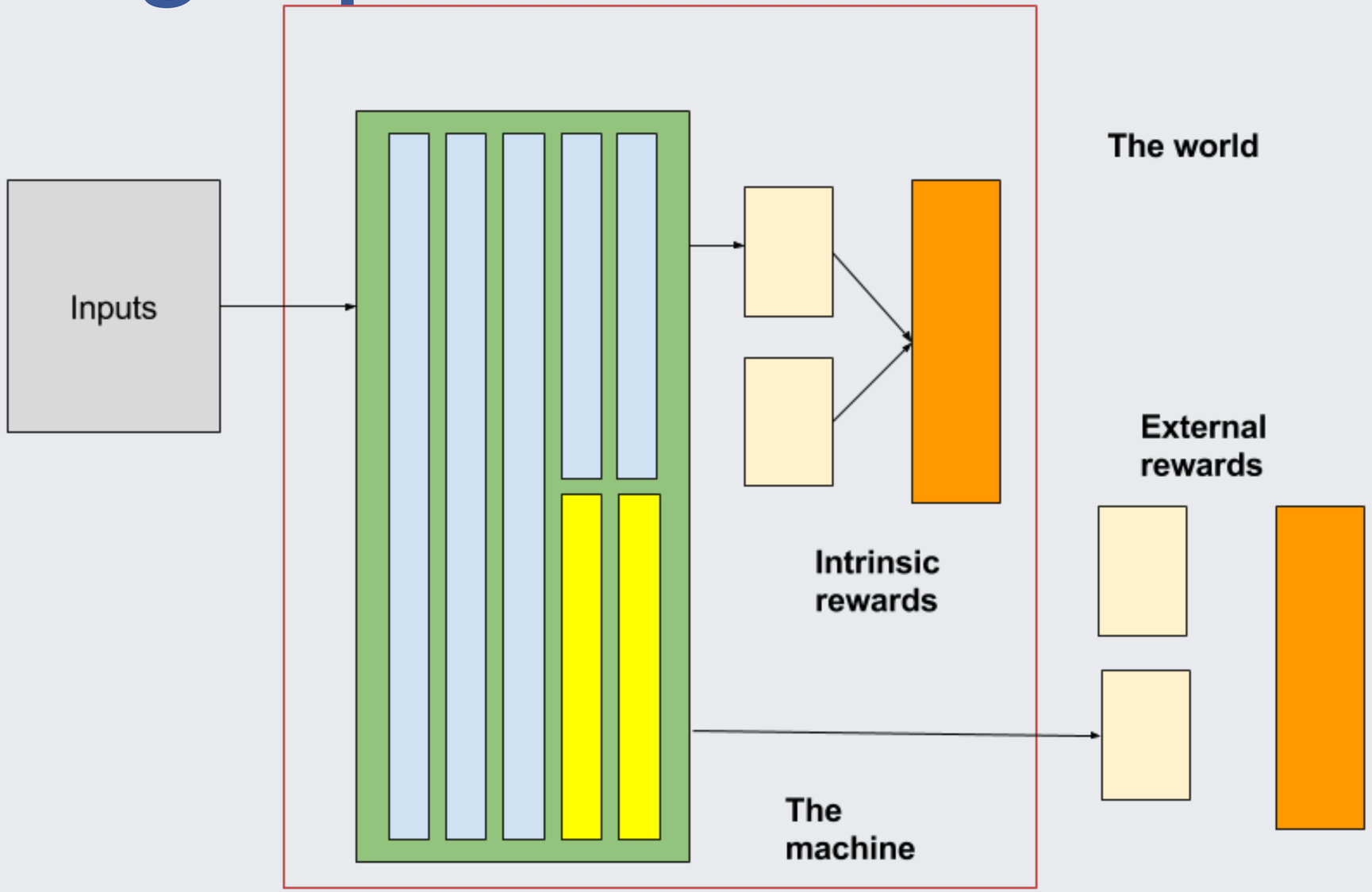


Unsupervised Learning

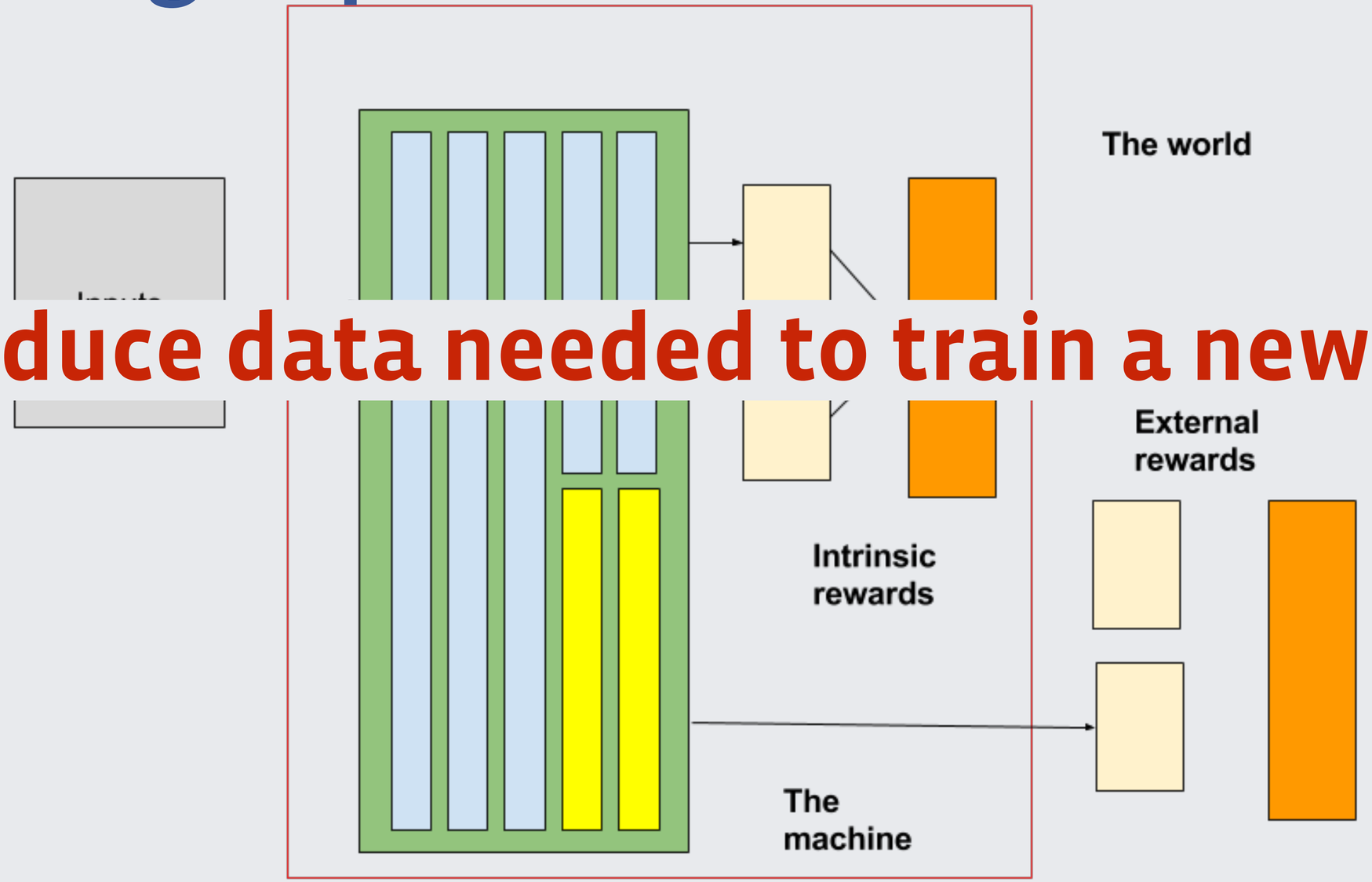
Reusing Representations



Reusing Representations



Reusing Representations



Reduce data needed to train a new task

Using the GAN feature representation

Table 2: SVHN classification with 1000 labels

Model	error rate
KNN	77.93%
TSVM	66.55%
M1+KNN	65.63%
M1+TSVM	54.33%
M1+M2	36.02%
SWWAE without dropout	27.83%
SWWAE with dropout	23.56%
DCGAN (ours) + L2-SVM	22.48%
Supervised CNN with the same architecture	28.87% (validation)

Radford et. al. DCGAN (2015)

Using the GAN feature representation

Model	Percentage of incorrectly predicted test examples for a given number of labeled samples		
	500	1000	2000
DGN [21]		36.02 ± 0.10	
Virtual Adversarial [22]		24.63	
Auxiliary Deep Generative Model [23]		22.86	
Skip Deep Generative Model [23]		16.61 ± 0.24	
Our model	18.44 ± 4.8	8.11 ± 1.3	6.16 ± 0.58
Ensemble of 10 of our models		5.88 ± 1.0	

Salimans et. al. "Improved Techniques for Training GANs" (2016)

Inception Score

Proposed in 2016

Improved Techniques for Training GANs

Tim Salimans tim@openai.com
Ian Goodfellow ian@openai.com
Wojciech Zaremba woj@openai.com
Vicki Cheung vicki@openai.com
Alec Radford alec.radford@gmail.com
Xi Chen peter@openai.com

Abstract

Inception Score

- Send generated image through Inception model (trained on Imagenet)

generated image to get the conditional label distribution $p(y|\boldsymbol{x})$. Images that contain meaningful objects should have a conditional label distribution $p(y|\boldsymbol{x})$ with low entropy. Moreover, we expect the model to generate varied images, so the marginal $\int p(y|\boldsymbol{x} = G(z))dz$ should have high entropy. Combining these two requirements, the metric that we propose is: $\exp(\mathbb{E}_{\boldsymbol{x}}\mathrm{KL}(p(y|\boldsymbol{x})||p(y)))$, where

Inception Score

- Send generated image through Inception model (trained on Imagenet)

generated image to get the conditional label distribution $p(y|\boldsymbol{x})$. Images that contain meaningful objects should have a conditional label distribution $p(y|\boldsymbol{x})$ with low entropy. Moreover, we expect the model to generate varied images, so the marginal $\int p(y|\boldsymbol{x} = G(z))dz$ should have high entropy. Combining these two requirements, the metric that we propose is: $\exp(\mathbb{E}_{\boldsymbol{x}}\mathrm{KL}(p(y|\boldsymbol{x})||p(y)))$, where

Inception Score

- Send generated image through Inception model (trained on Imagenet)

generated image to get the conditional label distribution $p(y|\boldsymbol{x})$. Images that contain meaningful objects should have a conditional label distribution $p(y|\boldsymbol{x})$ with low entropy. Moreover, we expect the model to generate varied images, so the marginal $\int p(y|\boldsymbol{x} = G(z))dz$ should have high entropy. Combining these two requirements, the metric that we propose is: $\exp(\mathbb{E}_{\boldsymbol{x}}\mathrm{KL}(p(y|\boldsymbol{x})||p(y)))$ where

Inception Score

- Send generated image through Inception model (trained on Imagenet)

generated image to get the conditional label distribution $p(y|\boldsymbol{x})$. Images that contain meaningful objects should have a conditional label distribution $p(y|\boldsymbol{x})$ with low entropy. Moreover, we expect the model to generate varied images, so the marginal $\int p(y|\boldsymbol{x} = G(z))dz$ should have high entropy. Combining these two requirements, the metric that we propose is: $\exp(\mathbb{E}_{\boldsymbol{x}}\mathrm{KL}(p(y|\boldsymbol{x})||p(y)))$ where

Used in subsequent literature to compare models

Frechet-Inception-Distance (FID)

GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium

Martin Heusel **Hubert Ramsauer** **Thomas Unterthiner** **Bernhard Nessler**

Sepp Hochreiter

LIT AI Lab & Institute of Bioinformatics,
Johannes Kepler University Linz
A-4040 Linz, Austria
{mhe,ramsauer,unterthiner,nessler,hochreit}@bioinf.jku.at

Frechet-Inception-Distance (FID)

- Drawback of the Inception Score is that the statistics of real world samples are not used and compared to the statistics of synthetic samples

Frechet-Inception-Distance (FID)

- 1. Send real images through Inception
  - Get outputs of last pooling layer

Frechet-Inception-Distance (FID)

- 1. Send real images through Inception
  - Get outputs of last pooling layer
- 2. Compute {mean, covariance} of output set (mu1, C1)

Frechet-Inception-Distance (FID)

- 1. Send real images through Inception
  - Get outputs of last pooling layer
- 2. Compute {mean, covariance} of output set (mu1, C1)
- 3. Repeat (1), (2) for generated images to get (mu2, C2)

Frechet-Inception-Distance (FID)

- 1. Send real images through Inception
  - Get outputs of last pooling layer
- 2. Compute {mean, covariance} of output set (mu1, C1)
- 3. Repeat (1), (2) for generated images to get (mu2, C2)
- 4. Fit two multivariate gaussians with
  - X1 ~ N(mu1, C1)
  - X2 ~ N(mu2, C2)

$$d^2((\boldsymbol{m}, \boldsymbol{C}), (\boldsymbol{m}_w, \boldsymbol{C}_w)) = \|\boldsymbol{m} - \boldsymbol{m}_w\|_2^2 + \operatorname{Tr}\left(\boldsymbol{C} + \boldsymbol{C}_w - 2\left(\boldsymbol{C}\boldsymbol{C}_w\right)^{1/2}\right)$$

Frechet-Inception-Distance (FID)

- More correlated with human judgement
- Uses real-world statistics as well

Regularizing GANs

- The gradient of the discriminator is unbounded!!!

Regularizing GANs

- The gradient of the discriminator is unbounded!!!

$$D_G^*(\boldsymbol{x}) = \frac{q_{\text{data}}(\boldsymbol{x})}{q_{\text{data}}(\boldsymbol{x}) + p_G(\boldsymbol{x})} = \text{sigmoid}(f^*(\boldsymbol{x})), \text{where } f^*(\boldsymbol{x}) = \log q_{\text{data}}(\boldsymbol{x}) - \log p_G(\boldsymbol{x}), \quad (3)$$

and its derivative

$$\nabla_{\boldsymbol{x}} f^*(x) = \frac{1}{q_{\text{data}}(\boldsymbol{x})} \nabla_{\boldsymbol{x}} q_{\text{data}}(\boldsymbol{x}) - \frac{1}{p_G(\boldsymbol{x})} \nabla_{\boldsymbol{x}} p_G(\boldsymbol{x}) \quad (4)$$

can be unbounded or even incomputable. This prompts us to introduce some regularity condition to the derivative of $f(\boldsymbol{x})$.

Miyato et. al. "Spectral Normalization for GANs" (2018)

Regularizing GANs

- Multiple papers (WGAN, Improved WGAN, LSGAN, Improved GAN, Spectral Normalization GANs) argue for boundedness of discriminator

Regularizing GANs

- Multiple papers (WGAN, Improved WGAN, LSGAN, Improved GAN, Spectral Normalization GANs) argue for boundedness of discriminator
- Current empirical best (and latest): Spectral Normalization

Regularizing GANs

- Multiple papers (WGAN, Improved WGAN, LSGAN, Improved GAN, Spectral Normalization GANs) argue for boundedness of discriminator
- Current empirical best (and latest): Spectral Normalization
- ″contemporary regularizations including weight normalization and weight clipping implicitly impose constraints on weight matrices that places unnecessary restriction on the search space of the discriminator. More specifically, we will show that weight normalization and weight clipping unwittingly favor low-rank weight matrices.″

Regularizing GANs (Spectral Normalization)

Algorithm 1 SGD with spectral normalization

- Initialize $\tilde{\boldsymbol{u}}_l \in \mathcal{R}^{d_l}$ for $l = 1, \ldots, L$ with a random vector (sampled from isotropic distribution).
- For each update and each layer l:
  1. Apply power iteration method to a unnormalized weight W^l:

$$\tilde{\boldsymbol{v}}_l \leftarrow (W^l)^{\mathrm{T}} \tilde{\boldsymbol{u}}_l / \|(W^l)^{\mathrm{T}} \tilde{\boldsymbol{u}}_l\|_2 \tag{20}$$

$$\tilde{\boldsymbol{u}}_l \leftarrow W^l \tilde{\boldsymbol{v}}_l / \|W^l \tilde{\boldsymbol{v}}_l\|_2 \tag{21}$$

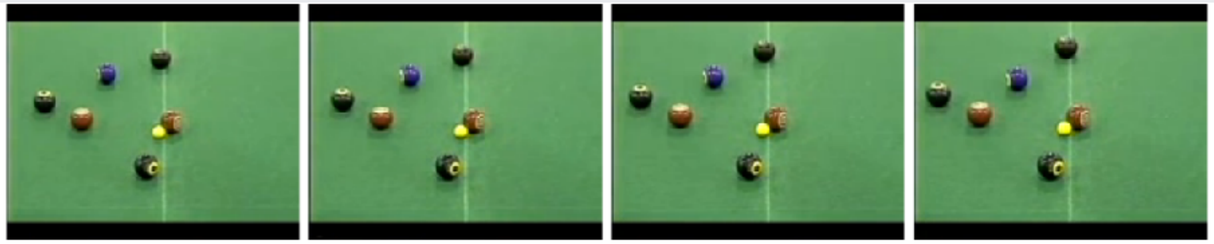
  2. Calculate $\bar{W}_{\mathrm{SN}}$ with the spectral norm:

$$\bar{W}^l_{\mathrm{SN}}(W^l) = W^l / \sigma(W^l), \text{ where } \sigma(W^l) = \tilde{\boldsymbol{u}}_l^{\mathrm{T}} W^l \tilde{\boldsymbol{v}}_l \tag{22}$$

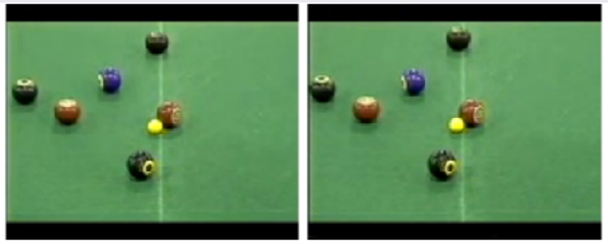
  3. Update W^l with SGD on mini-batch dataset $\mathcal{D}_M$ with a learning rate α:

$$W^l \leftarrow W^l - \alpha \nabla_{W^l} \ell(\bar{W}^l_{\mathrm{SN}}(W^l), \mathcal{D}_M) \tag{23}$$

GANs as forward models

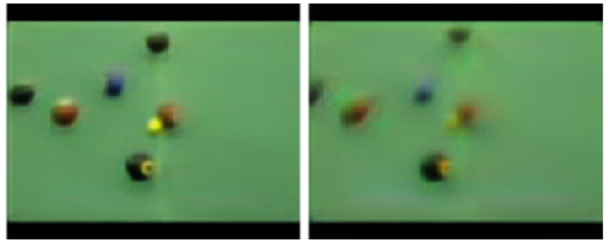


Input frames

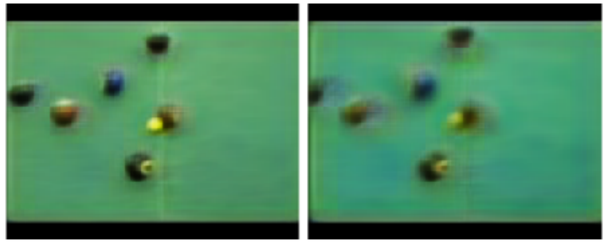


Ground truth

ℓ_2 result

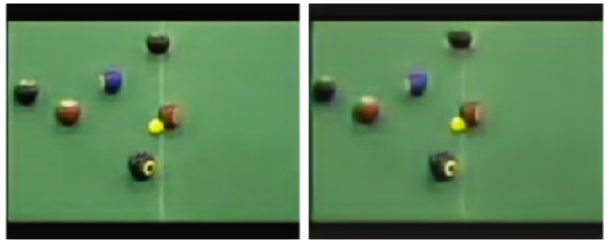


ℓ_1 result



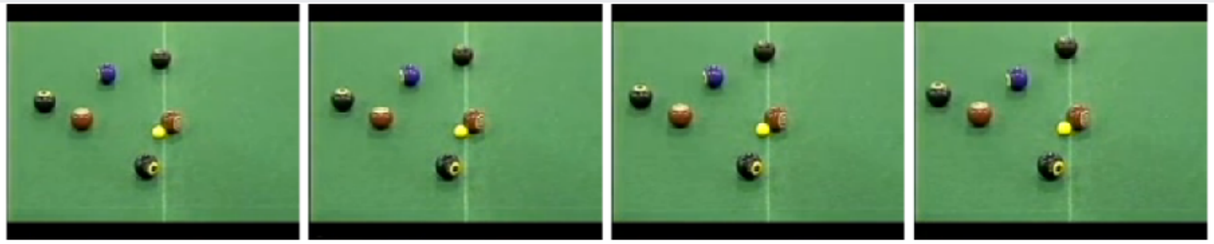
GDL ℓ_1 result

Adversarial result

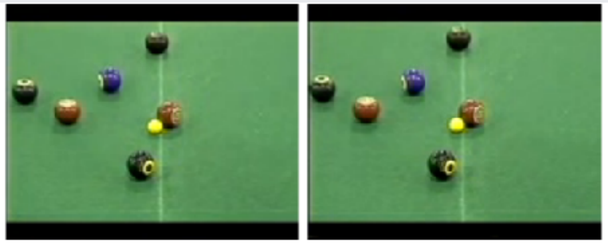


Adversarial+GDL result

GANs as forward models

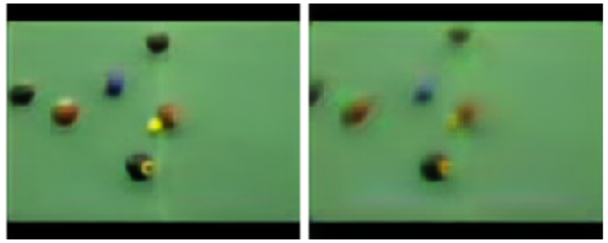


Input frames

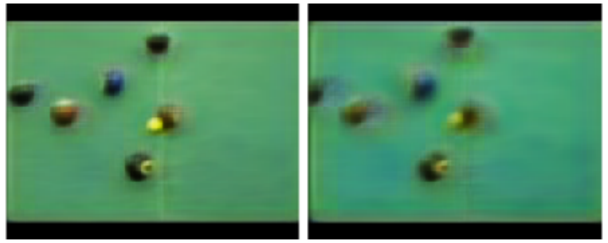


Ground truth

ℓ_2 result

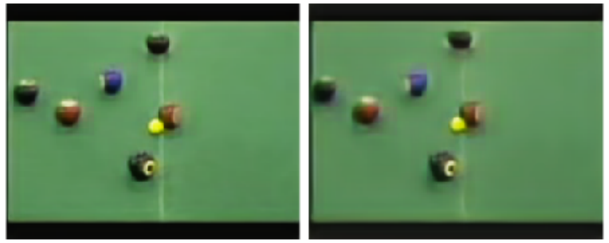


ℓ_1 result



GDL ℓ_1 result

Adversarial result



Adversarial+GDL result

Do we really need to predict in pixel space?

Generative models as forward models

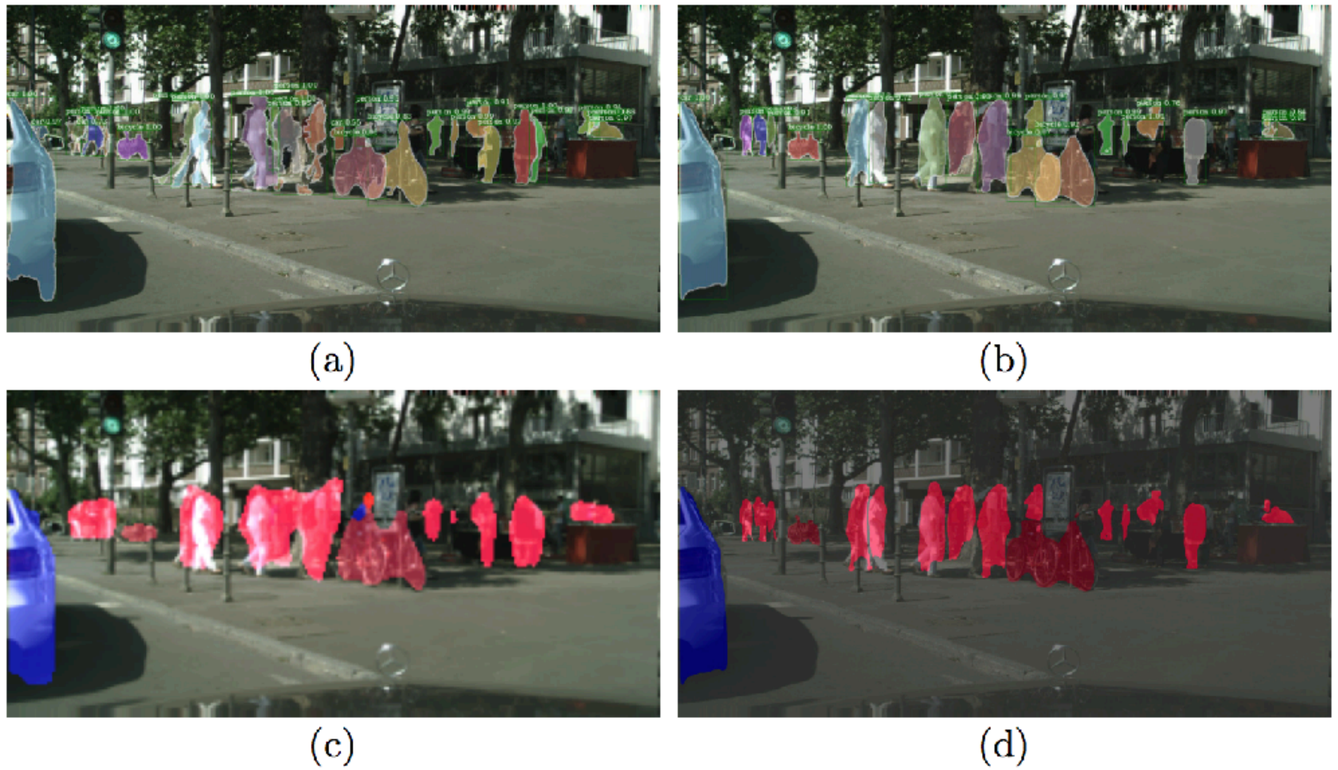


Fig. 1: Predicting 0.5 sec. into the future. Instance segmentations with (a) optical flow baseline and (b) our approach. Semantic segmentation (c) from [8] and (d) derived from our instance semantic segmentation approach. Instance modeling significantly improves the segmentation accuracy of the individual pedestrians.

Generative models as forward models

- Predict in semantic space
  - complexity is much lower

Other domains / problems

- Video Prediction
- Image in-painting
- image to image translation
- text-conditional

Video Prediction GANs

arxiv.org/abs/1511.05440

Cornell University Library

arXiv.org > cs > arXiv:1511.05440

Search or

Computer Science > Learning

Deep multi-scale video prediction beyond mean square error

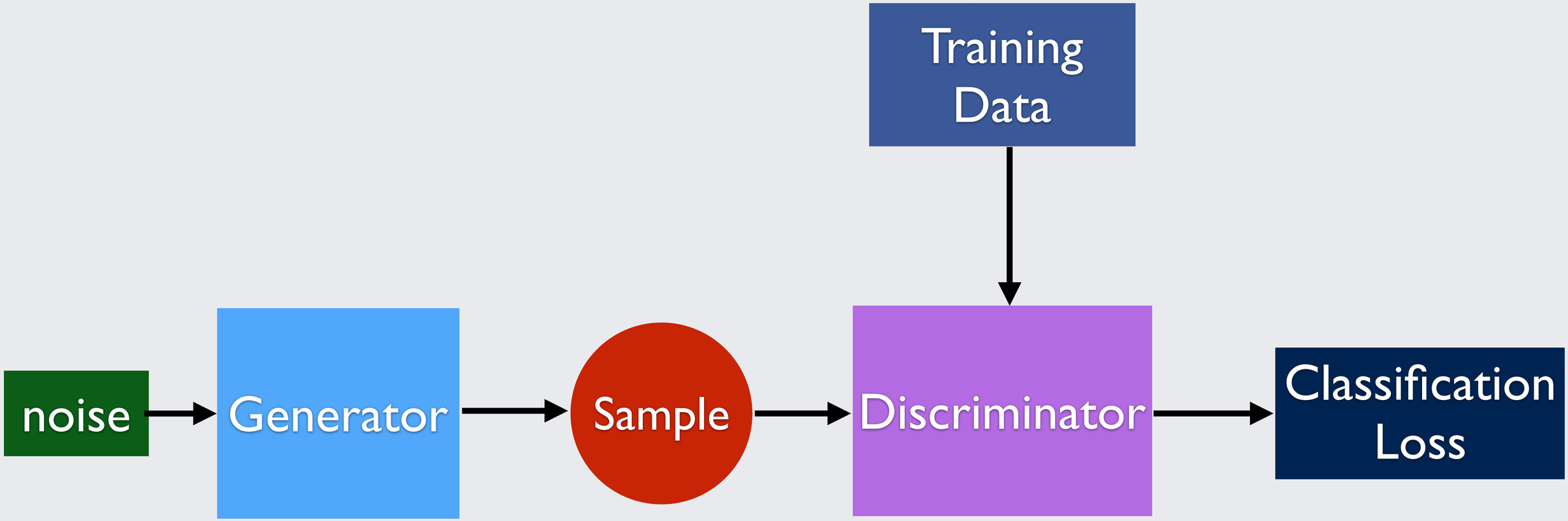
Michael Mathieu, Camille Couprie, Yann LeCun

(Submitted on 17 Nov 2015 (v1), last revised 26 Feb 2016 (this version, v6))

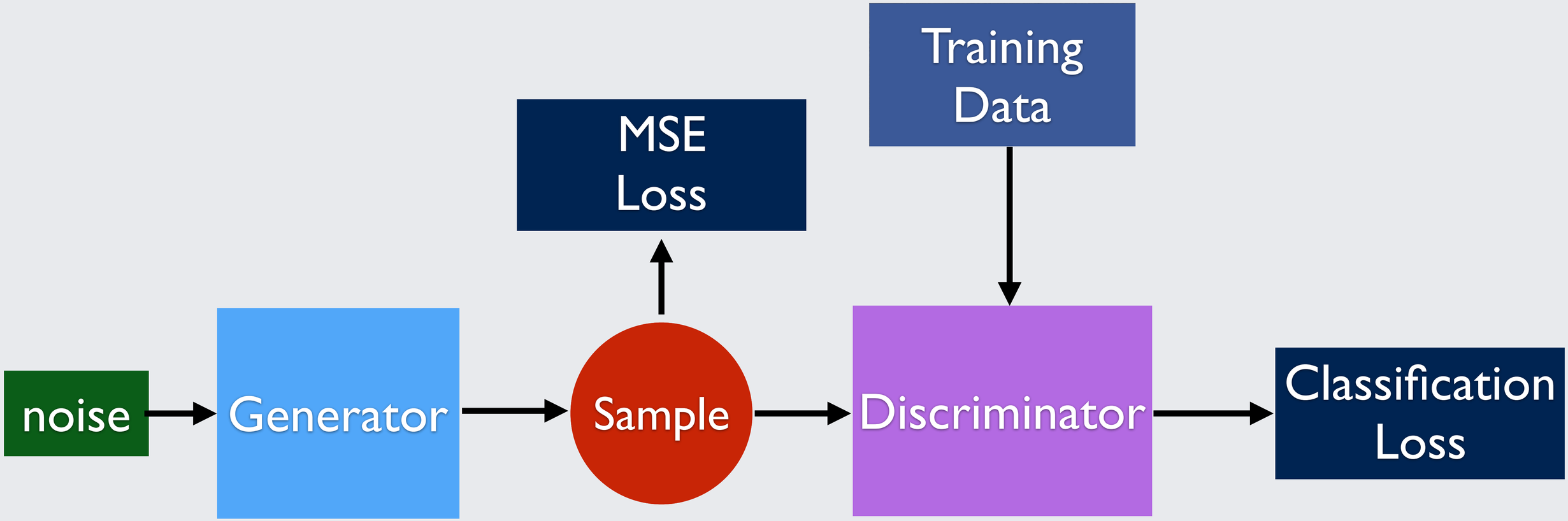
Learning to predict future images from a video sequence involves the construction of an internal representation that models the image evolution accurately, and therefore, to some degree, its content and dynamics. This is why pixel-space video prediction may be viewed as a promising avenue for unsupervised feature learning. In addition, while optical flow has been a very studied problem in computer vision for a long time, future frame prediction is rarely approached. Still, many vision applications could benefit from the knowledge of the next frames of videos, that does not require the complexity of tracking every pixel trajectories. In this work, we train a convolutional network to generate future frames given an input sequence. To deal with the inherently blurry predictions obtained from the standard Mean Squared Error (MSE) loss function, we propose three different and complementary feature learning strategies: a multi-scale architecture, an adversarial training method, and an image gradient difference loss function. We compare our predictions to different published results based on recurrent neural networks on the UCF101 dataset

Subjects: **Learning (cs.LG)**; Computer Vision and Pattern Recognition (cs.CV); Machine Learning (stat.ML)
Cite as: **arXiv:1511.05440 [cs.LG]**
(or **arXiv:1511.05440v6 [cs.LG]** for this version)

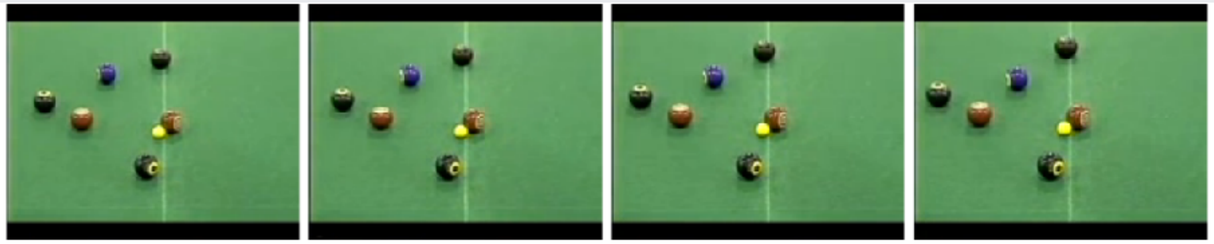
Video Prediction GANs



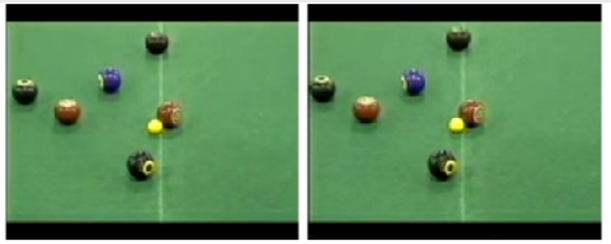
Video Prediction GANs



Video Prediction GANs

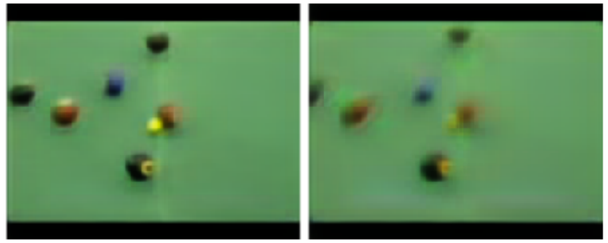


Input frames

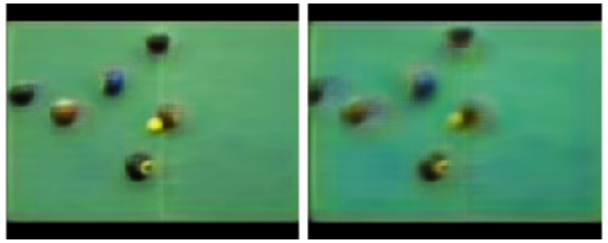


Ground truth

ℓ_2 result

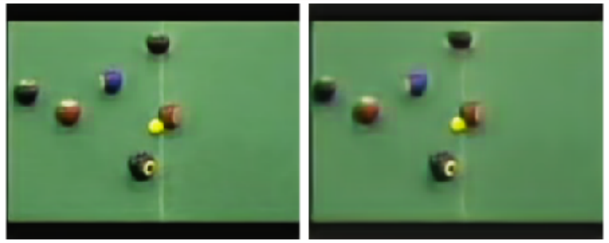


ℓ_1 result



GDL ℓ_1 result

Adversarial result



Adversarial+GDL result

In-painting GANs

Context Encoders: Feature Learning by Inpainting

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, Alexei A. Efros

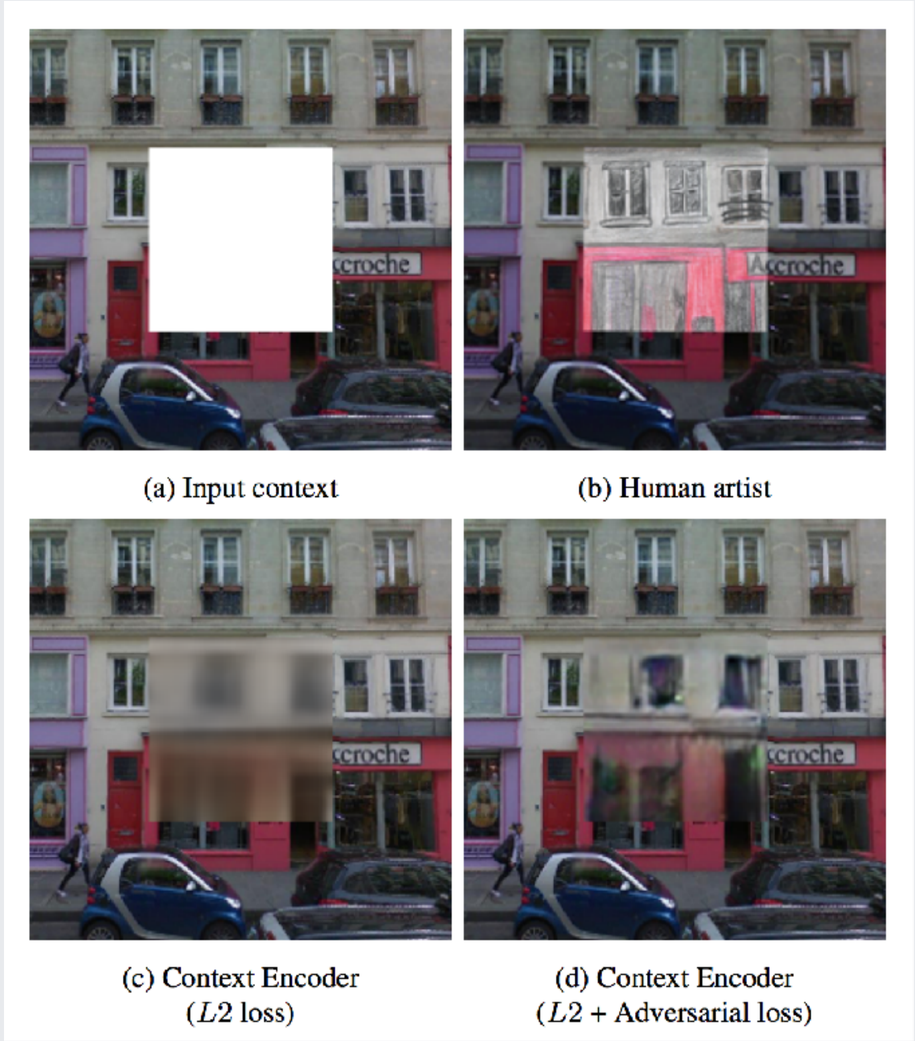
(Submitted on 25 Apr 2016)

We present an unsupervised visual feature learning algorithm driven by context-based pixel prediction. By analogy with auto-encoders, we propose Context Encoders -- a convolutional neural network trained to generate the contents of an arbitrary image region conditioned on its surroundings. In order to succeed at this task, context encoders need to both understand the content of the entire image, as well as produce a plausible hypothesis for the missing part(s). When training context encoders, we have experimented with both a standard pixel-wise reconstruction loss, as well as a reconstruction plus an adversarial loss. The latter produces much sharper results because it can better handle multiple modes in the output. We found that a context encoder learns a representation that captures not just appearance but also the semantics of visual structures. We quantitatively demonstrate the effectiveness of our learned features for CNN pre-training on classification, detection, and segmentation tasks. Furthermore, context encoders can be used for semantic inpainting tasks, either stand-alone or as initialization for non-parametric methods.

Comments: CVPR 2016
Subjects: **Computer Vision and Pattern Recognition (cs.CV)**; Artificial Intelligence (cs.AI); Graphics (cs.GR); Learning (cs.LG)
Cite as: **arXiv:1604.07379 [cs.CV]**
(or **arXiv:1604.07379v1 [cs.CV]** for this version)

In-painting GANs

In-painting GANs



(a) Input context

(b) Human artist

(c) Context Encoder ($L2$ loss)

(d) Context Encoder ($L2$ + Adversarial loss)

Text-conditional GANs

Generative Adversarial Text to Image Synthesis

Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, Honglak Lee

(Submitted on 17 May 2016 (v1), last revised 5 Jun 2016 (this version, v2))

Automatic synthesis of realistic images from text would be interesting and useful, but current AI systems are still far from this goal. However, in recent years generic and powerful recurrent neural network architectures have been developed to learn discriminative text feature representations. Meanwhile, deep convolutional generative adversarial networks (GANs) have begun to generate highly compelling images of specific categories, such as faces, album covers, and room interiors. In this work, we develop a novel deep architecture and GAN formulation to effectively bridge these advances in text and image model- ing, translating visual concepts from characters to pixels. We demonstrate the capability of our model to generate plausible images of birds and flowers from detailed text descriptions.

Comments: ICML 2016
Subjects: **Neural and Evolutionary Computing (cs.NE)**; Computer Vision and Pattern Recognition (cs.CV)
Cite as: **arXiv:1605.05396 [cs.NE]**
(or **arXiv:1605.05396v2 [cs.NE]** for this version)

Text-conditional GANs

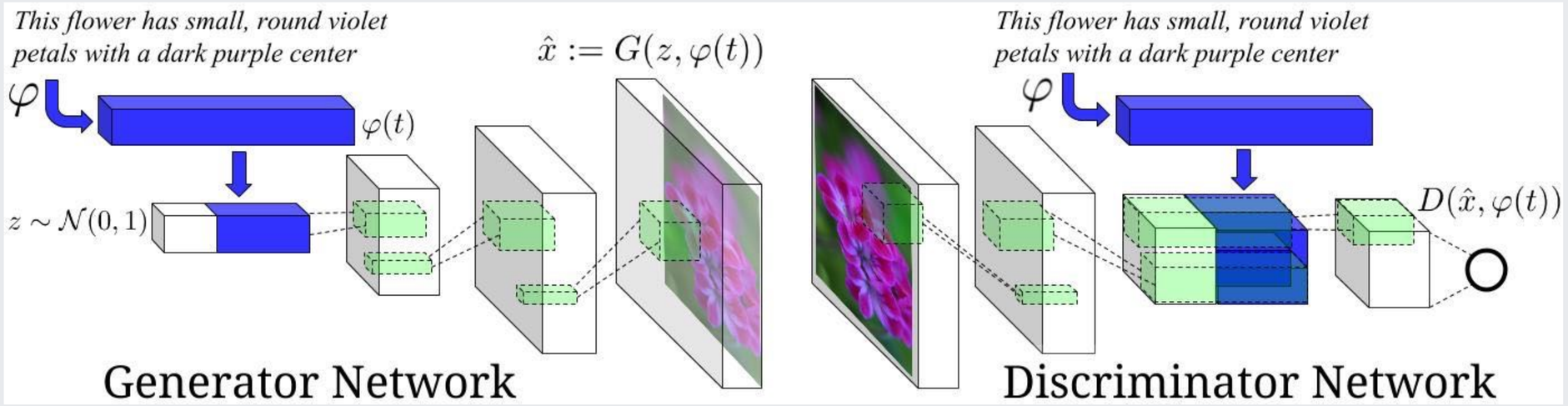


Figure from Reed et. al. 2016

Text-conditional GANs

Caption	Image
a pitcher is about to throw the ball to the batter	
a group of people on skis stand in the snow	
a man in a wet suit riding a surfboard on a wave	

Text-conditional GANs

Caption	Image
this flower has white petals and a yellow stamen	
the center is yellow surrounded by wavy dark purple petals	
this flower has lots of small round pink petals	

Text-conditional GANs

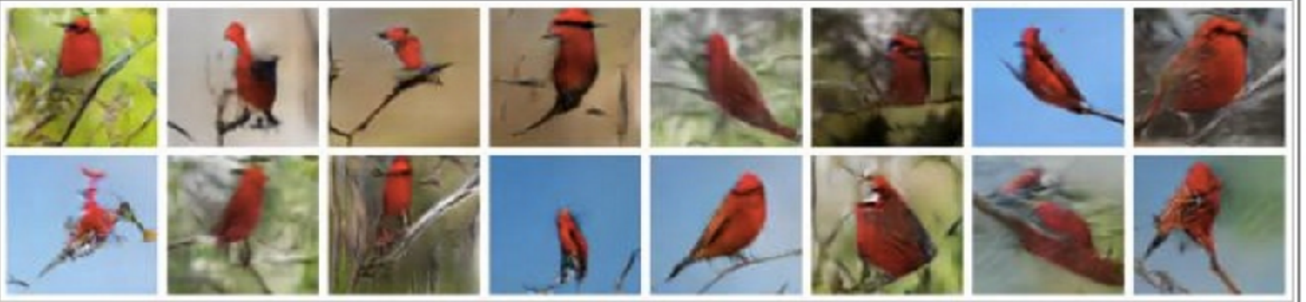
Caption	Image
this vibrant red bird has a pointed black beak	
this bird is yellowish orange with black wings	
the bright blue bird has a white colored belly	

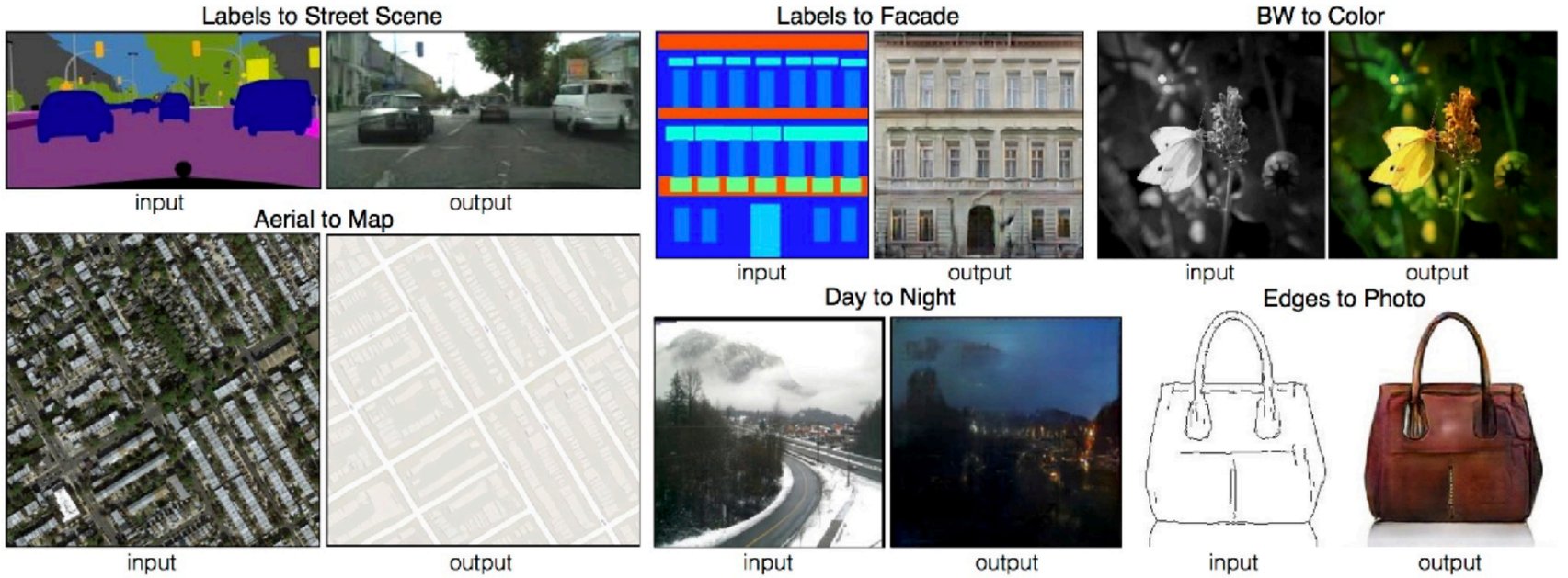
Image-to-image translation

Image-to-Image Translation with Conditional Adversarial Nets

Phillip Isola Jun-Yan Zhu Tinghui Zhou Alexei A. Efros

Univerity of California, Berkeley
In CVPR 2017

[Paper] [GitHub]



Example results on several image-to-image translation problems. In each case we use the same architecture and objective, simply training on different data.

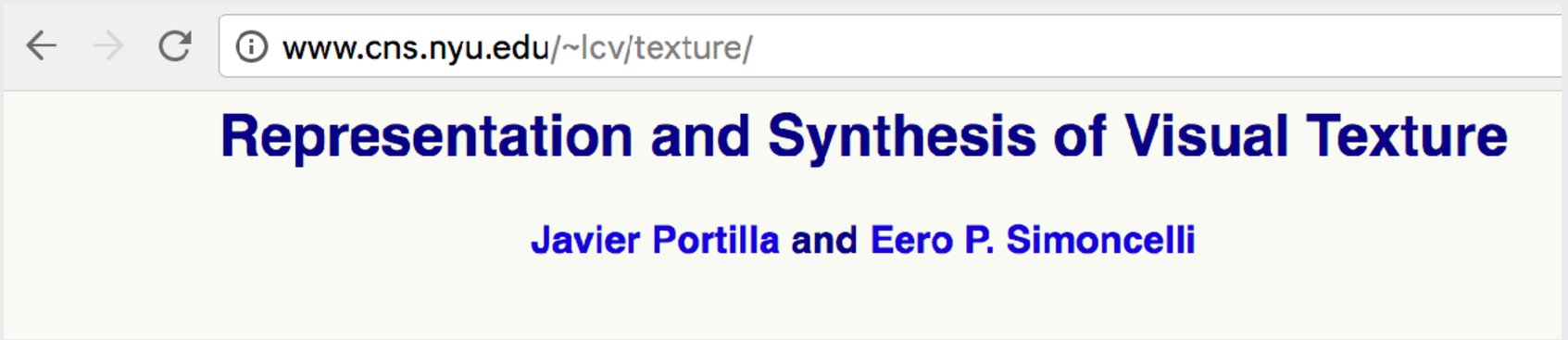
Enough of GANs!!!!!!

Generating Images

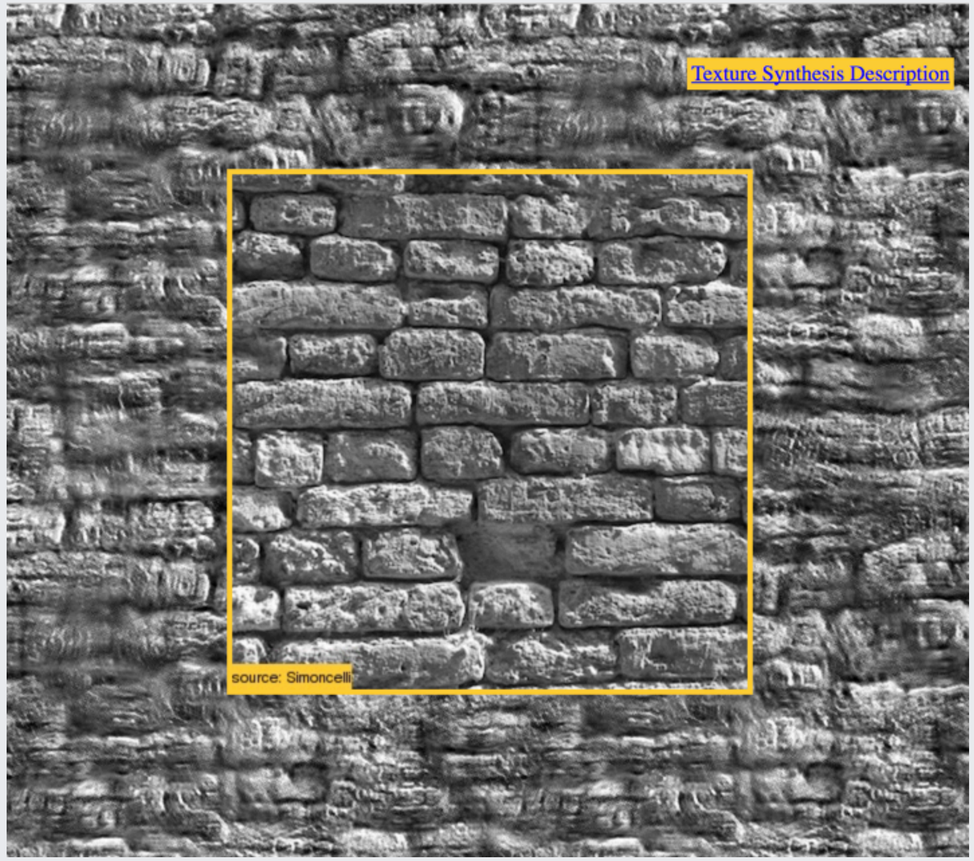
Different approaches in literature

- Wavelet-based
- Quilting
- Auto-encoders
- RBMs
- VAE
- GAN
- Auto-regressive models
- GLO

Wavelet-based



Wavelet-based



Wavelet-based

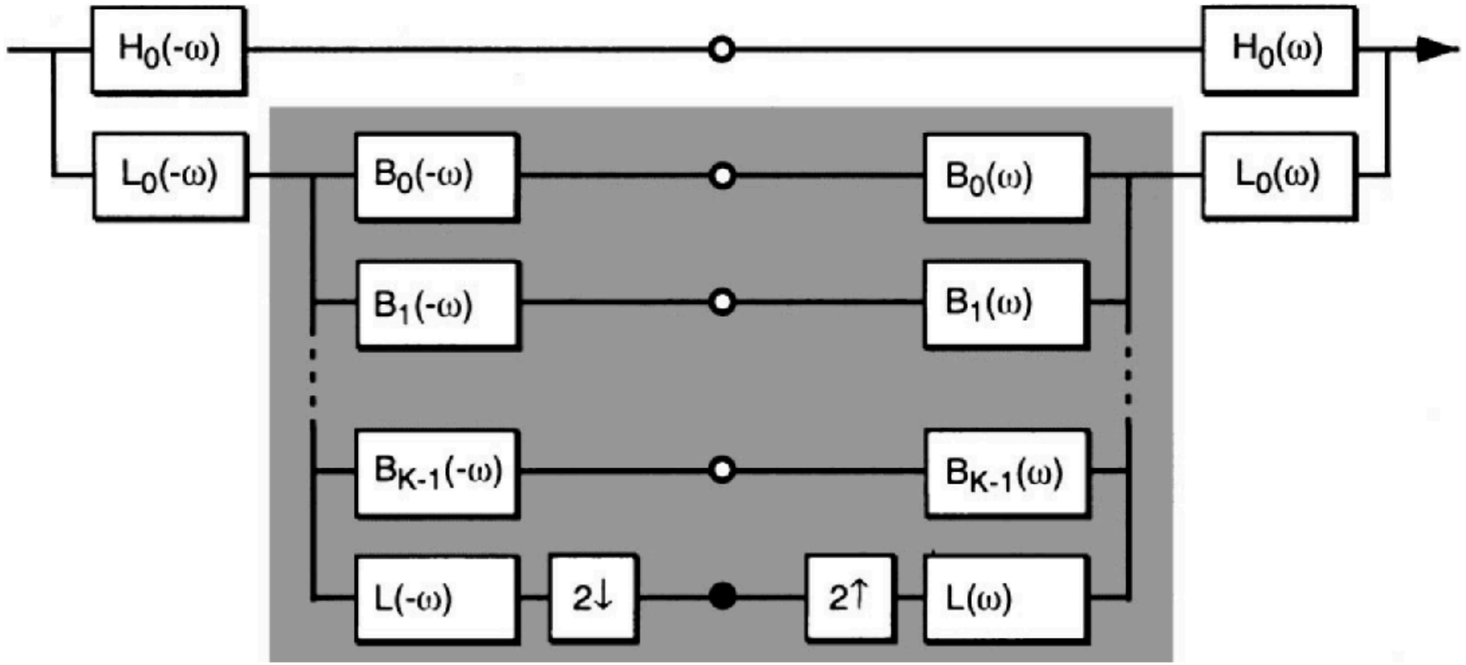


Figure 1. System diagram for the steerable pyramid (Simoncelli and Freeman, 1995). The input image is initially split into high- and lowpass bands. The lowpass band is then further split into a lower-frequency band and a set of oriented subbands. The recursive construction of a pyramid is achieved by inserting a copy of the diagram contents indicated by the shaded region at the location of the solid circle (i.e., the lowpass branch).

Wavelet-based

Wavelet-based

- In-essence, Google's DeepDream is similar
- https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html

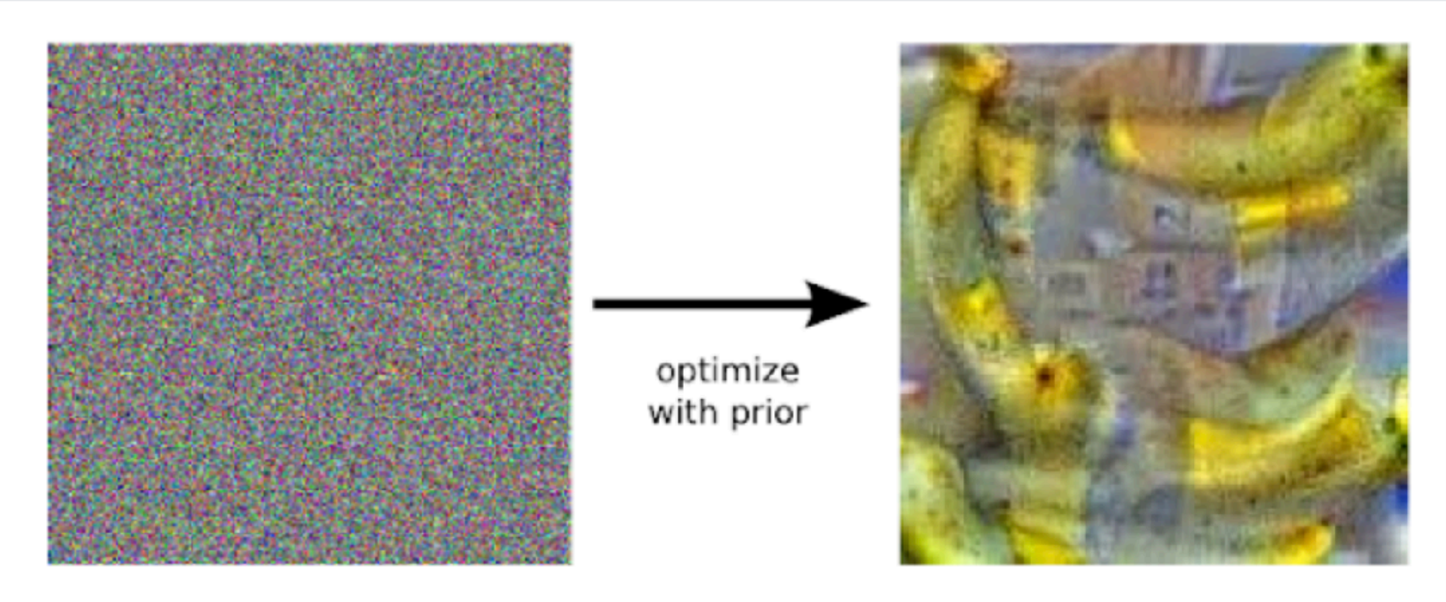


Image quilting

Image Quilting for Texture Synthesis and Transfer

Alexei A. Efros[1,2] William T. Freeman[2]

[1]University of California, Berkeley [2]Mitsubishi Electric Research Laboratories

Image quilting

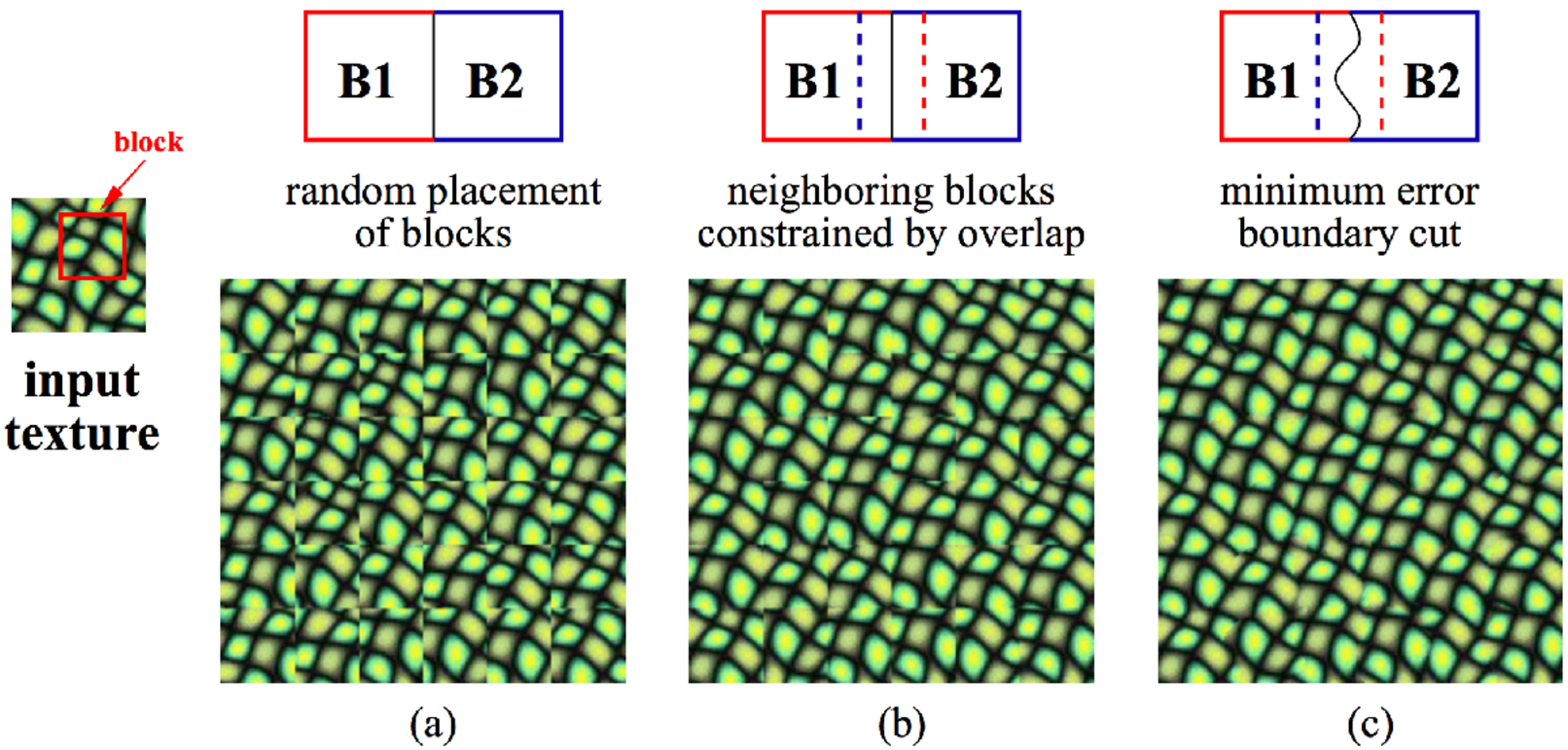


Figure 2: Quilting texture. Square blocks from the input texture are patched together to synthesize a new texture sample: (a) blocks are chosen randomly (similar to [21, 18]), (b) the blocks overlap and each new block is chosen so as to "agree" with its neighbors in the region of overlap, (c) to reduce blockiness the boundary between blocks is computed as a minimum cost path through the error surface at the overlap.

Image quilting

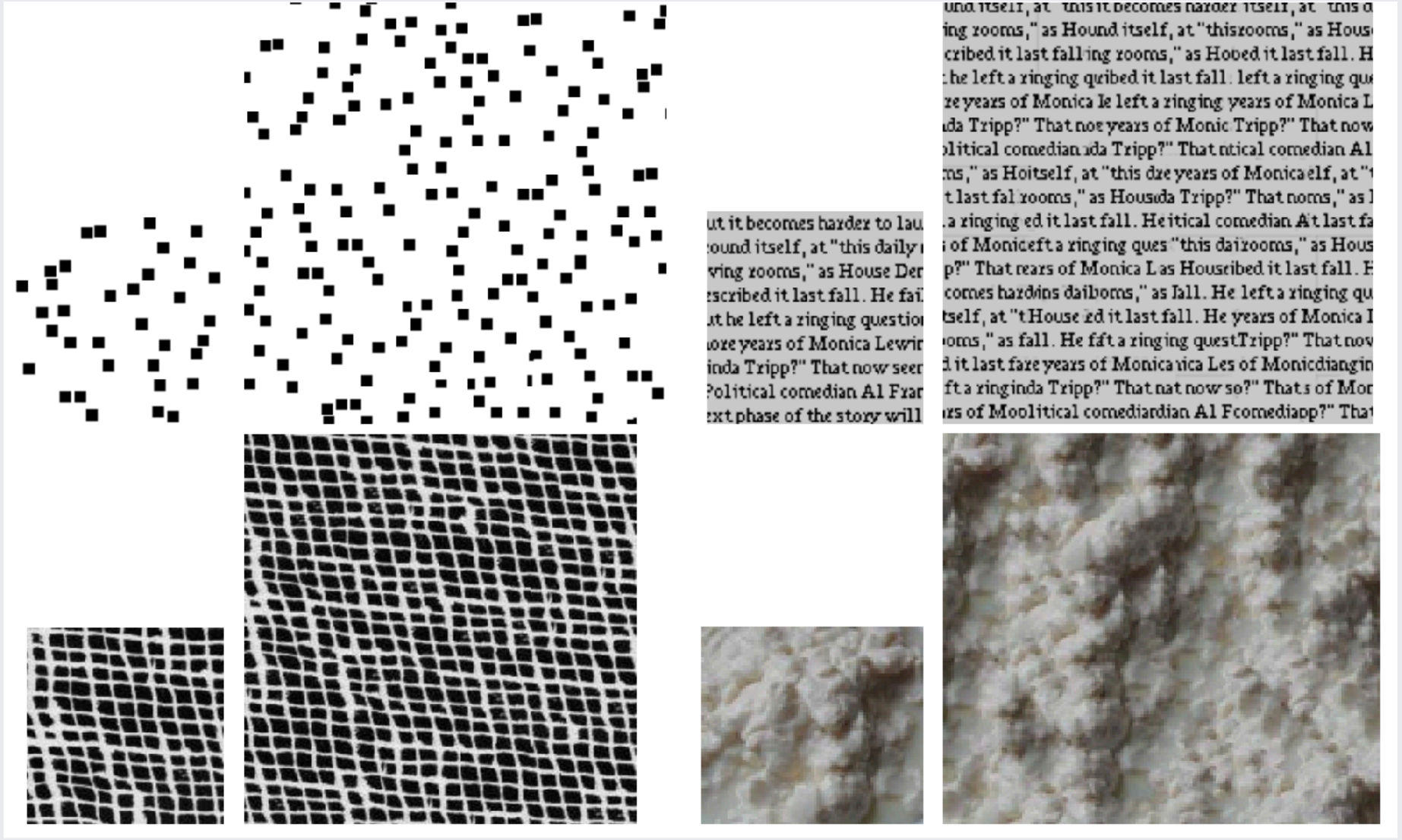


Image quilting

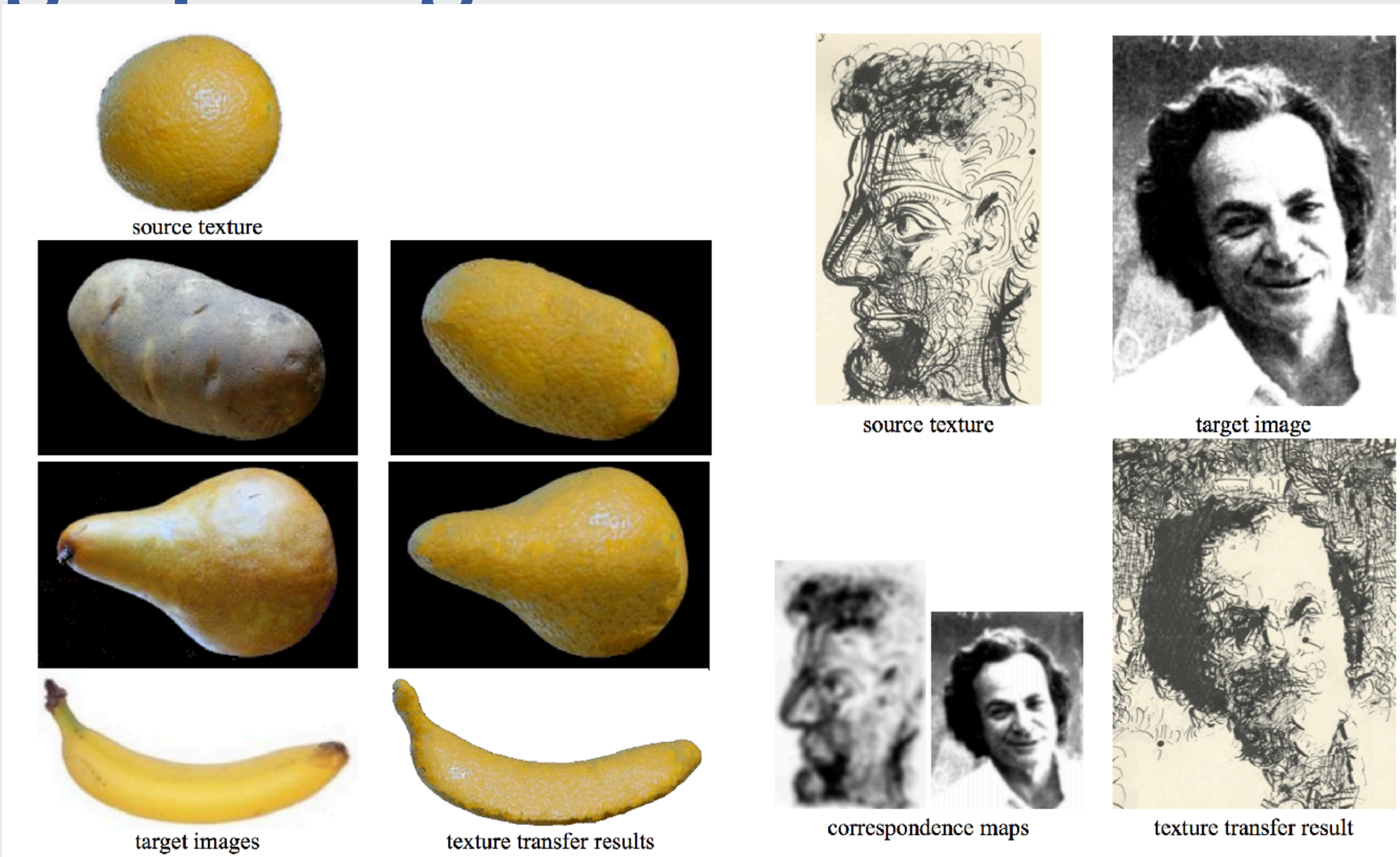


Figure 5: Texture transfer: here, we take the texture from the orange and the Picasso drawing and transfer it onto different objects. The result has the texture of the source image and the correspondence map values of the target image.

Autoencoders

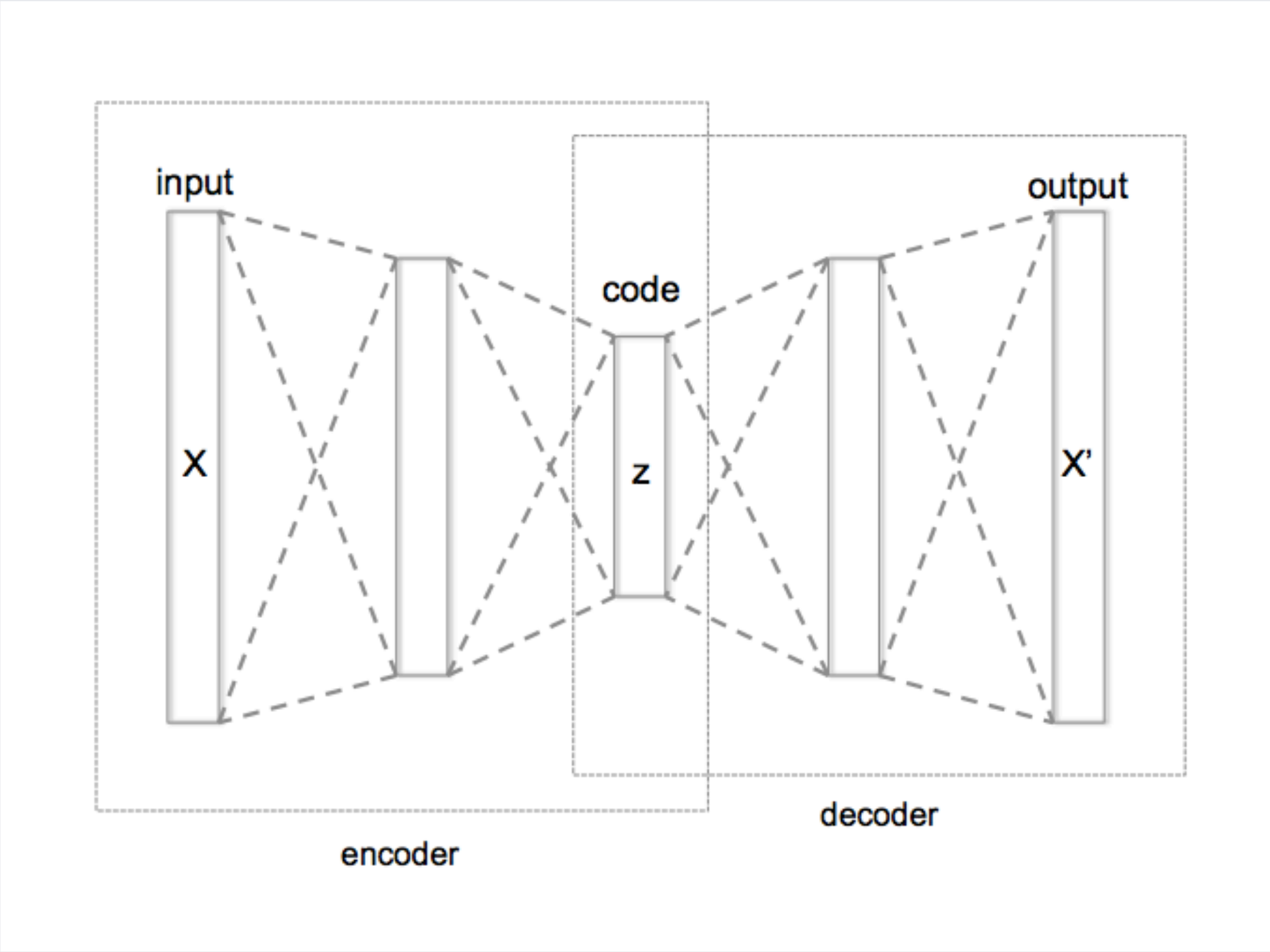


image from wikipedia

Variational Autoencoders

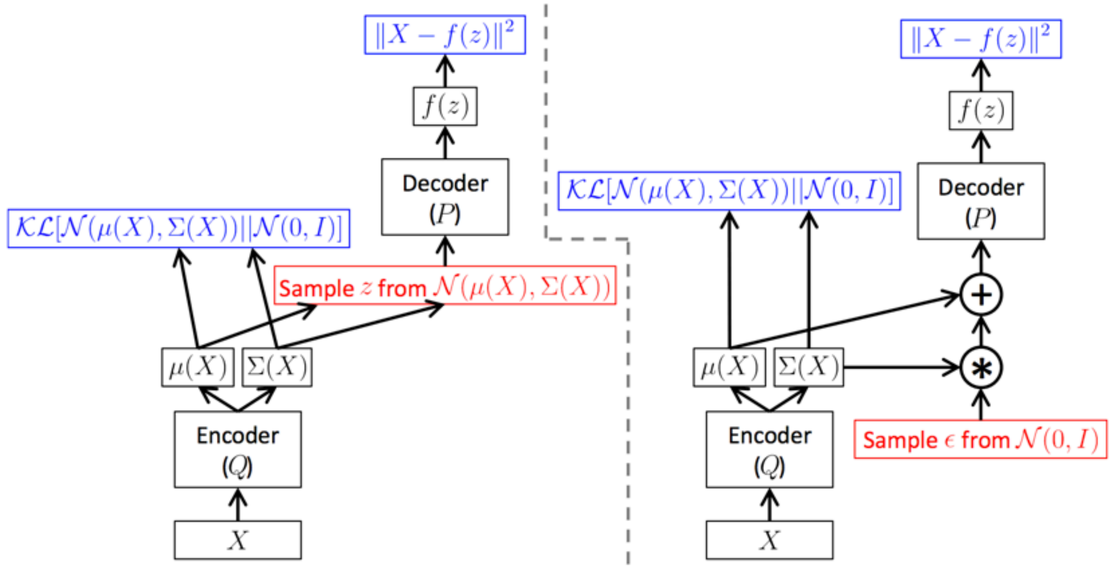
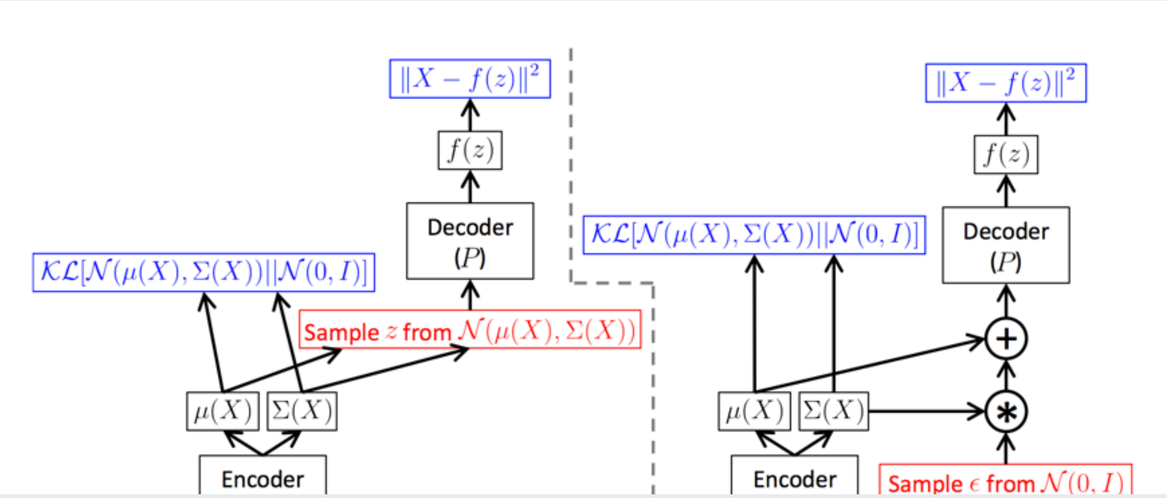


Figure 4: A training-time variational autoencoder implemented as a feed-forward neural network, where $P(X|z)$ is Gaussian. Left is without the "reparameterization trick", and right is with it. Red shows sampling operations that are non-differentiable. Blue shows loss layers. The feedforward behavior of these networks is identical, but backpropagation can be applied only to the right network.

Figure from Carl Doersch, "Tutorial on Variational Autoencoders": https://arxiv.org/abs/1606.05908

Variational Autoencoders



Distinct characteristic: blurry images

Figure 4: A training-time variational autoencoder implemented as a feedforward neural network, where $P(X|z)$ is Gaussian. Left is without the "reparameterization trick", and right is with it. Red shows sampling operations that are non-differentiable. Blue shows loss layers. The feedforward behavior of these networks is identical, but backpropagation can be applied only to the right network.

Figure from Carl Doersch, "Tutorial on Variational Autoencoders": https://arxiv.org/abs/1606.05908

Auto-regressive models

Conditional Image Generation with PixelCNN Decoders

Aäron van den Oord
Google DeepMind
avdnoord@google.com

Nal Kalchbrenner
Google DeepMind
nalk@google.com

Oriol Vinyals
Google DeepMind
vinyals@google.com

Lasse Espeholt
Google DeepMind
espeholt@google.com

Alex Graves
Google DeepMind
gravesa@google.com

Koray Kavukcuoglu
Google DeepMind
korayk@google.com

Pixel-CNN

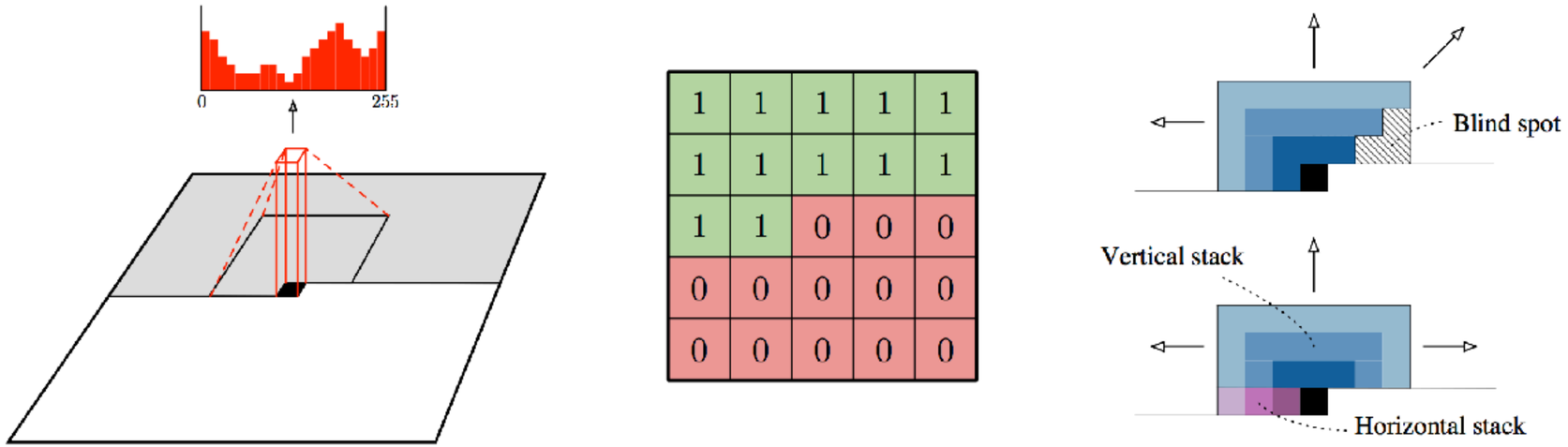


Figure 1: **Left**: A visualization of the PixelCNN that maps a neighborhood of pixels to prediction for the next pixel. To generate pixel x_i the model can only condition on the previously generated pixels $x_1, \ldots x_{i-1}$. **Middle**: an example matrix that is used to mask the 5x5 filters to make sure the model cannot read pixels below (or strictly to the right) of the current pixel to make its predictions. **Right**: Top: PixelCNNs have a *blind spot* in the receptive field that can not be used to make predictions. Bottom: Two convolutional stacks (blue and purple) allow to capture the whole receptive field.

Pixel-CNN

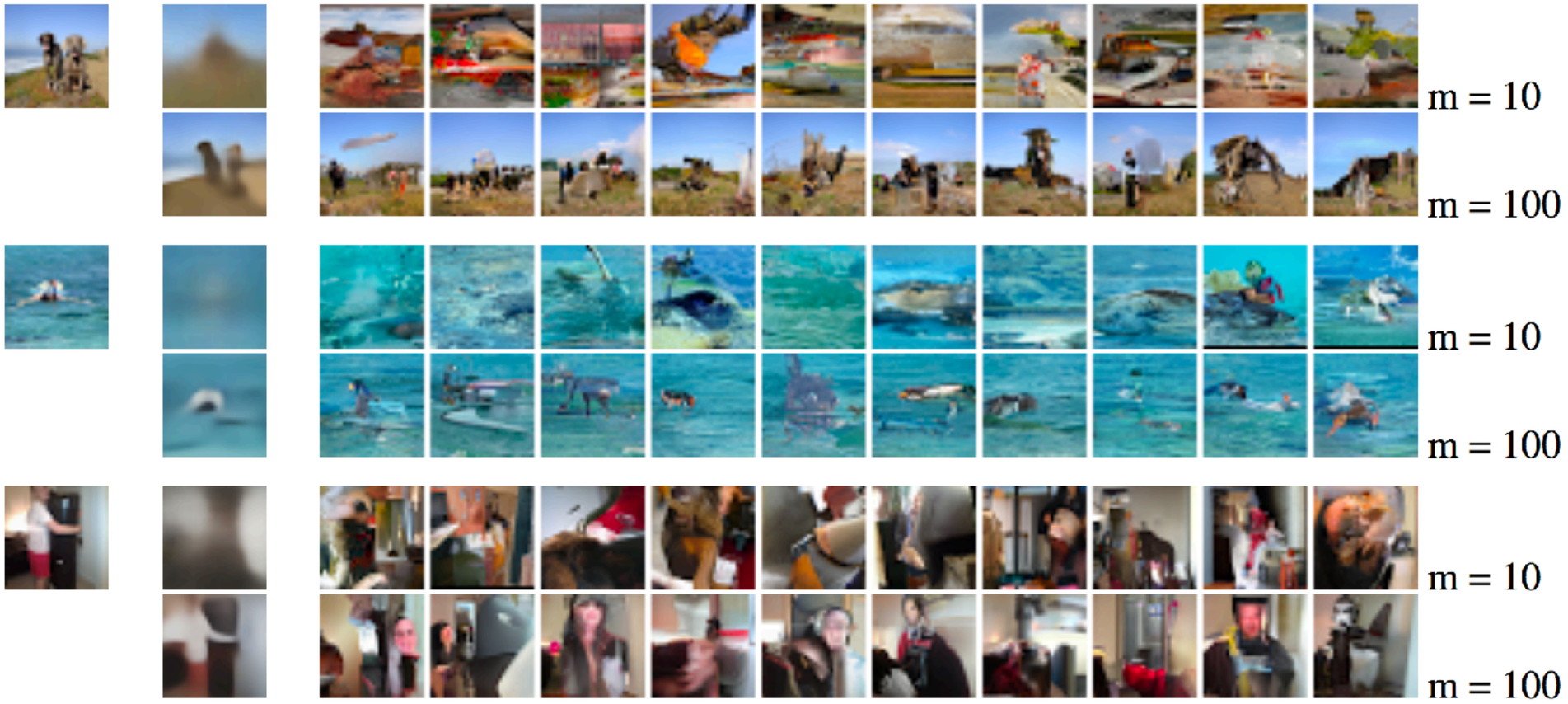
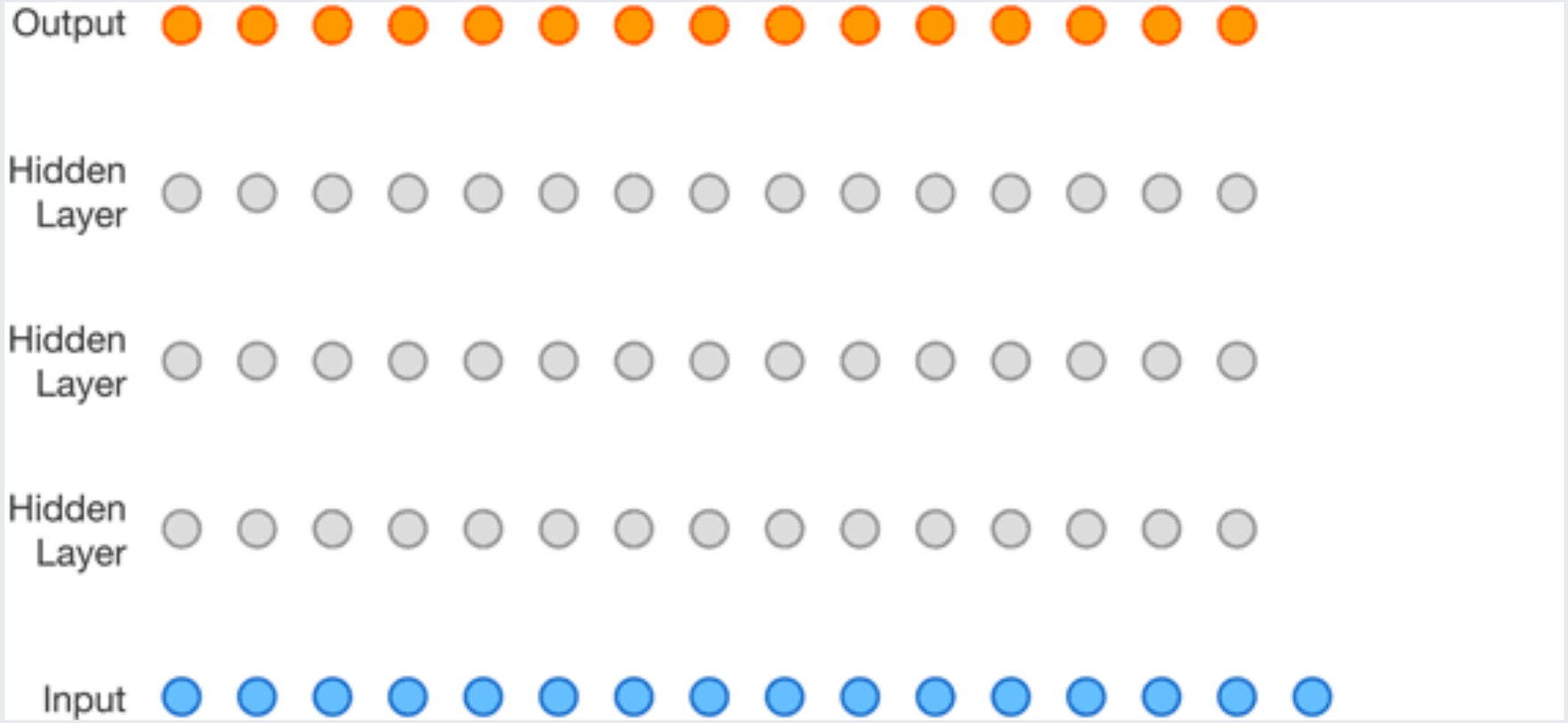


Figure 6: Left to right: original image, reconstruction by an auto-encoder trained with MSE, conditional samples from a PixelCNN auto-encoder. Both auto-encoders were trained end-to-end with a $m = 10$-dimensional bottleneck and a $m = 100$ dimensional bottleneck.

Wavenet



A Van Den Oord et. al. "Wavenet: A generative model for raw audio" (2016)

GLO: Optimizing the Latent Space of Generative Networks

Research question To model natural images with GANs, the generator and the discriminator are often parametrized as deep Convolutional Networks (convnets) [LeCun et al., 1998a]. Therefore, it is reasonable to hypothesize that the reasons for the success of GANs in modeling natural images come from two complementary sources:

(A1) Leveraging the powerful inductive bias of deep convnets.

(A2) The adversarial training protocol.

This work attempts to disentangle the factors of sucess (A1) and (A2) in GAN models. Specifically, we build an algorithm that relies on (A1), avoids (A2), and obtains competitive results when compared to a GAN.

Bojanowski et. al. 2017

GLO: Optimizing the Latent Space of Generative Networks

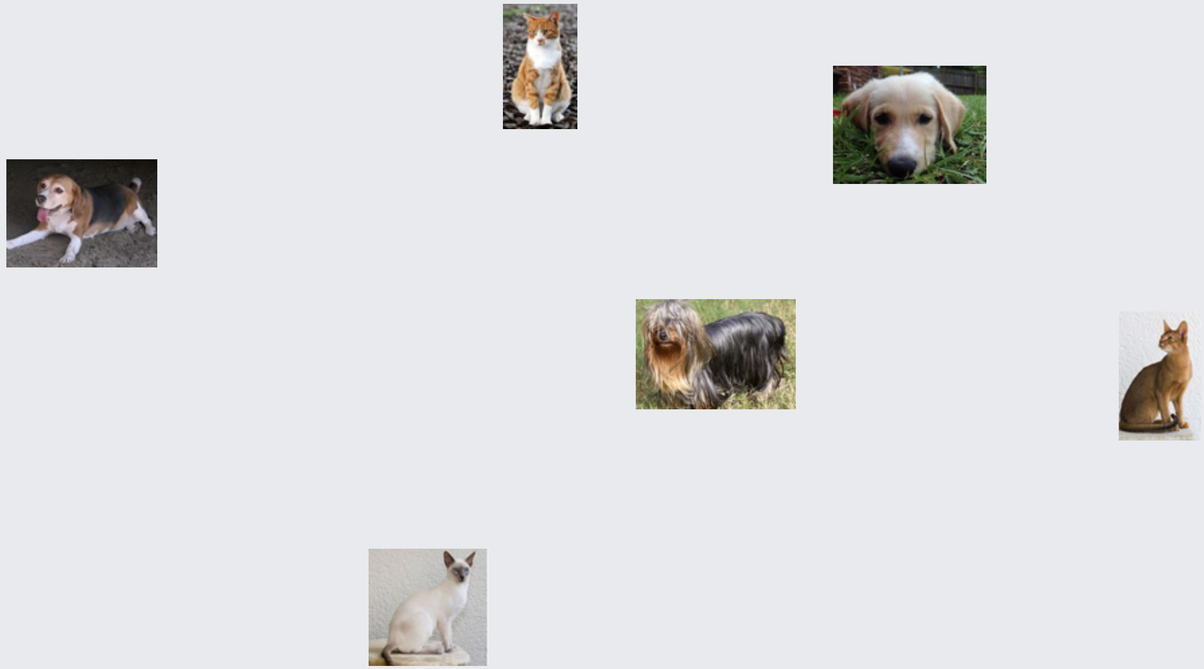
First, we consider a large set of images $\{x_1, \ldots, x_N\}$, where each image $x_i \in \mathcal{X}$ has dimensions $3 \times w \times h$. Second, we initialize a set of d-dimensional random vectors $\{z_1, \ldots, z_N\}$, where $z_i \in \mathcal{Z} \subseteq \mathbb{R}^d$ for all $i = 1, \ldots N$. Third, we pair the dataset of images with the random vectors, obtaining the dataset $\{(z_1, x_1), \ldots, (z_N, x_N)\}$. Finally, we jointly learn the parameters θ in Θ of a generator $g_\theta : \mathcal{Z} \to \mathcal{X}$ and the optimal noise vector z_i for each image x_i, by solving:

$$\min_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^{N} \left[\min_{z_i \in \mathcal{Z}} \ell\left(g_\theta(z_i), x_i\right) \right], \tag{1}$$

In the previous, $\ell : \mathcal{X} \times \mathcal{X}$ is a loss function measuring the reconstruction error from $g(z_i)$ to x_i. We call this model Generative Latent Optimization (GLO). Next, let us describe the most distinctive features of GLO.

Bojanowski et. al. 2017

GLO: Optimizing the Latent Space of Generative Networks

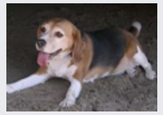


Bojanowski et. al. 2017

GLO: Optimizing the Latent Space of Generative Networks

Iteratively optimize!

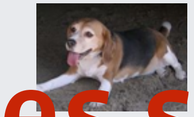





Bojanowski et. al. 2017

GLO: Optimizing the Latent Space of Generative Networks






images start in random positions

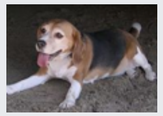
Bojanowski et. al. 2017

GLO: Optimizing the Latent Space of Generative Networks

Iteratively optimize:

1. bring similar images closer (optimize z)






Bojanowski et. al. 2017

Open problems

- Stability of GANs
- Evaluation of generative models
  - Read Lucas Theis et. al. "A note on the evaluation of generative models" (2016)
- long-term video prediction