Generative models

Soumith Chintala

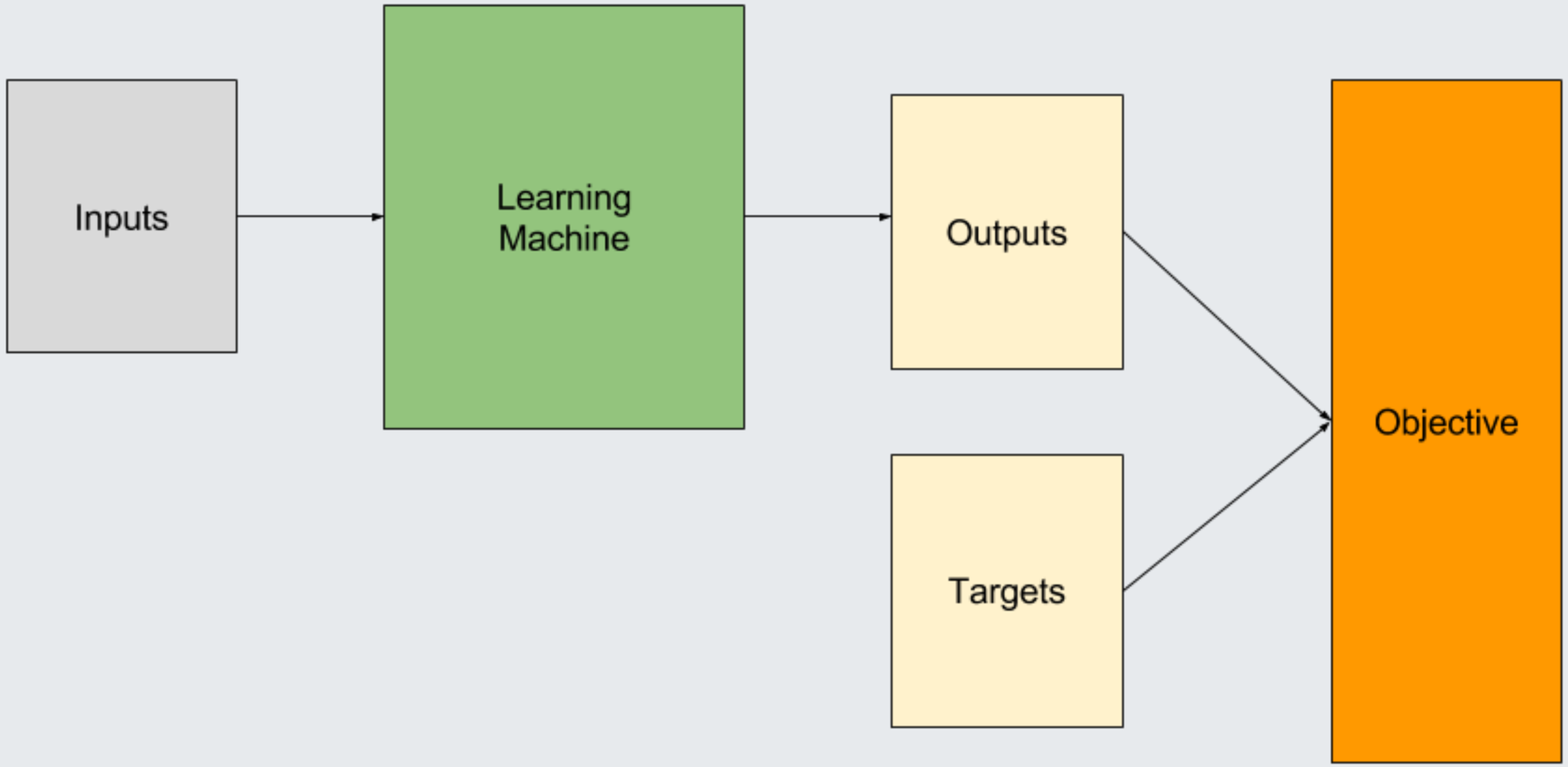
Facebook AI Research

Acknowledgments

- Significant material borrowed from:
  - cs231n lecture on generative models:
    - Fei-Fei Li, Justin Johnson, Serena Yeung
  - Papers that we are covering

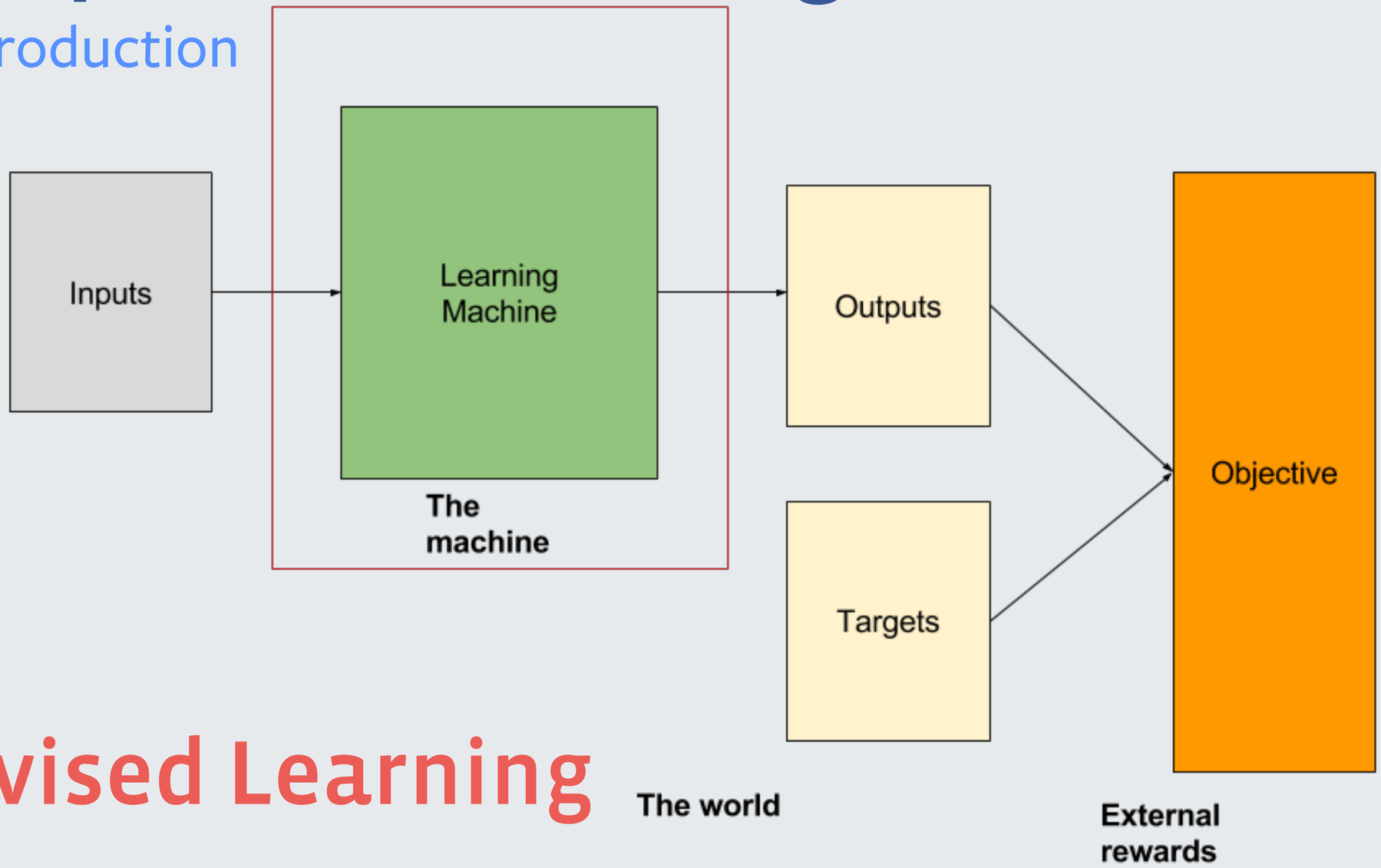
Unsupervised Learning

An introduction



Unsupervised Learning

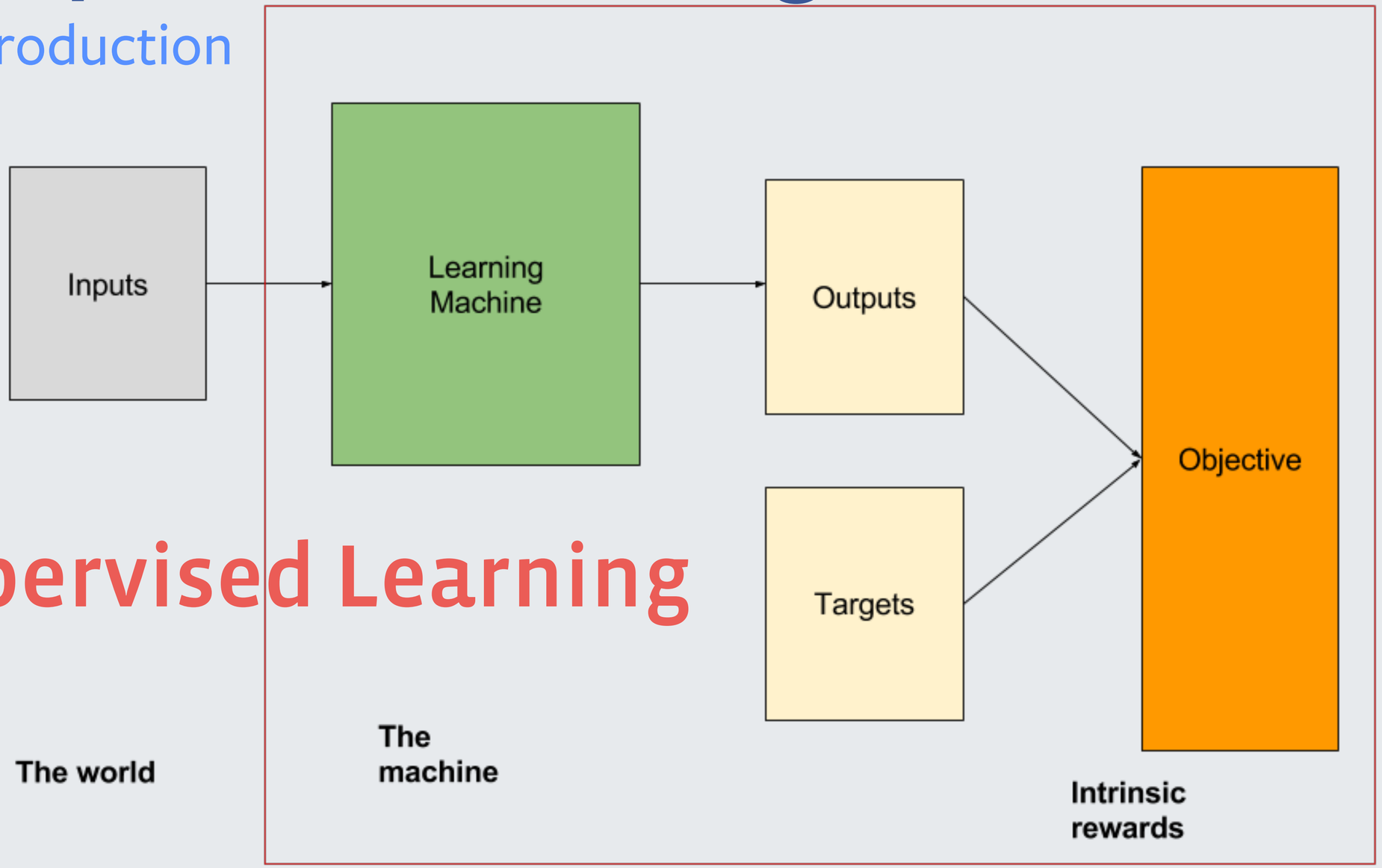
An introduction



Supervised Learning

Unsupervised Learning

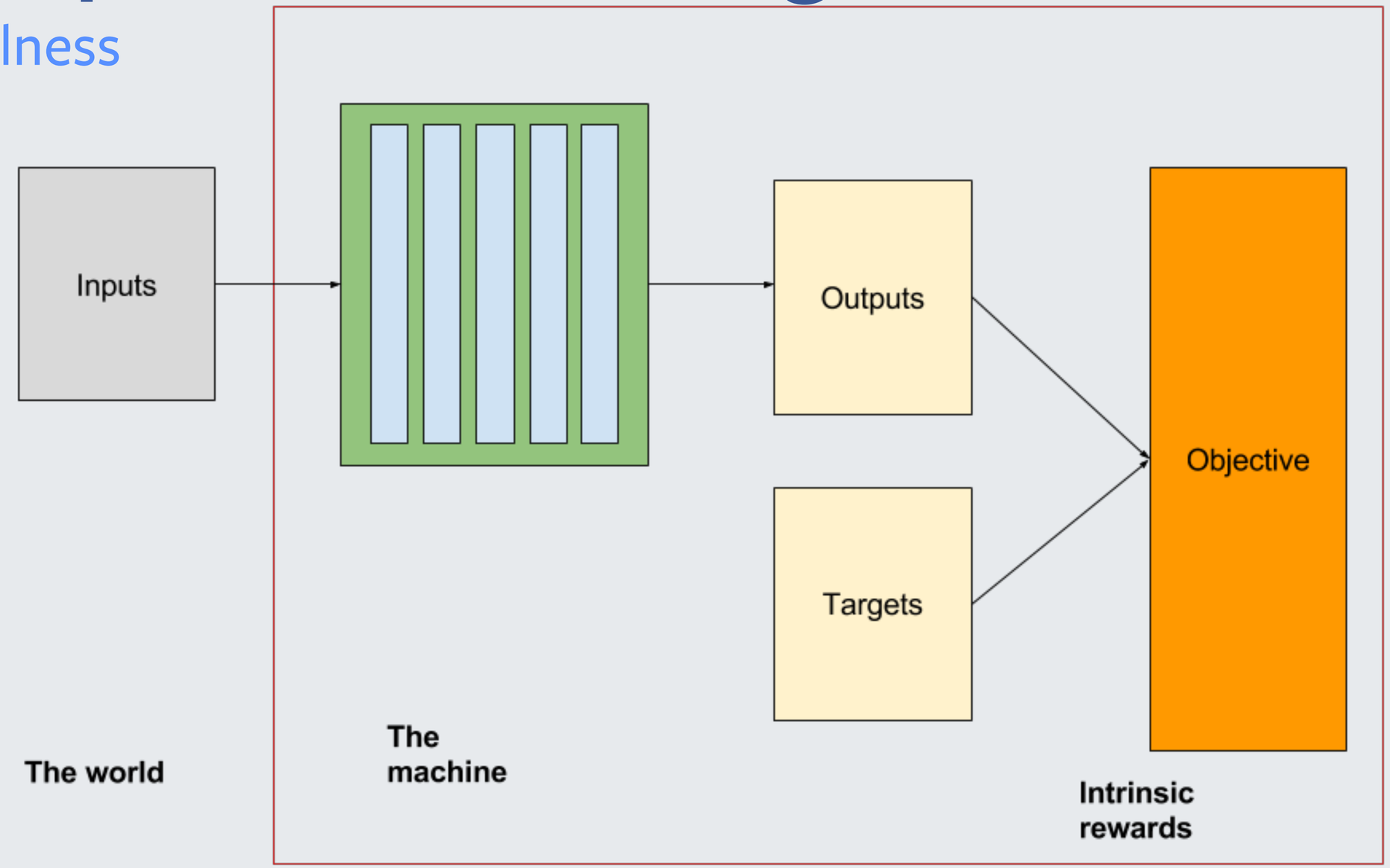
An introduction



Unsupervised Learning

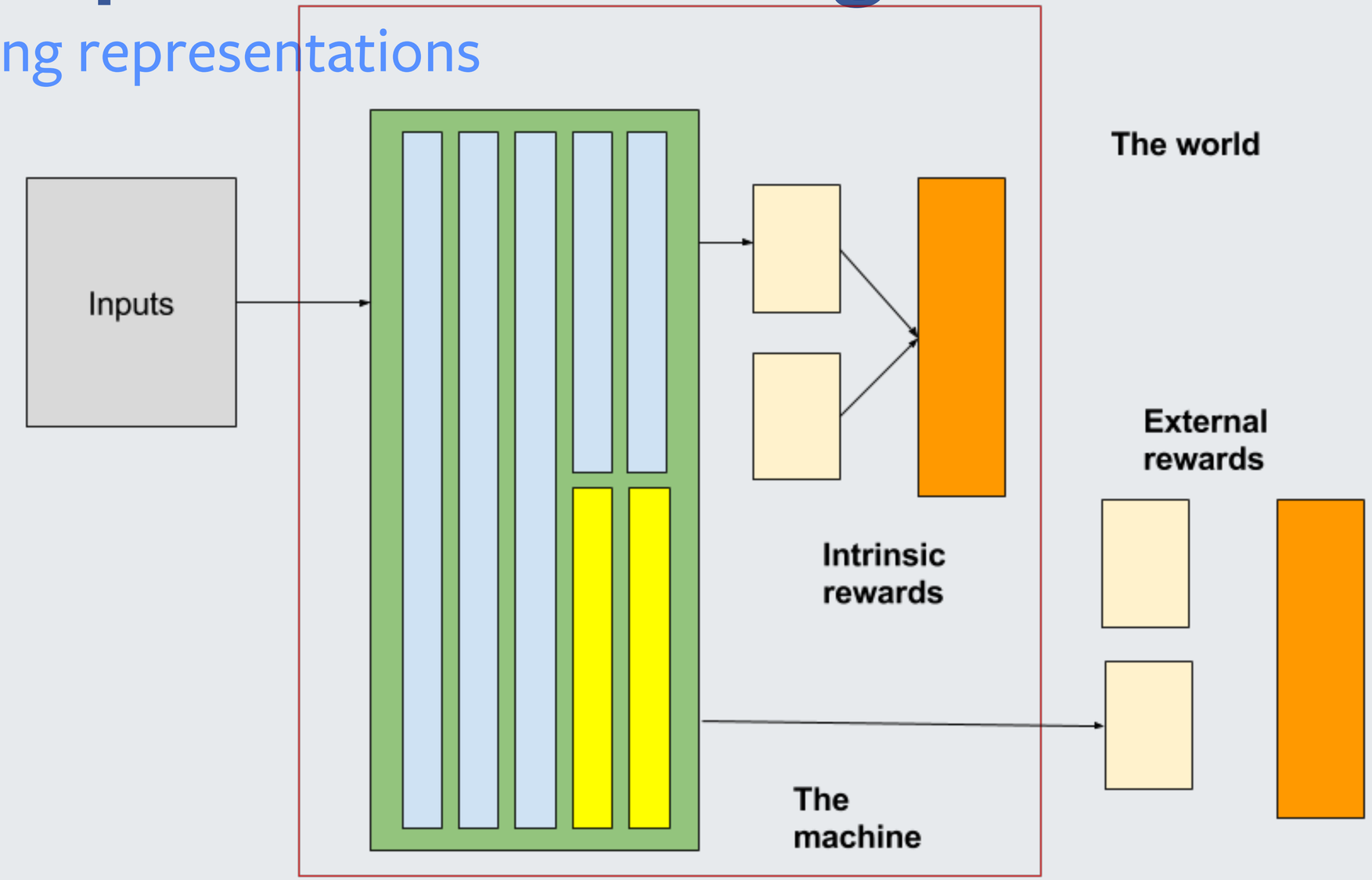
Unsupervised Learning

Usefulness



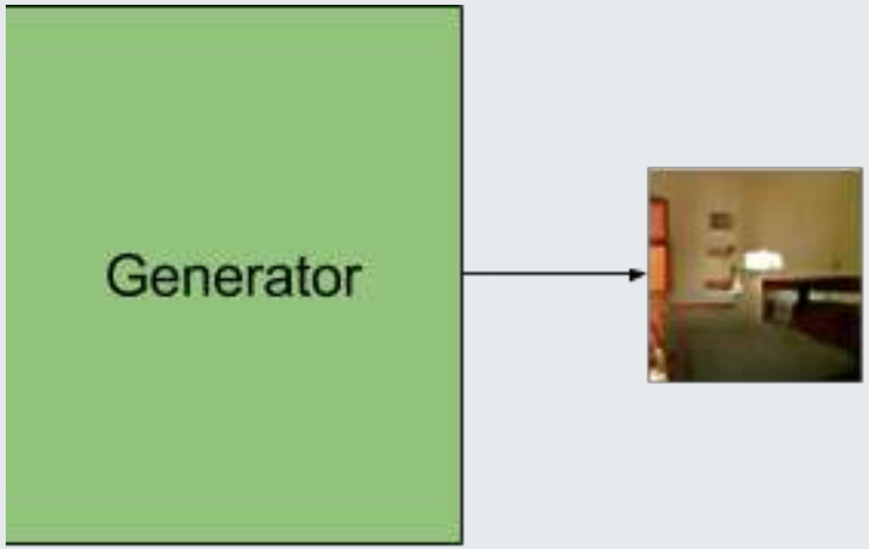
Unsupervised Learning

Reusing representations



Generative Models

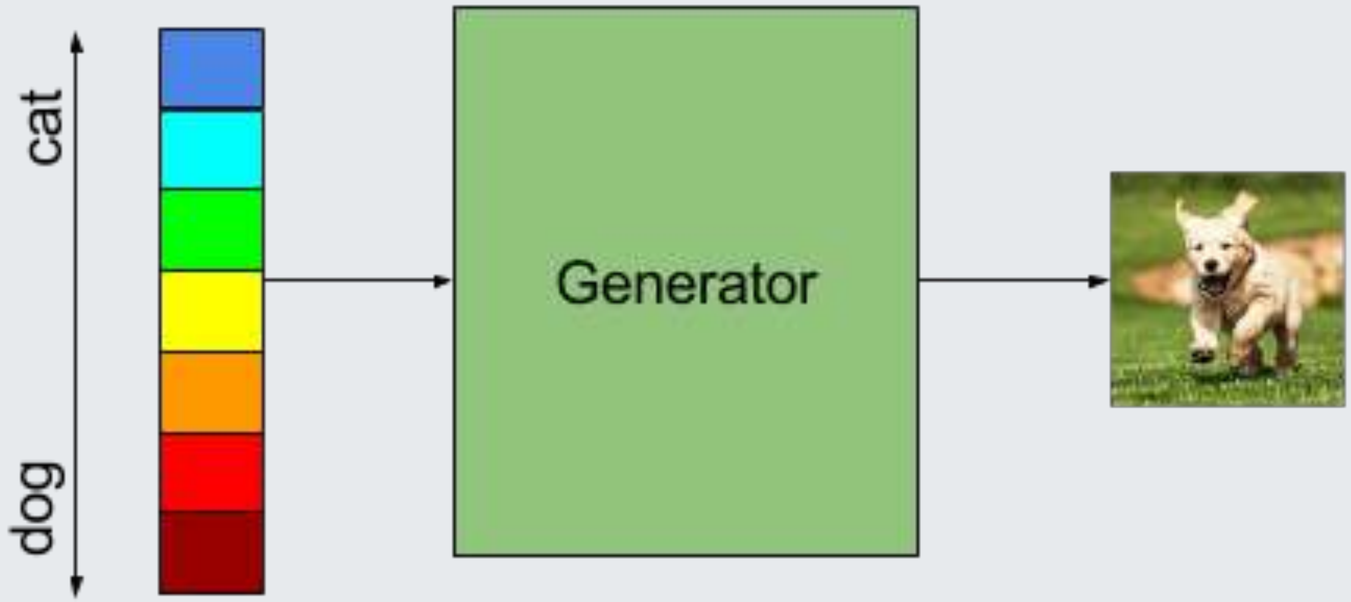
An introduction



A model that learns a distribution of images

Generative Models

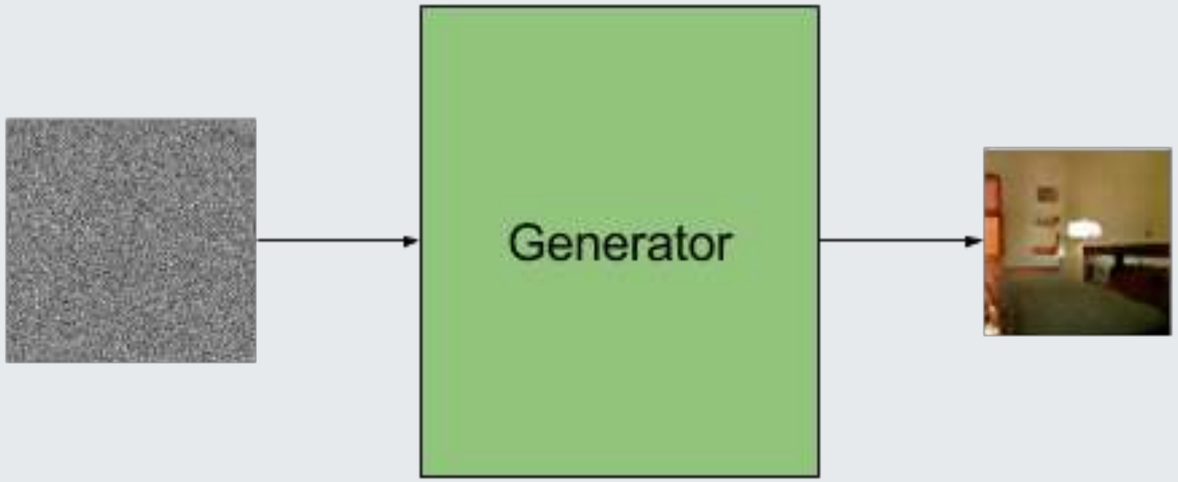
An introduction



X = P(z), z controls dogness or catness

Generative Models

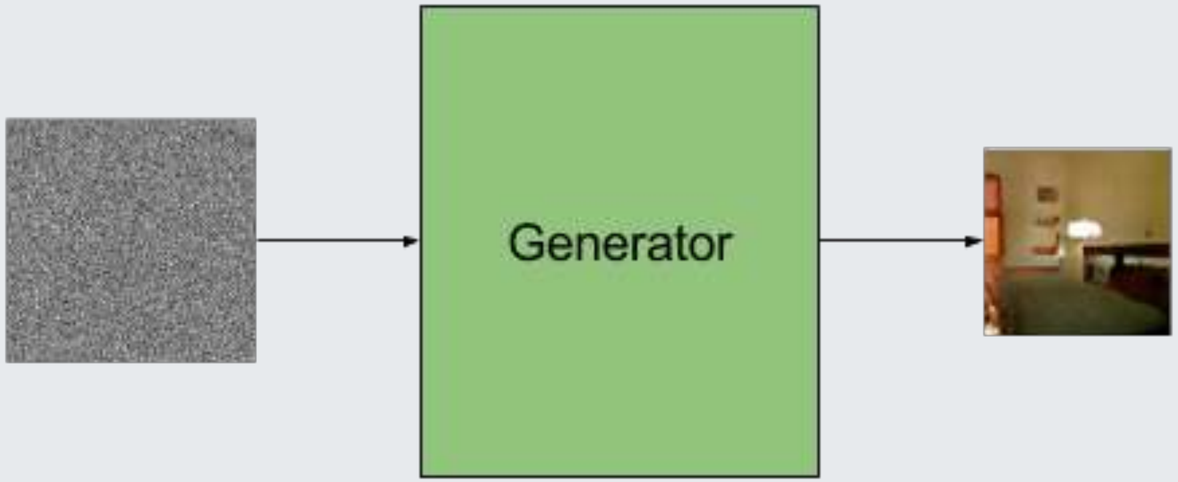
An introduction



X = P(z), z is a latent variable

Generative Models

An introduction



P(z) = neural network

Generative Models

Given training data, generate new samples from same distribution

Training data ~ $p_{data}(x)$

Generated samples ~ $p_{model}(x)$

Want to learn $p_{model}(x)$ similar to $p_{data}(x)$

Addresses density estimation, a core problem in unsupervised learning

Several flavors:

- Explicit density estimation: explicitly define and solve for $p_{model}(x)$
- Implicit density estimation: learn model that can sample from $p_{model}(x)$ w/o explicitly defining it

Taxonomy of Generative Models

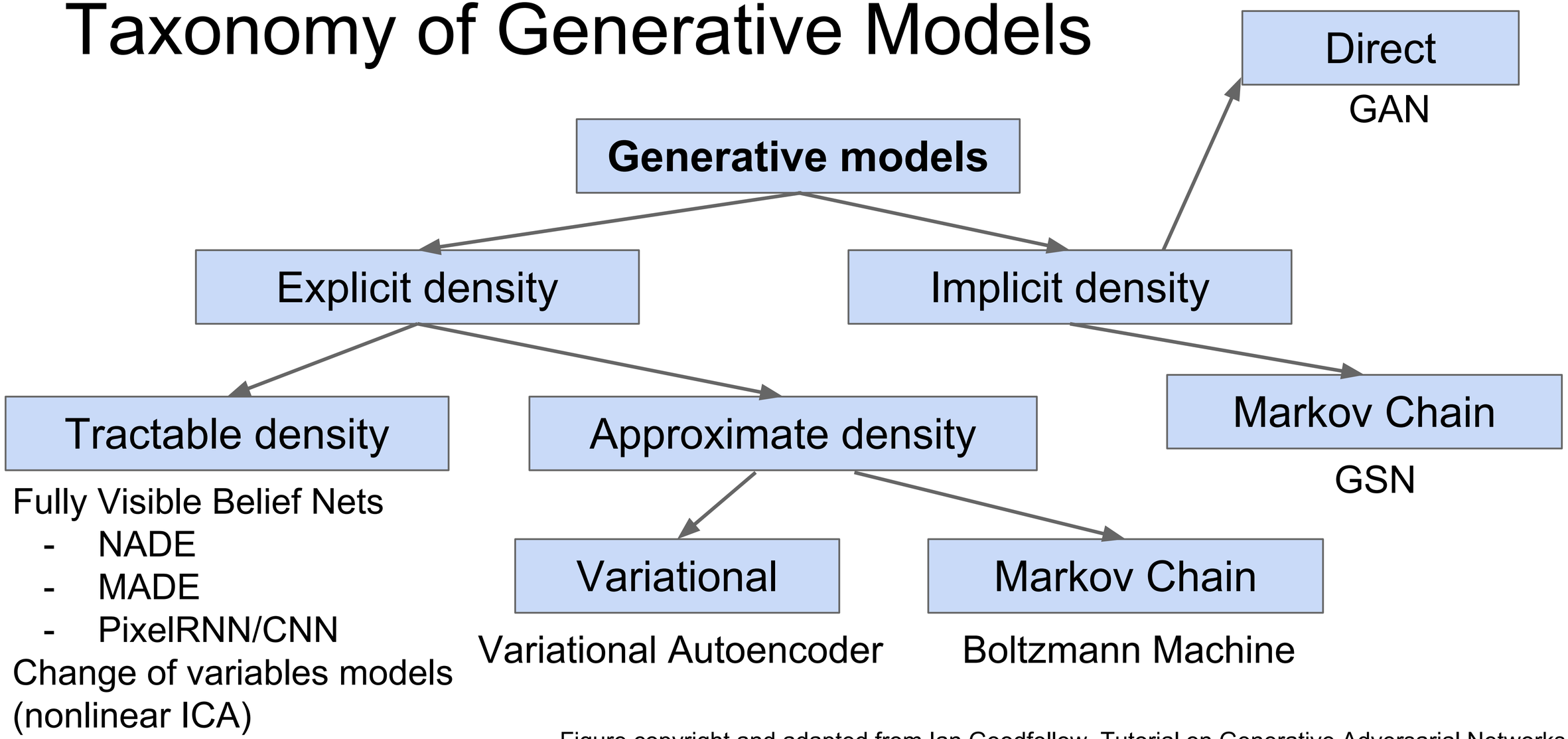


Figure copyright and adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

Taxonomy of Generative Models

Today: discuss 3 most popular types of generative models today

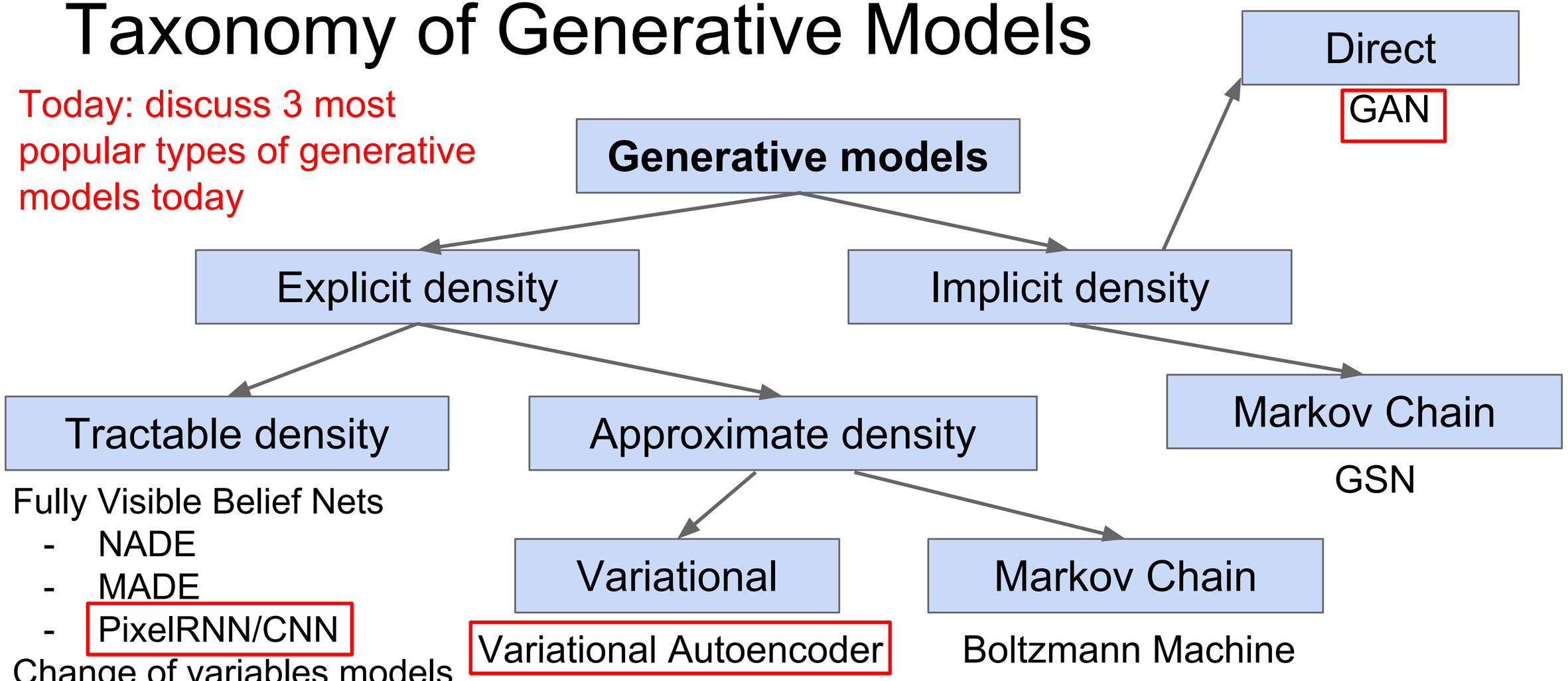


Figure copyright and adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

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

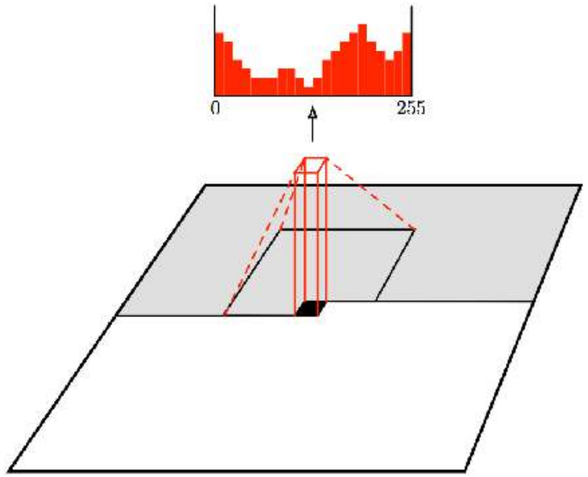
Oriol Vinyals
Google DeepMind
vinyals@google.com

Lasse Espeholt
Google DeepMind
espeholt@google.com

Alex Graves
Google DeepMind
gravesa@google.com

Koray Kavukcuoglu
Google DeepMind
korayk@google.com

Pixel-CNN



1	1	1	1	1
1	1	1	1	1
1	1	0	0	0
0	0	0	0	0
0	0	0	0	0

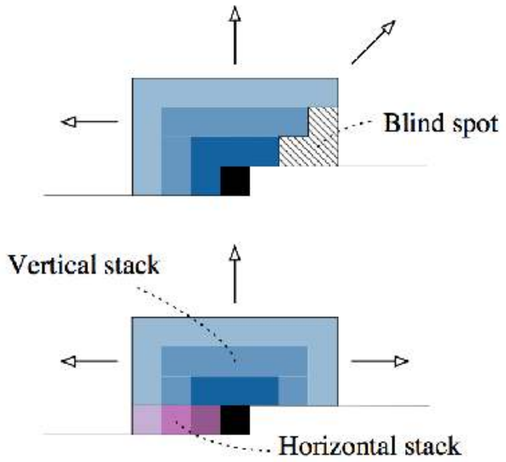


Figure 1: **Left**: A visualization of the PixelCNN that maps a neighborhood of pixels to prediction for the next pixel. To generate pixel x_i the model can only condition on the previously generated pixels $x_1, \ldots x_{i-1}$. **Middle**: an example matrix that is used to mask the 5x5 filters to make 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

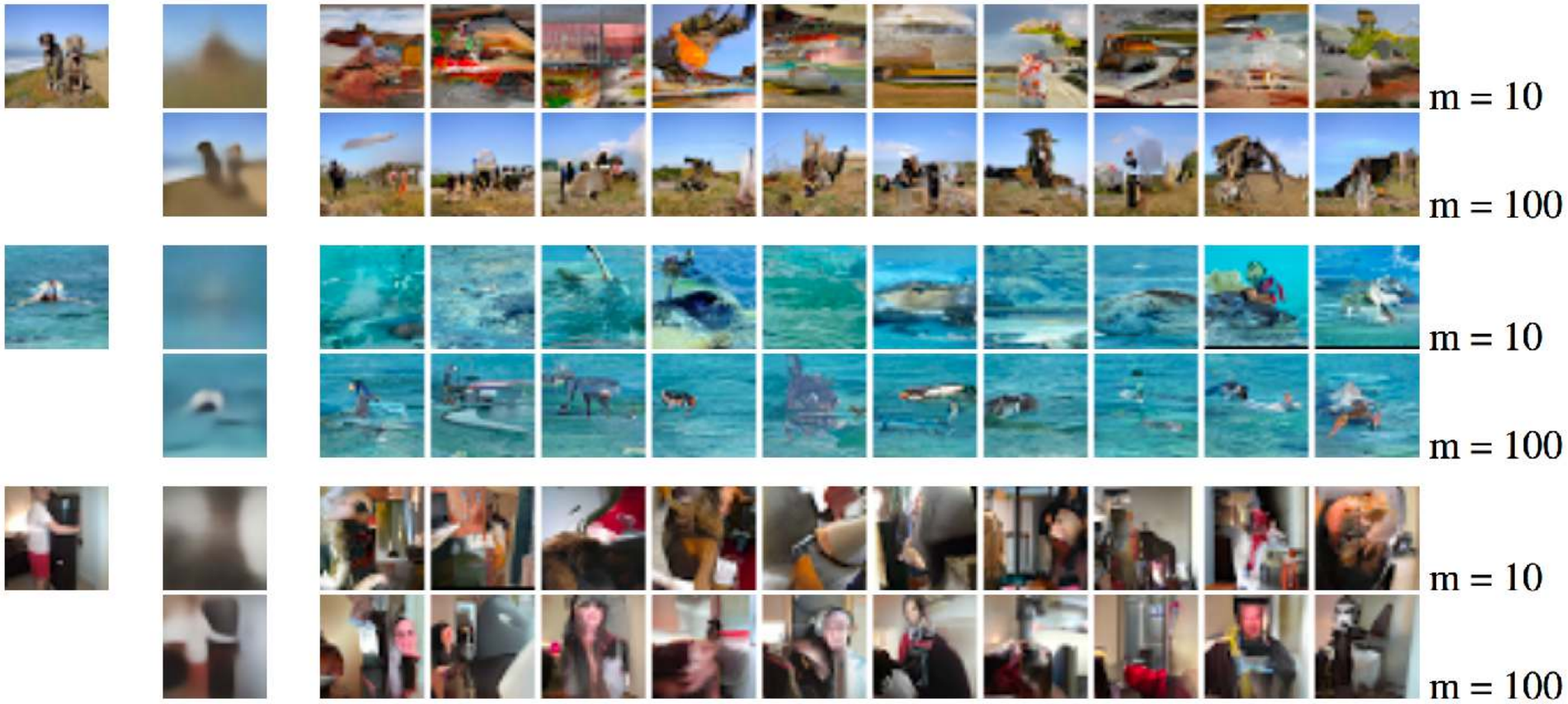
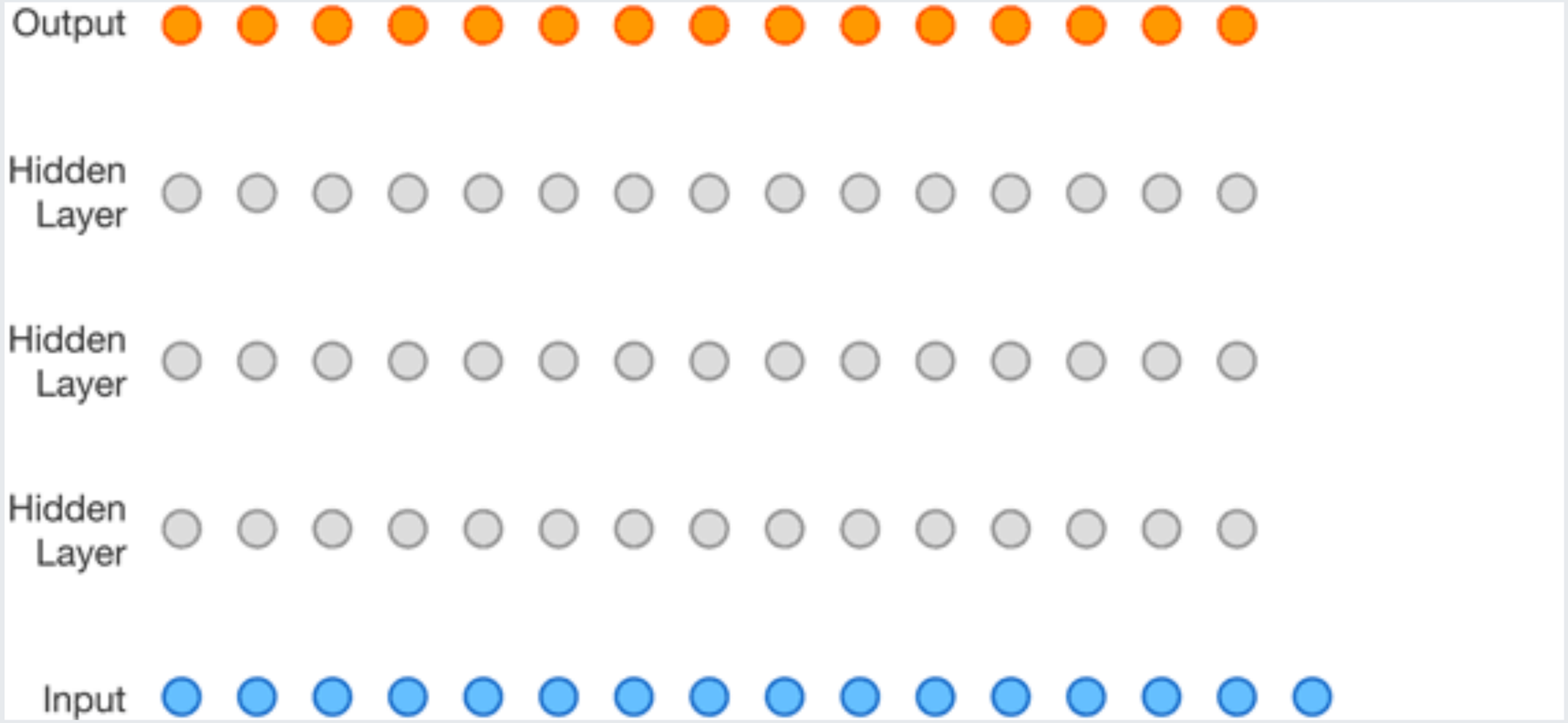


Figure 6: Left to right: original image, reconstruction by an auto-encoder trained with MSE, conditional samples from a PixelCNN auto-encoder. Both auto-encoders were trained end-to-end with a $m = 10$-dimensional bottleneck and a $m = 100$ dimensional bottleneck.

Wavenet



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

PixelRNN and PixelCNN

Pros:

- Can explicitly compute likelihood p(x)
- Explicit likelihood of training data gives good evaluation metric
- Good samples

Con:

- Sequential generation => slow

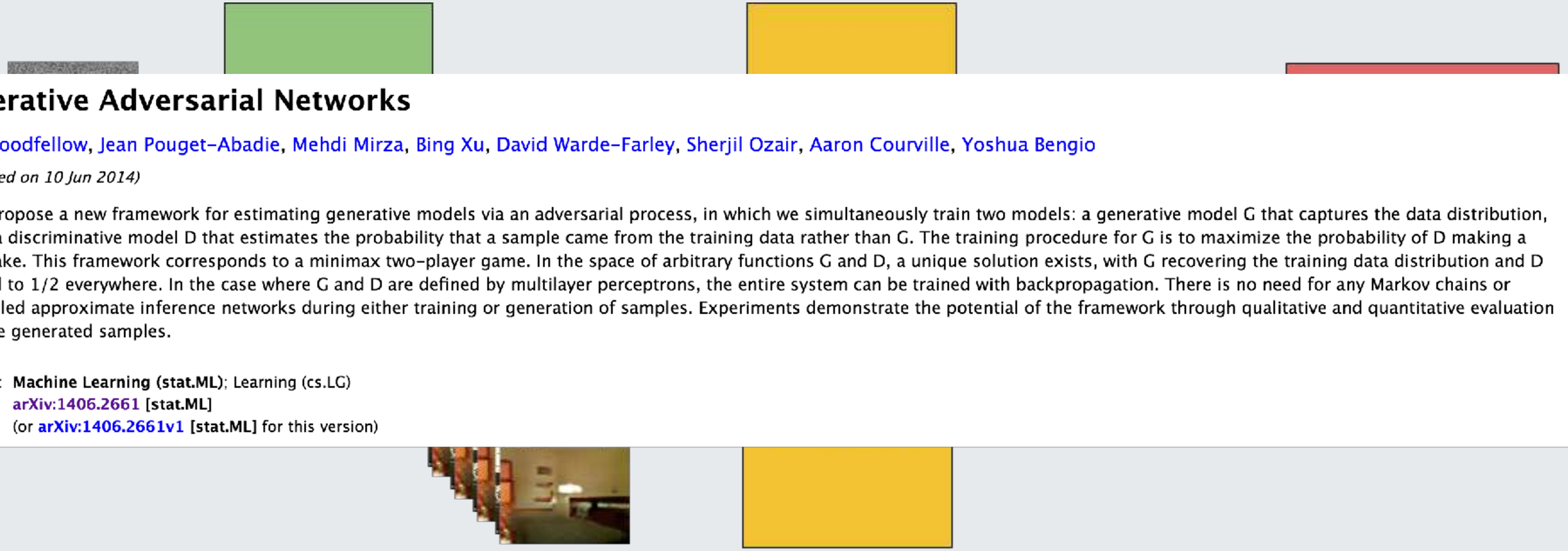
Improving PixelCNN performance

- Gated convolutional layers
- Short-cut connections
- Discretized logistic loss
- Multi-scale
- Training tricks
- Etc…

See

- Van der Oord et al. NIPS 2016
- Salimans et al. 2017 (PixelCNN++)

Generative Adversarial Networks



Generative Adversarial Networks

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio

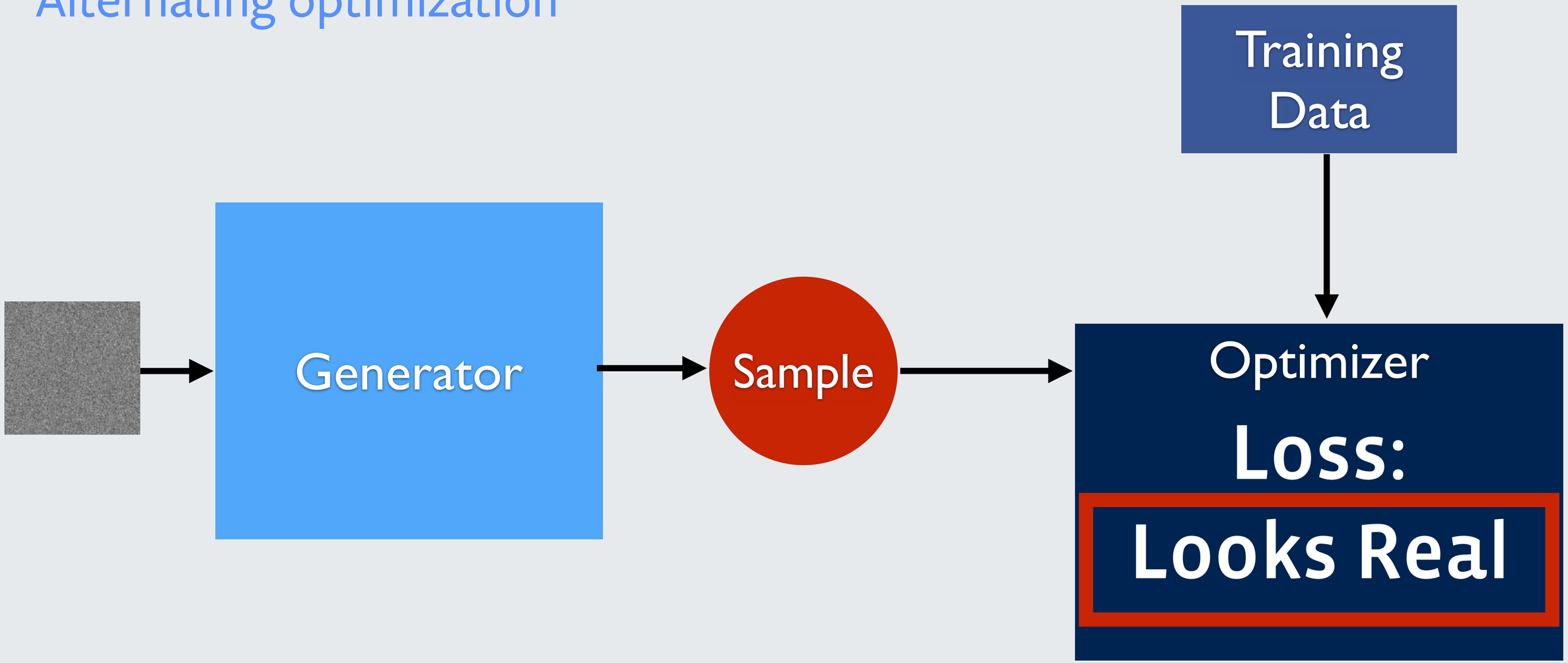
(Submitted on 10 Jun 2014)

We propose a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D, a unique solution exists, with G recovering the training data distribution and D equal to 1/2 everywhere. In the case where G and D are defined by multilayer perceptrons, the entire system can be trained with backpropagation. There is no need for any Markov chains or unrolled approximate inference networks during either training or generation of samples. Experiments demonstrate the potential of the framework through qualitative and quantitative evaluation of the generated samples.

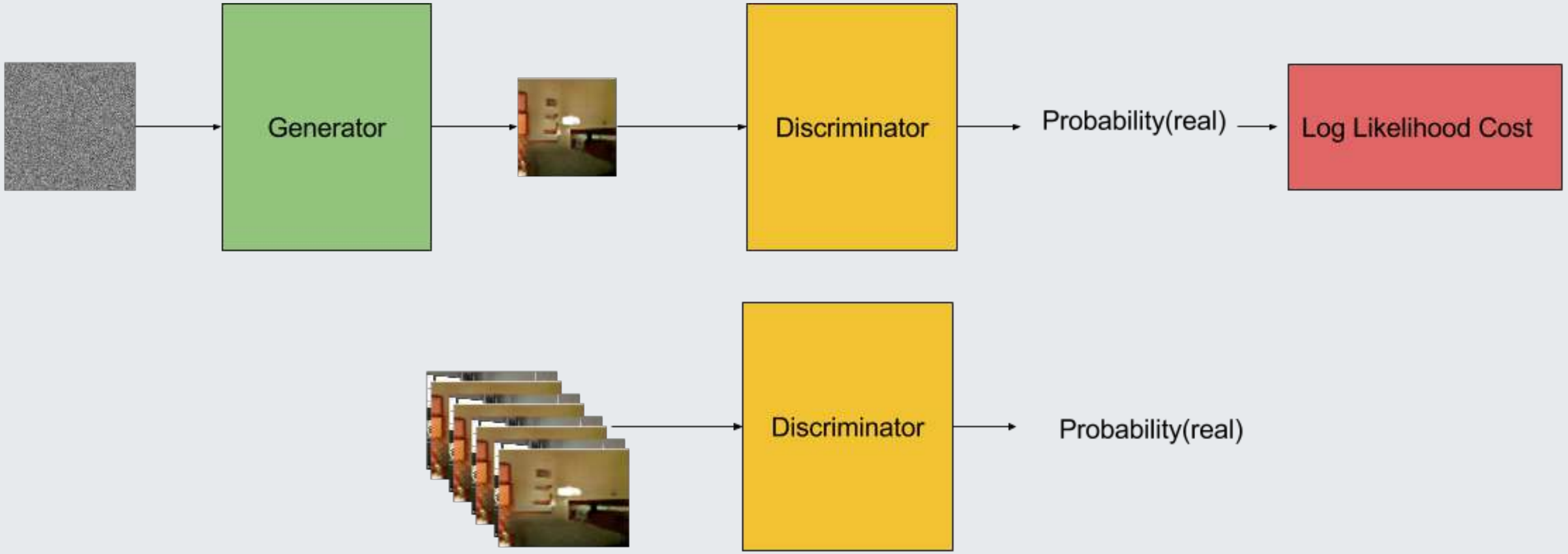
Subjects: **Machine Learning (stat.ML)**; Learning (cs.LG)
Cite as: **arXiv:1406.2661 [stat.ML]**
(or **arXiv:1406.2661v1 [stat.ML]** for this version)

Generative Adversarial Networks

Alternating optimization

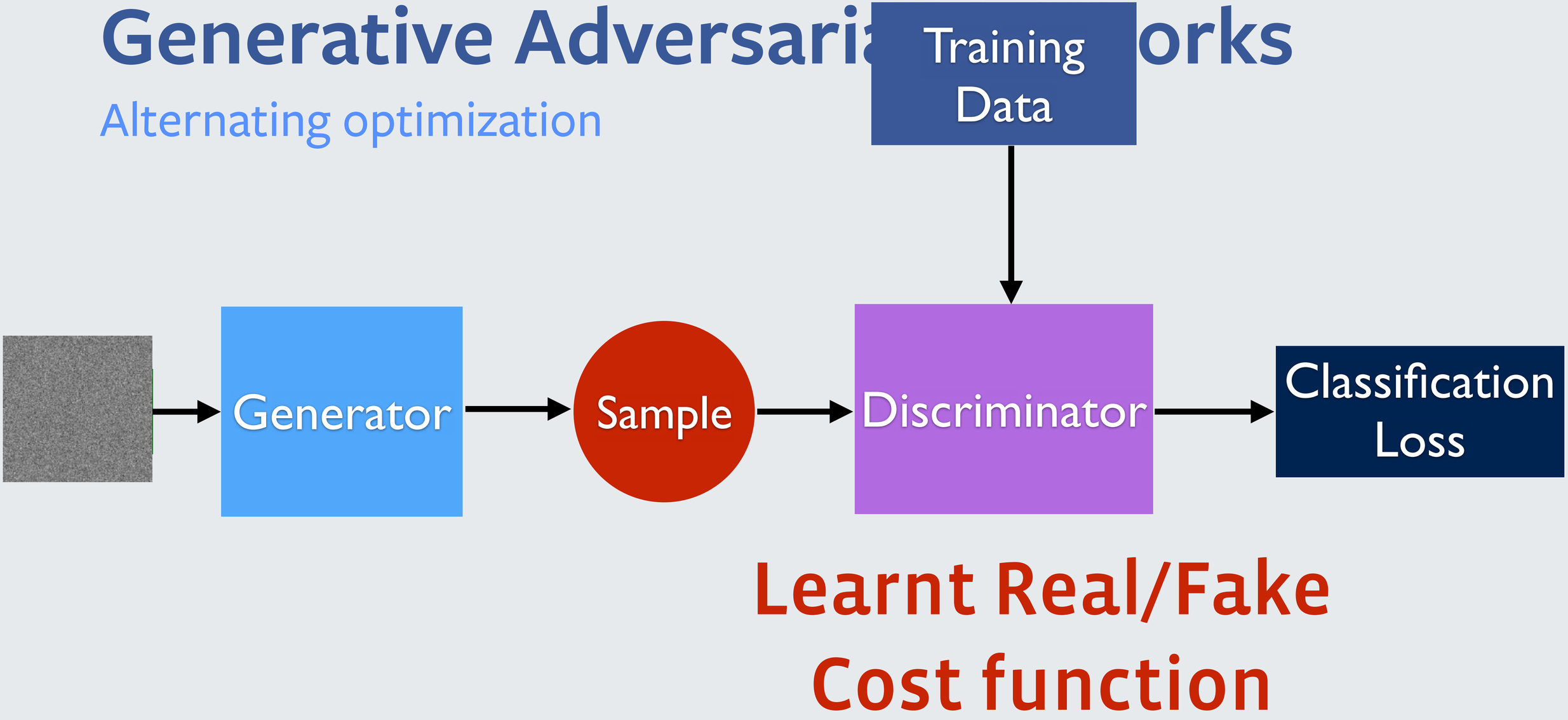


Generative Adversarial Networks



Generative Adversari orks

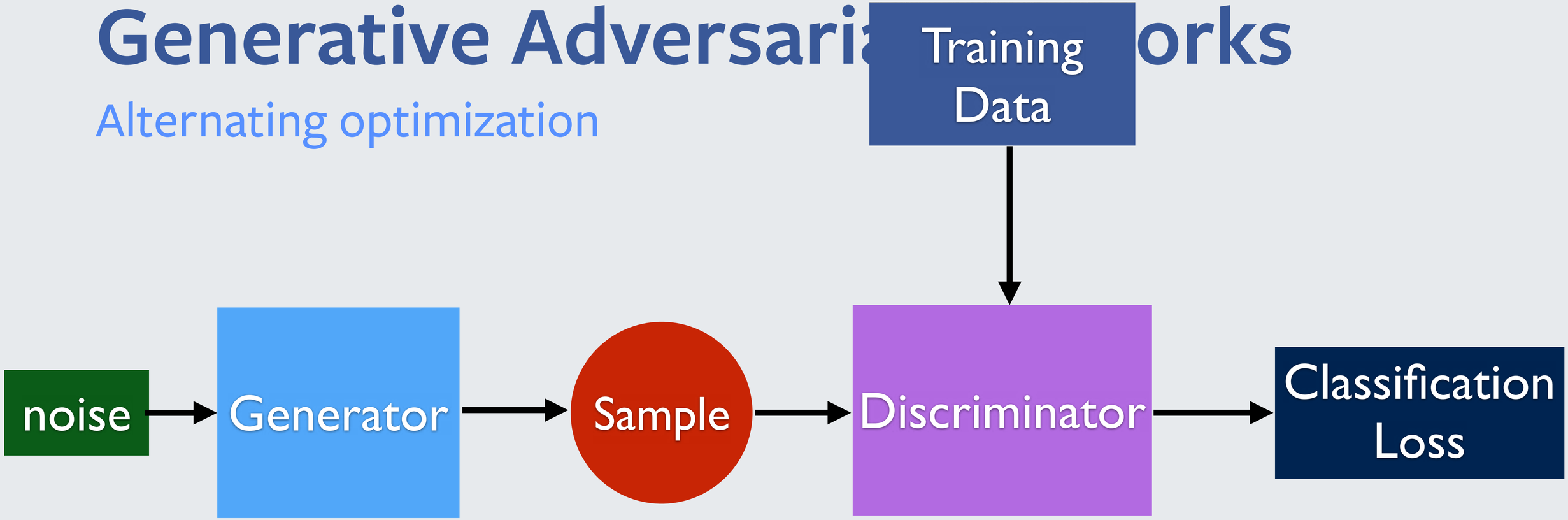
Alternating optimization



Learnt Real/Fake Cost function

Generative Adversari orks

Alternating optimization

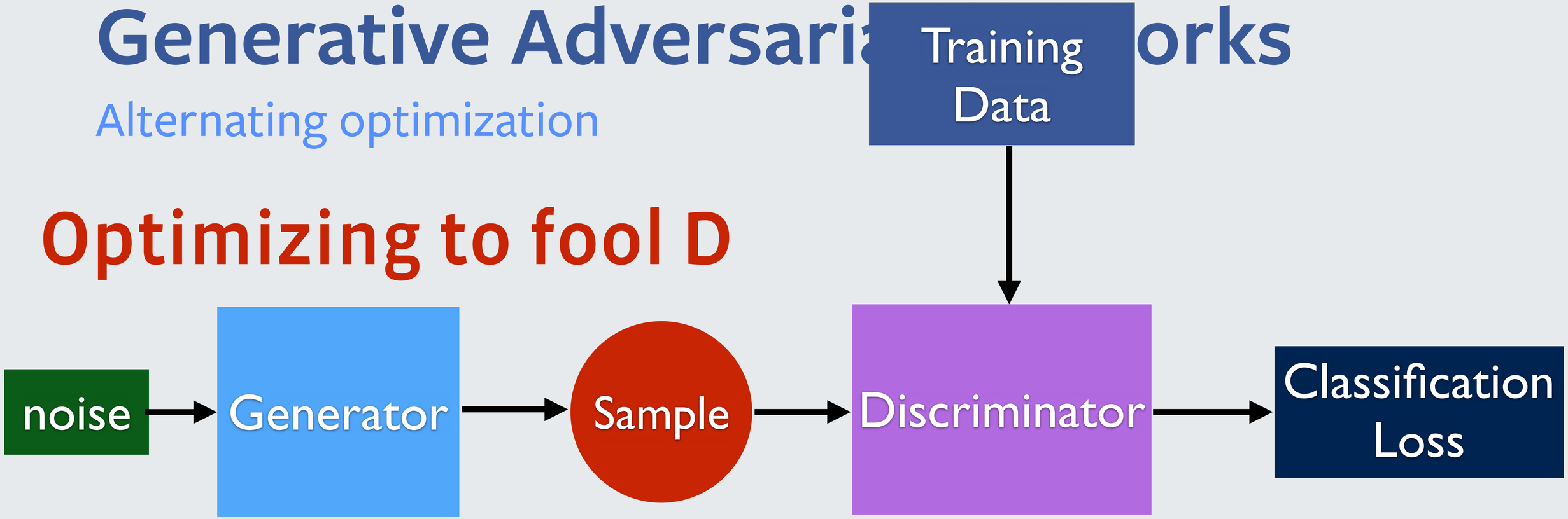


Trained via Gradient Descent

Generative Adversari orks

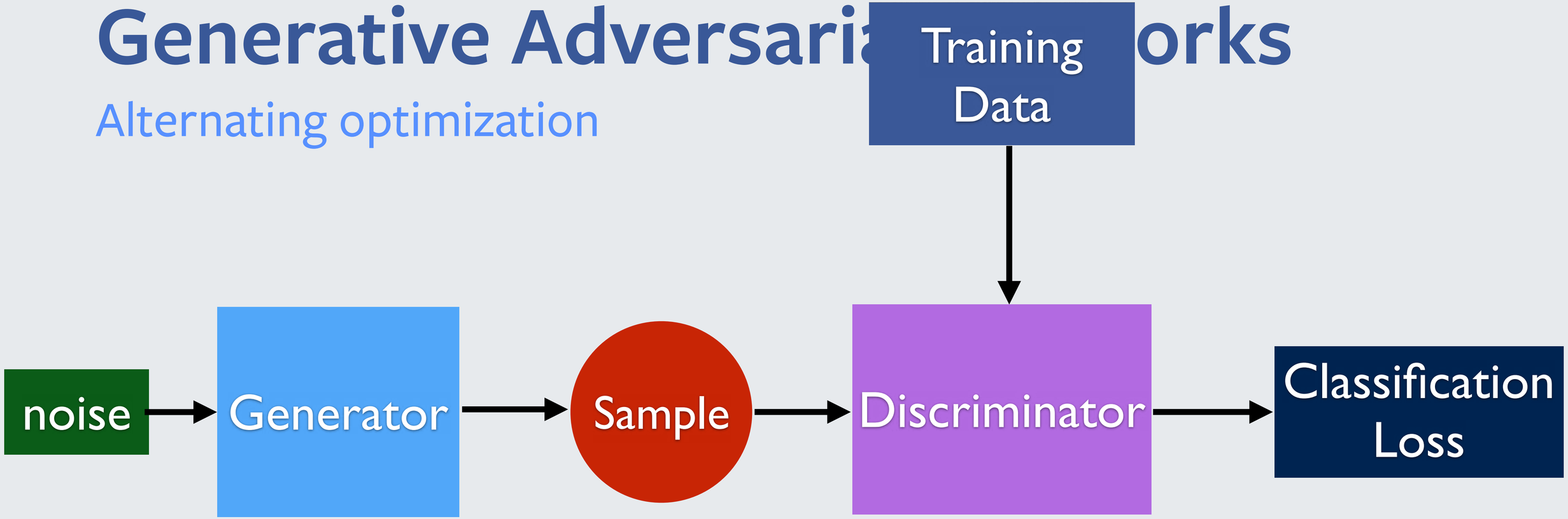
Alternating optimization

Optimizing to fool D



Generative Adversari orks

Alternating optimization



Optimizing to not get fooled by G

Generative Adversarial Networks

Optimizes Jensen-Shannon Divergence

Theorem 1. *The global minimum of the virtual training criterion $C(G)$ is achieved if and only if $p_g = p_{data}$. At that point, $C(G)$ achieves the value $-\log 4$.*

Proof. For $p_g = p_{\text{data}}$, $D_G^*(\boldsymbol{x}) = \frac{1}{2}$, (consider Eq. 2). Hence, by inspecting Eq. 4 at $D_G^*(\boldsymbol{x}) = \frac{1}{2}$, we find $C(G) = \log \frac{1}{2} + \log \frac{1}{2} = -\log 4$. To see that this is the best possible value of $C(G)$, reached only for $p_g = p_{\text{data}}$, observe that

$$\mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}} [-\log 2] + \mathbb{E}_{\boldsymbol{x} \sim p_g} [-\log 2] = -\log 4$$

and that by subtracting this expression from $C(G) = V(D_G^*, G)$, we obtain:

$$C(G) = -\log(4) + KL\left(p_{\text{data}} \middle\| \frac{p_{\text{data}} + p_g}{2}\right) + KL\left(p_g \middle\| \frac{p_{\text{data}} + p_g}{2}\right) \tag{5}$$

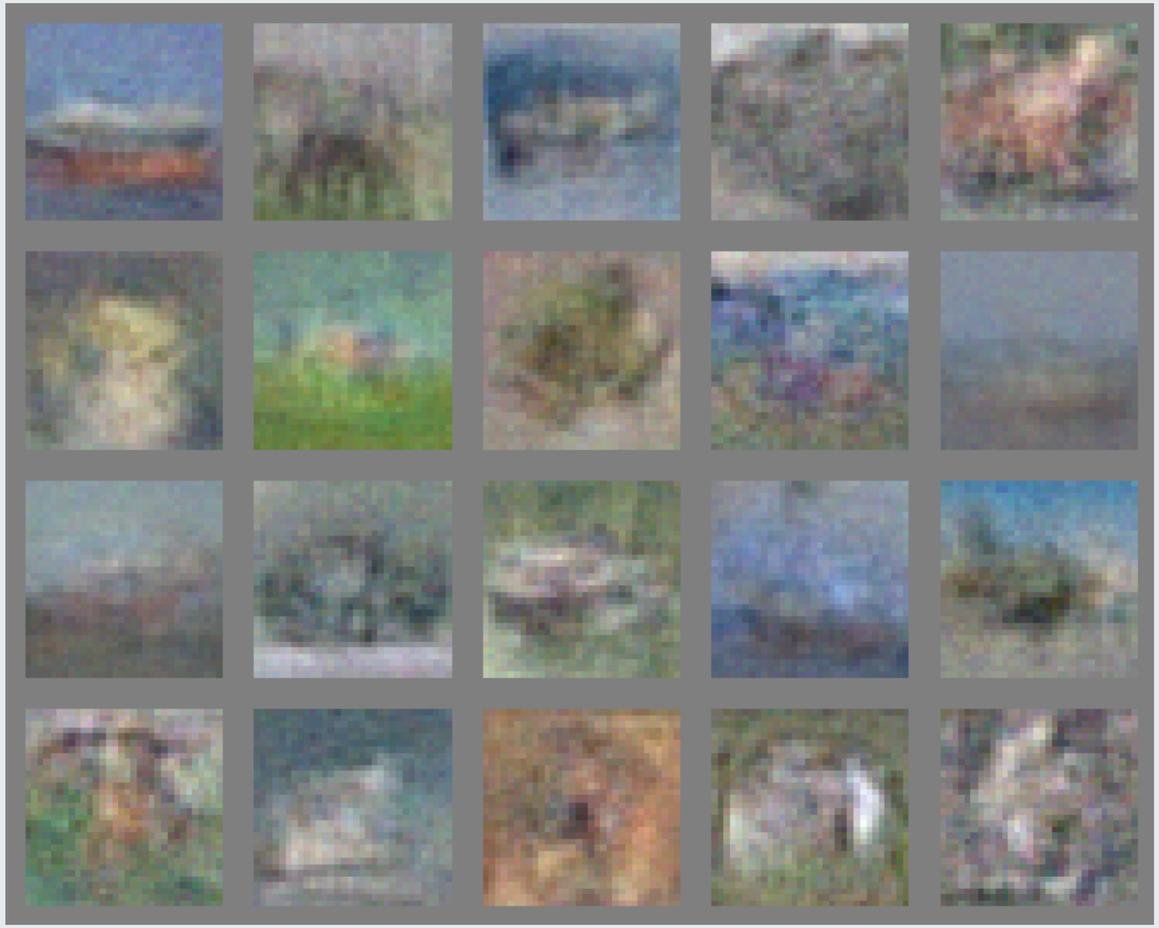
where KL is the Kullback–Leibler divergence. We recognize in the previous expression the Jensen–Shannon divergence between the model's distribution and the data generating process:

$$C(G) = -\log(4) + 2 \cdot JSD\left(p_{\text{data}} \| p_g\right) \tag{6}$$

Since the Jensen–Shannon divergence between two distributions is always non-negative and zero only when they are equal, we have shown that $C^* = -\log(4)$ is the global minimum of $C(G)$ and that the only solution is $p_g = p_{\text{data}}$, i.e., the generative model perfectly replicating the data generating process. □

Generative Adversarial Networks

Samples



Class-conditional GANs

arxiv.org/abs/1506.05751

Cornell University Library

arXiv.org > cs > arXiv:1506.05751

Search

Computer Science > Computer Vision and Pattern Recognition

Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks

Emily Denton, Soumith Chintala, Arthur Szlam, Rob Fergus

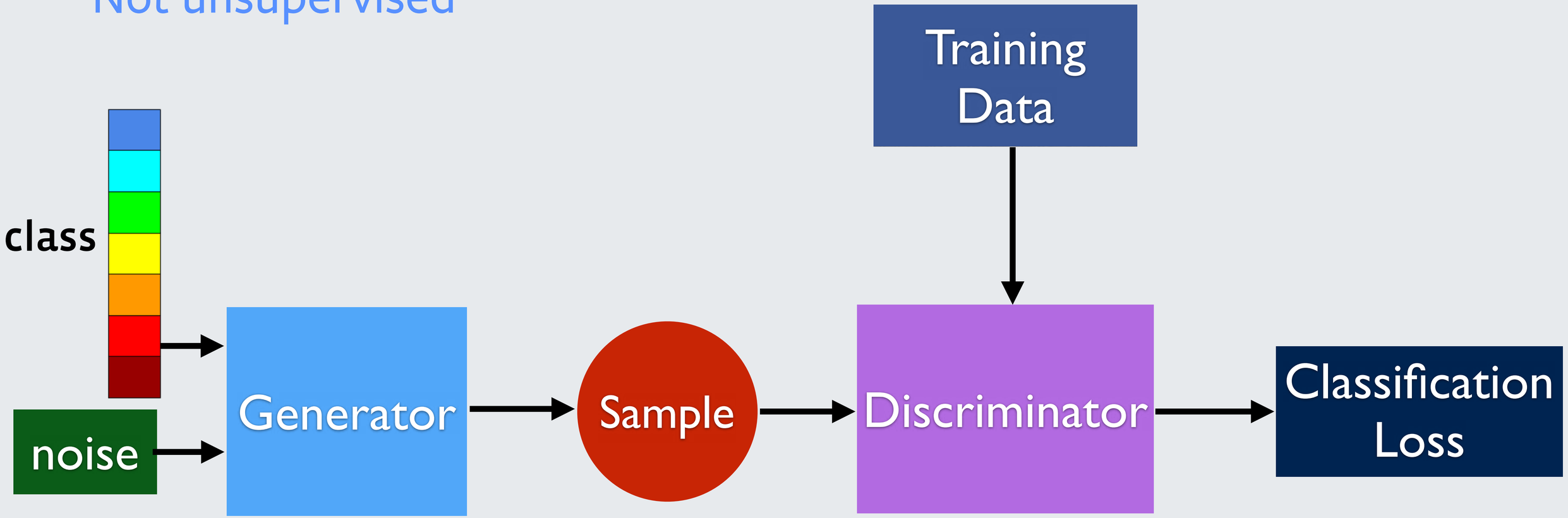
(Submitted on 18 Jun 2015)

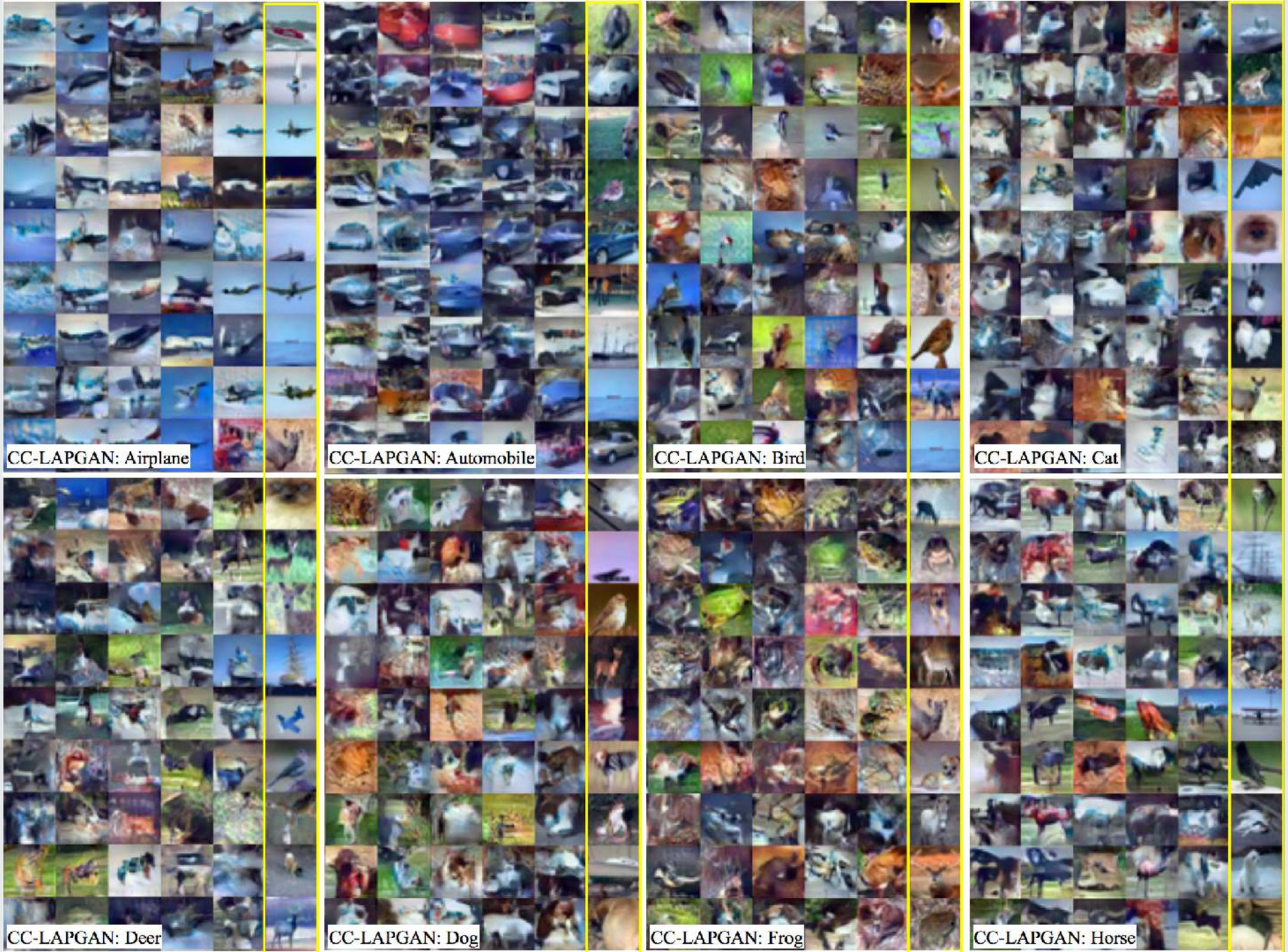
In this paper we introduce a generative parametric model capable of producing high quality samples of natural images. Our approach uses a cascade of convolutional networks within a Laplacian pyramid framework to generate images in a coarse-to-fine fashion. At each level of the pyramid, a separate generative convnet model is trained using the Generative Adversarial Nets (GAN) approach (Goodfellow et al.). Samples drawn from our model are of significantly higher quality than alternate approaches. In a quantitative assessment by human evaluators, our CIFAR10 samples were mistaken for real images around 40% of the time, compared to 10% for samples drawn from a GAN baseline model. We also show samples from models trained on the higher resolution images of the LSUN scene dataset.

Subjects: **Computer Vision and Pattern Recognition (cs.CV)**
Cite as: **arXiv:1506.05751 [cs.CV]**
(or **arXiv:1506.05751v1 [cs.CV]** for this version)

Class-conditional GANs

Not unsupervised




CC-LAPGAN: Airplane
CC-LAPGAN: Automobile
CC-LAPGAN: Bird
CC-LAPGAN: Cat
CC-LAPGAN: Deer
CC-LAPGAN: Dog
CC-LAPGAN: Frog
CC-LAPGAN: Horse

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

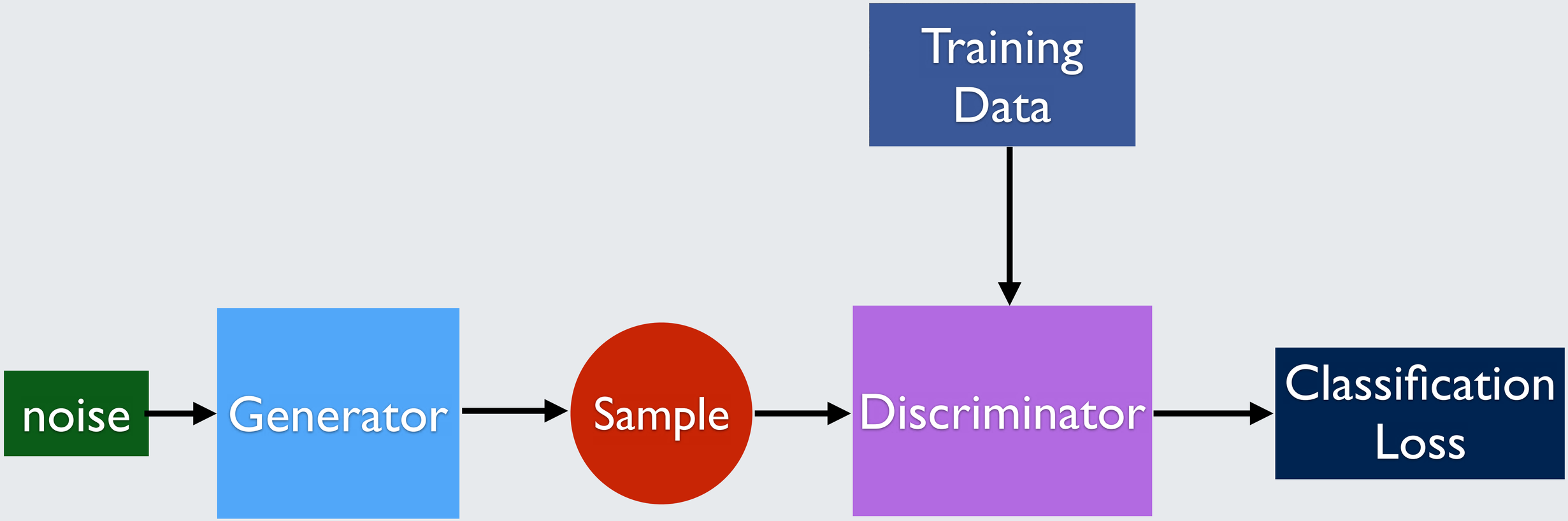
Michael Mathieu, Camille Couprie, Yann LeCun

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

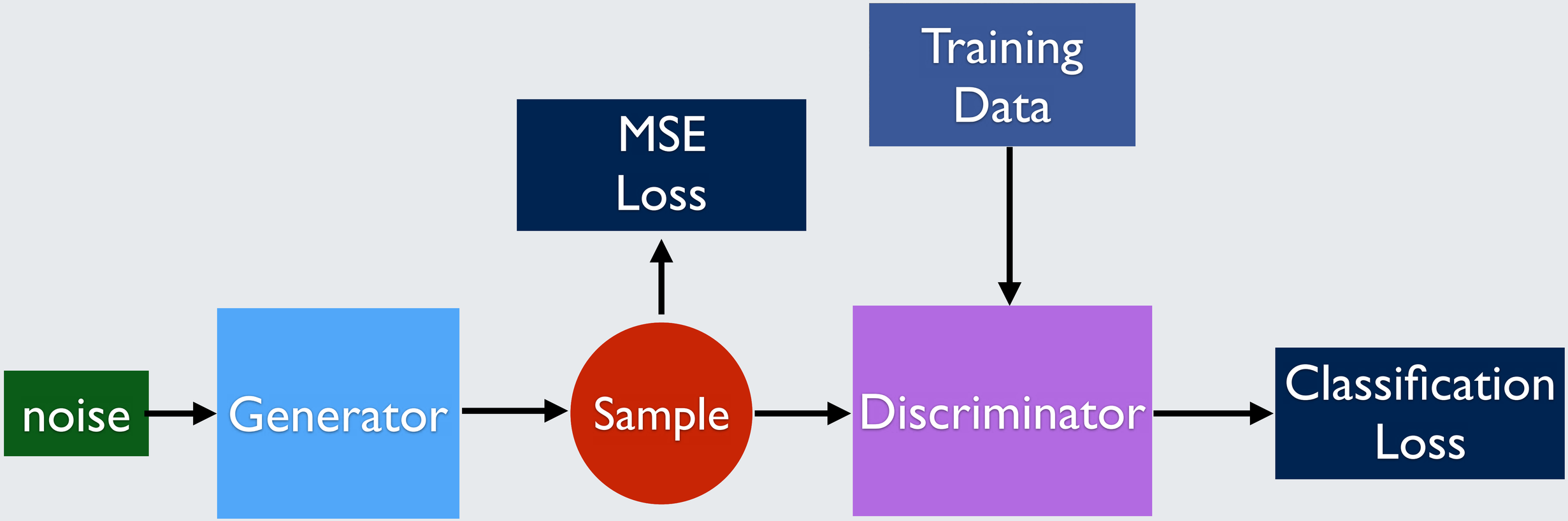
Learning to predict future images from a video sequence involves the construction of an internal representation that models the image evolution accurately, and therefore, to some degree, its content and dynamics. This is why pixel-space video prediction may be viewed as a promising avenue for unsupervised 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



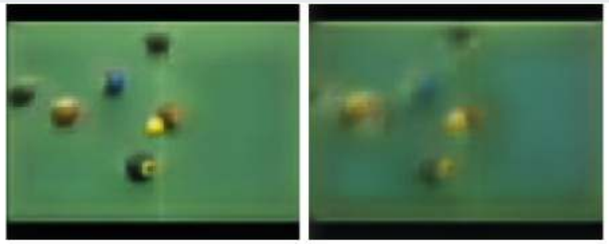
Video Prediction GANs



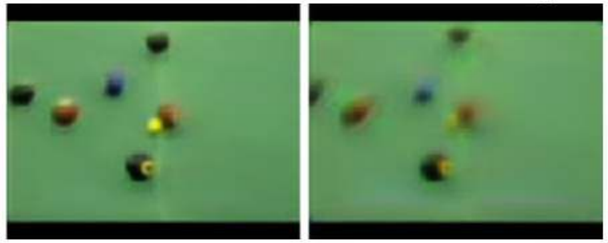
Video Prediction GANs

Input frames

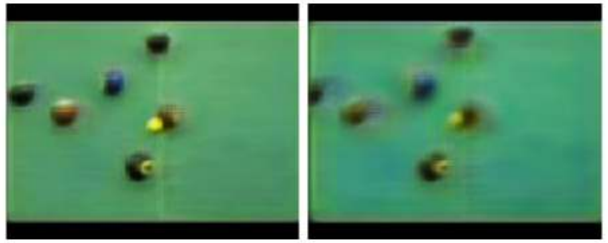
Ground truth



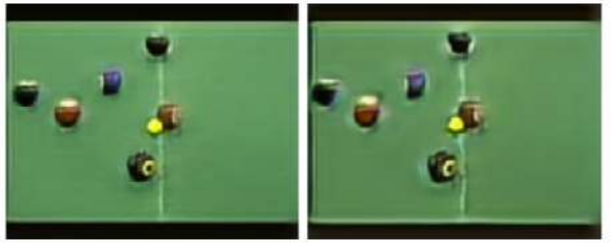
ℓ_2 result



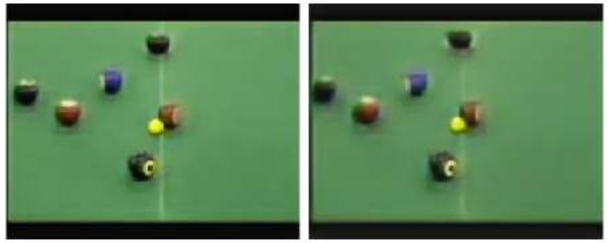
ℓ_1 result



GDL ℓ_1 result



Adversarial result



Adversarial+GDL result

DCGANs

arxiv.org/abs/1511.06434

Cornell University Library

arXiv.org > cs > arXiv:1511.06434

Search

Computer Science > Learning

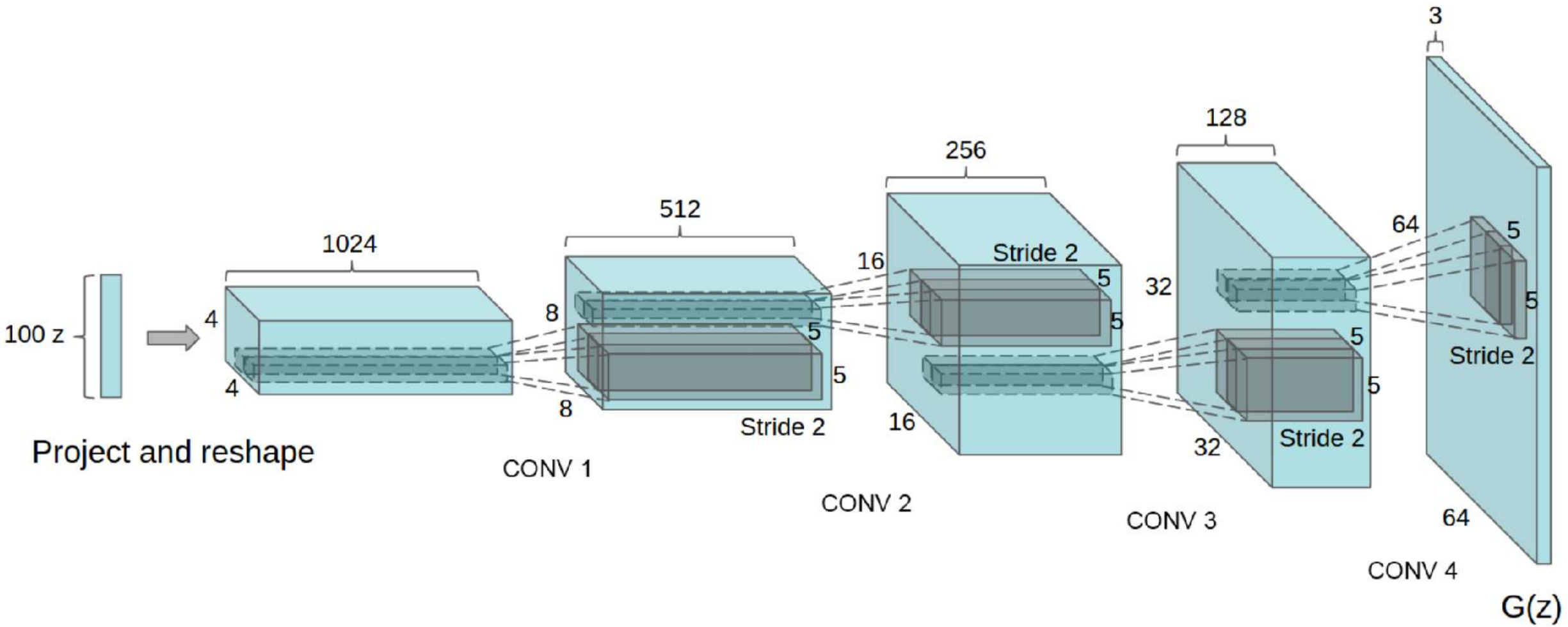
Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks

Alec Radford, Luke Metz, Soumith Chintala

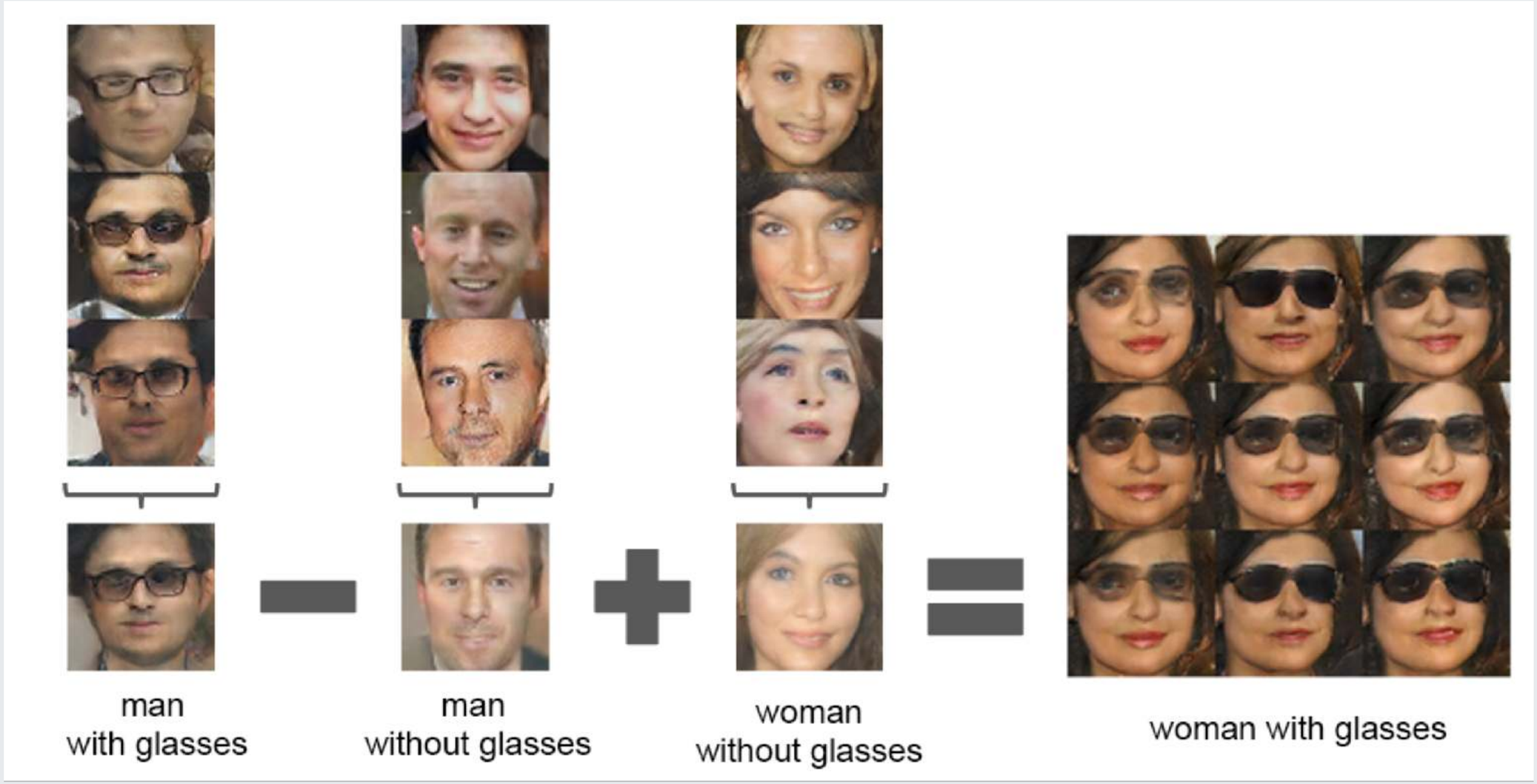
(Submitted on 19 Nov 2015 (v1), last revised 7 Jan 2016 (this version, v2))

In recent years, supervised learning with convolutional networks (CNNs) has seen huge adoption in computer vision applications. Comparatively, unsupervised learning with CNNs has received less attention. In this work we hope to help bridge the gap between the success of CNNs for supervised learning and unsupervised learning. We introduce a class of CNNs called deep convolutional generative adversarial networks (DCGANs), that have certain architectural constraints, and demonstrate that they are a strong candidate for unsupervised learning. Training on various image datasets, we show convincing evidence that our deep convolutional adversarial pair learns a hierarchy of representations from object parts to scenes in both the generator and discriminator. Additionally, we use the learned features for novel tasks - demonstrating their applicability as general image representations.

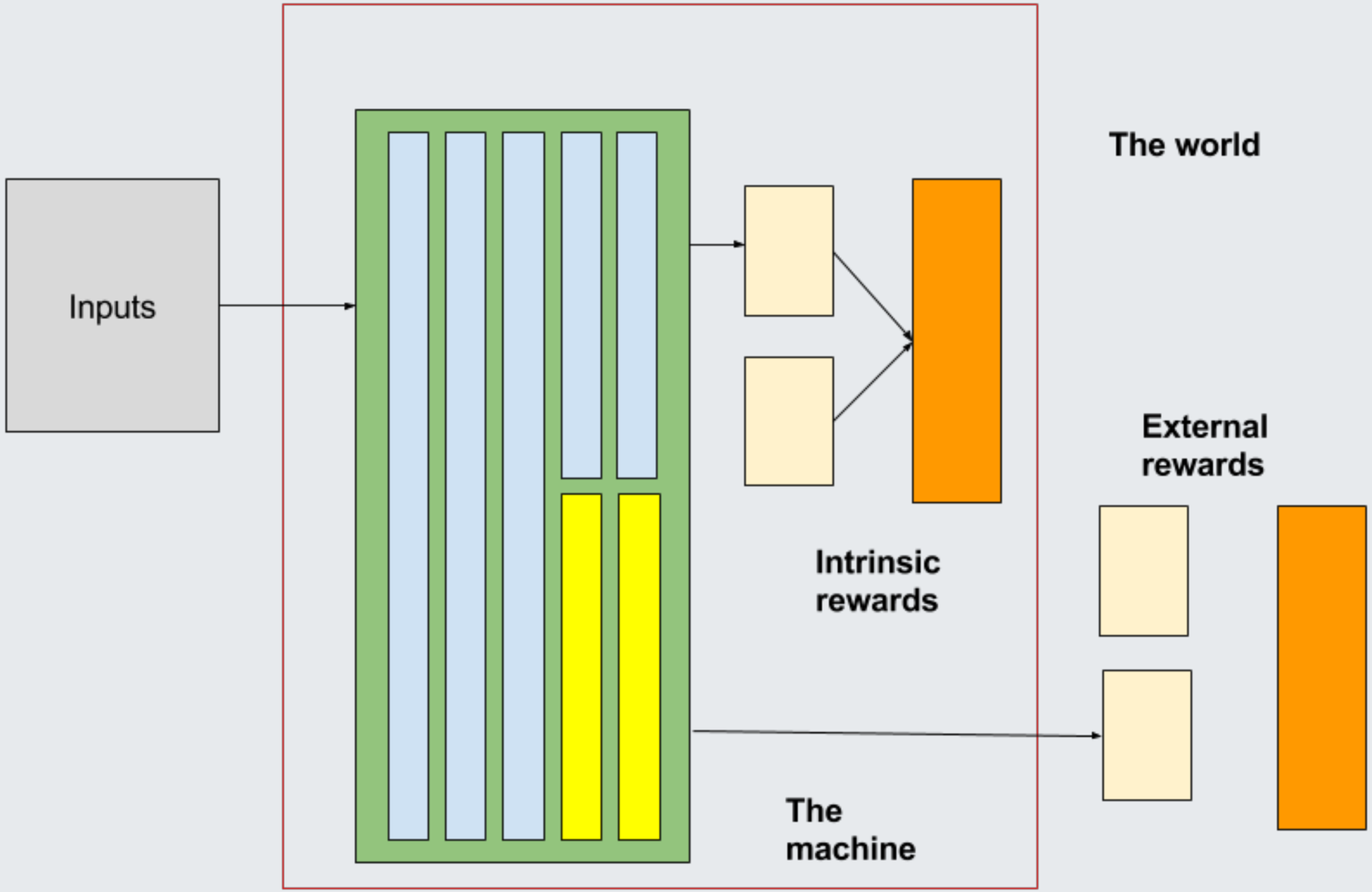
Comments: Under review as a conference paper at ICLR 2016
Subjects: **Learning (cs.LG)**; Computer Vision and Pattern Recognition (cs.CV)
Cite as: arXiv:1511.06434 [cs.LG]
(or arXiv:1511.06434v2 [cs.LG] for this version)


100 z
Project and reshape
4
4
1024
8
8
512
Stride 2
CONV 1
16
16
256
Stride 2
CONV 2
32
32
128
Stride 2
CONV 3
64
64
3
Stride 2
CONV 4
G(z)
5

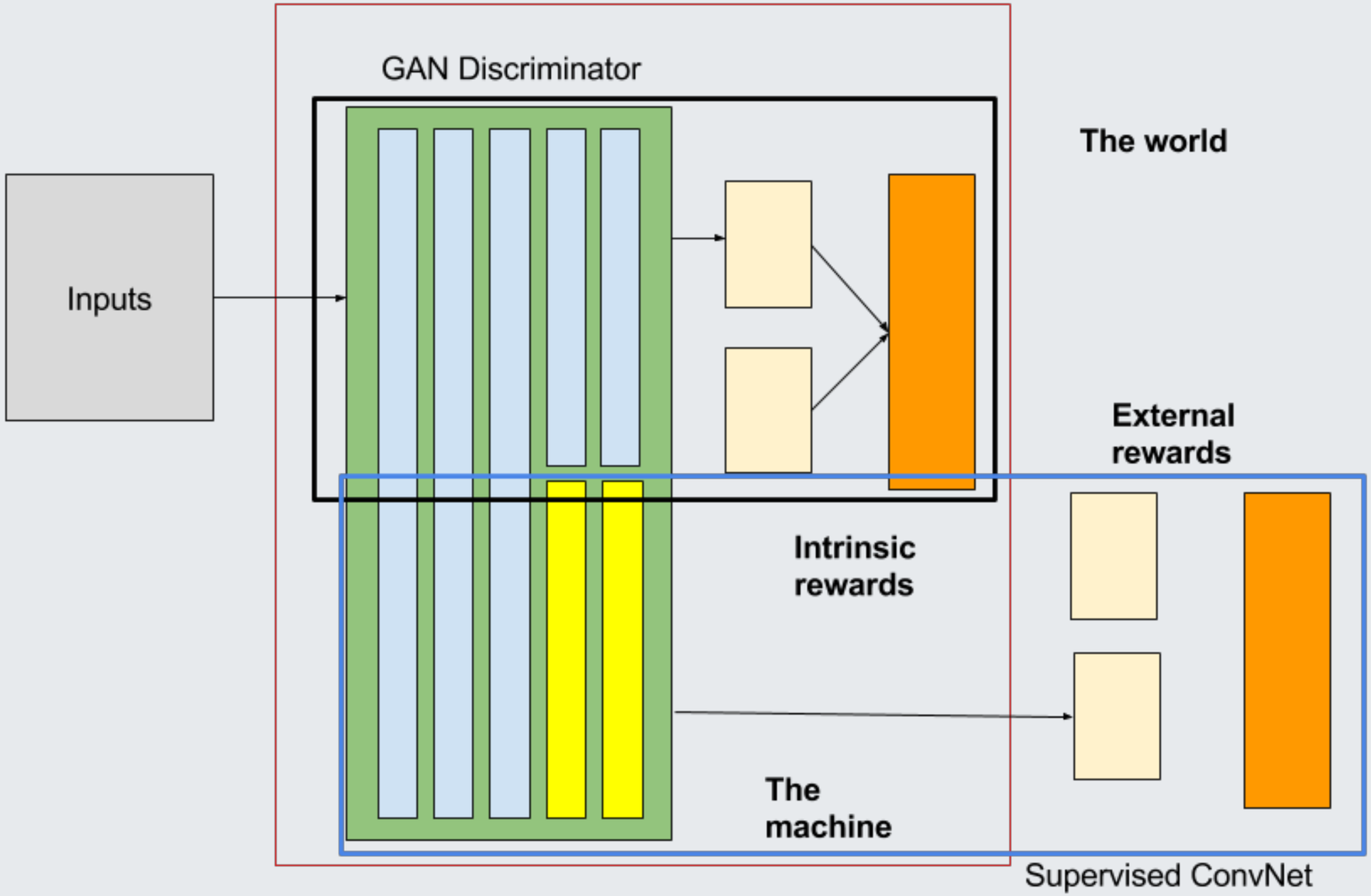
Latent space arithmetic



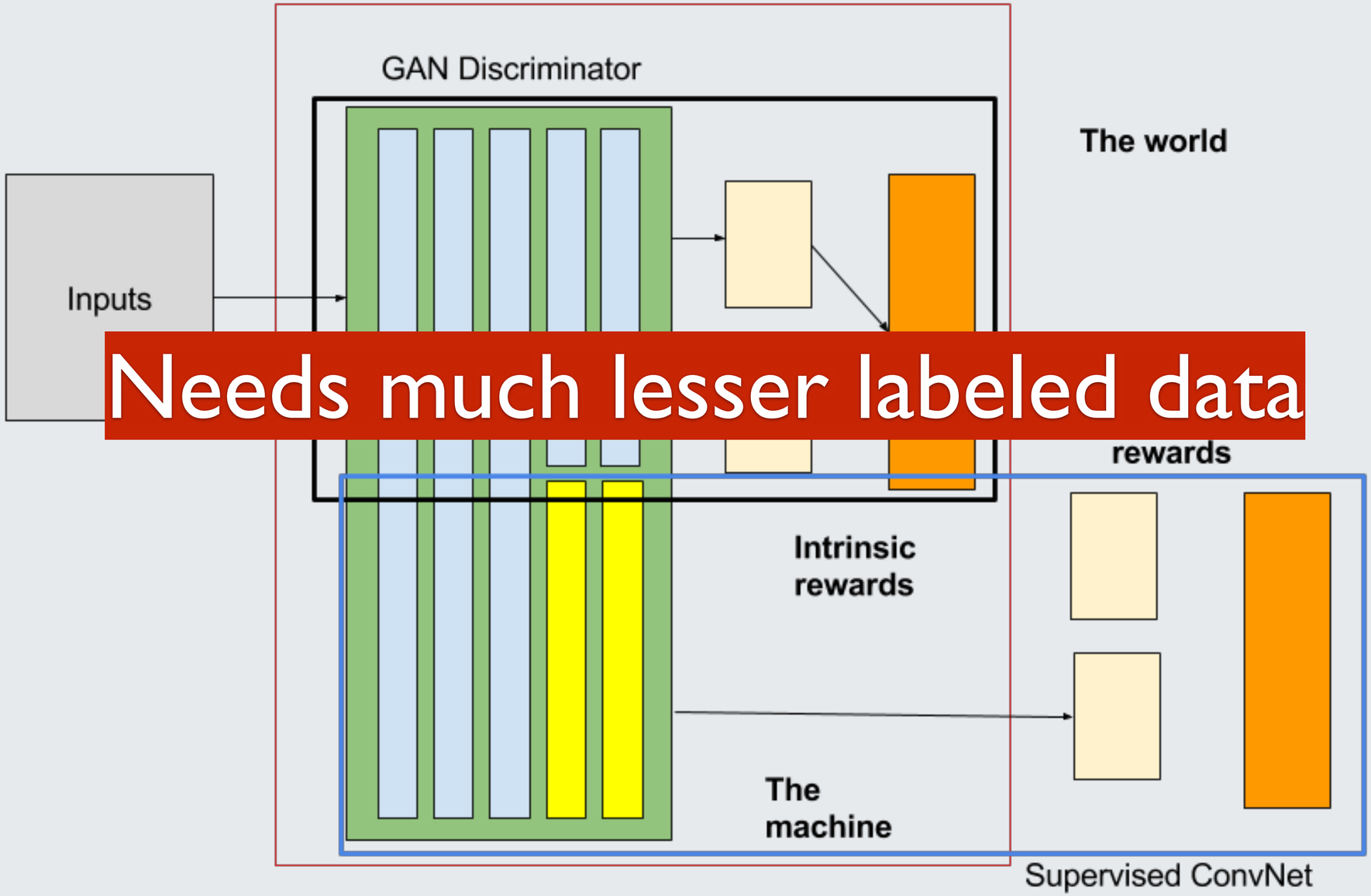
Using the GAN feature representation



Using the GAN feature representation



Using the GAN feature representation



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)

In-painting GANs

Context Encoders: Feature Learning by Inpainting

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

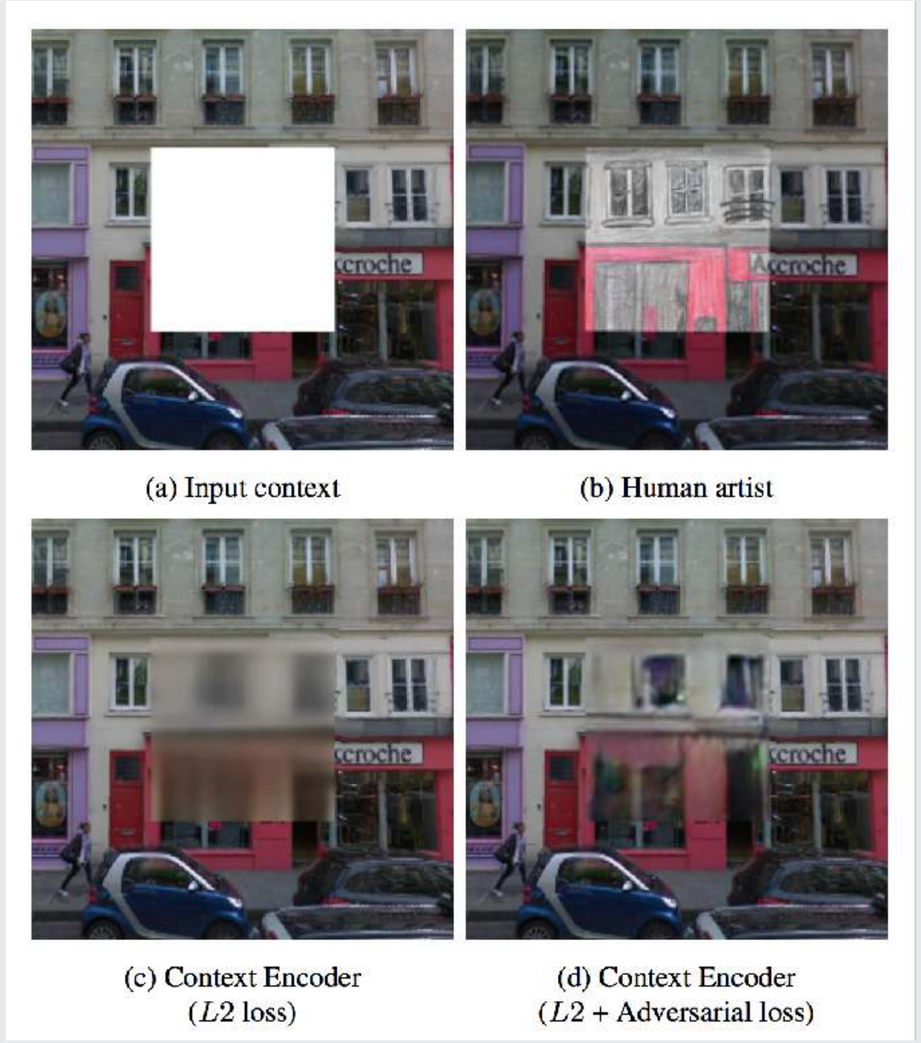
(Submitted on 25 Apr 2016)

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

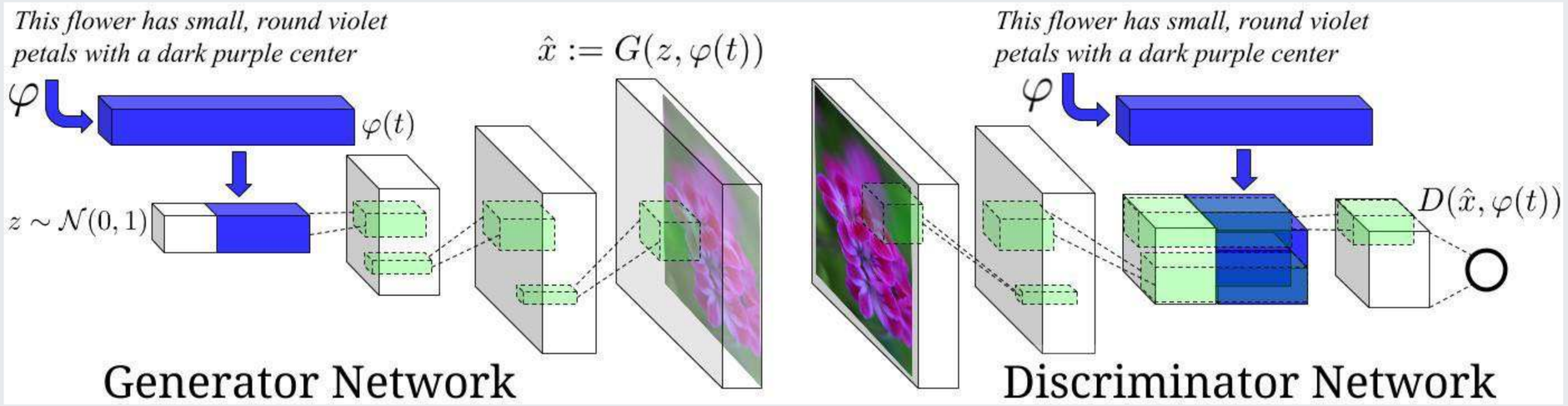


Figure from Reed et. al. 2016

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	

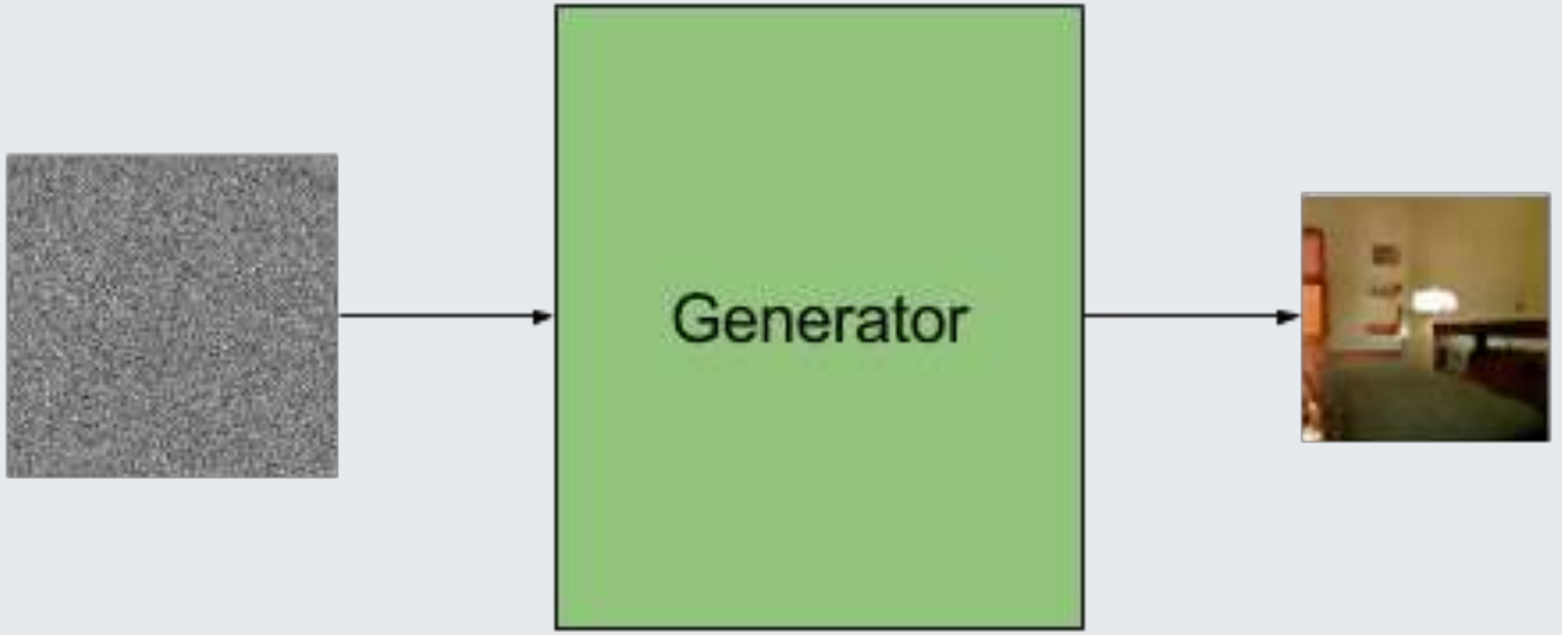
Text-conditional GANs

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	

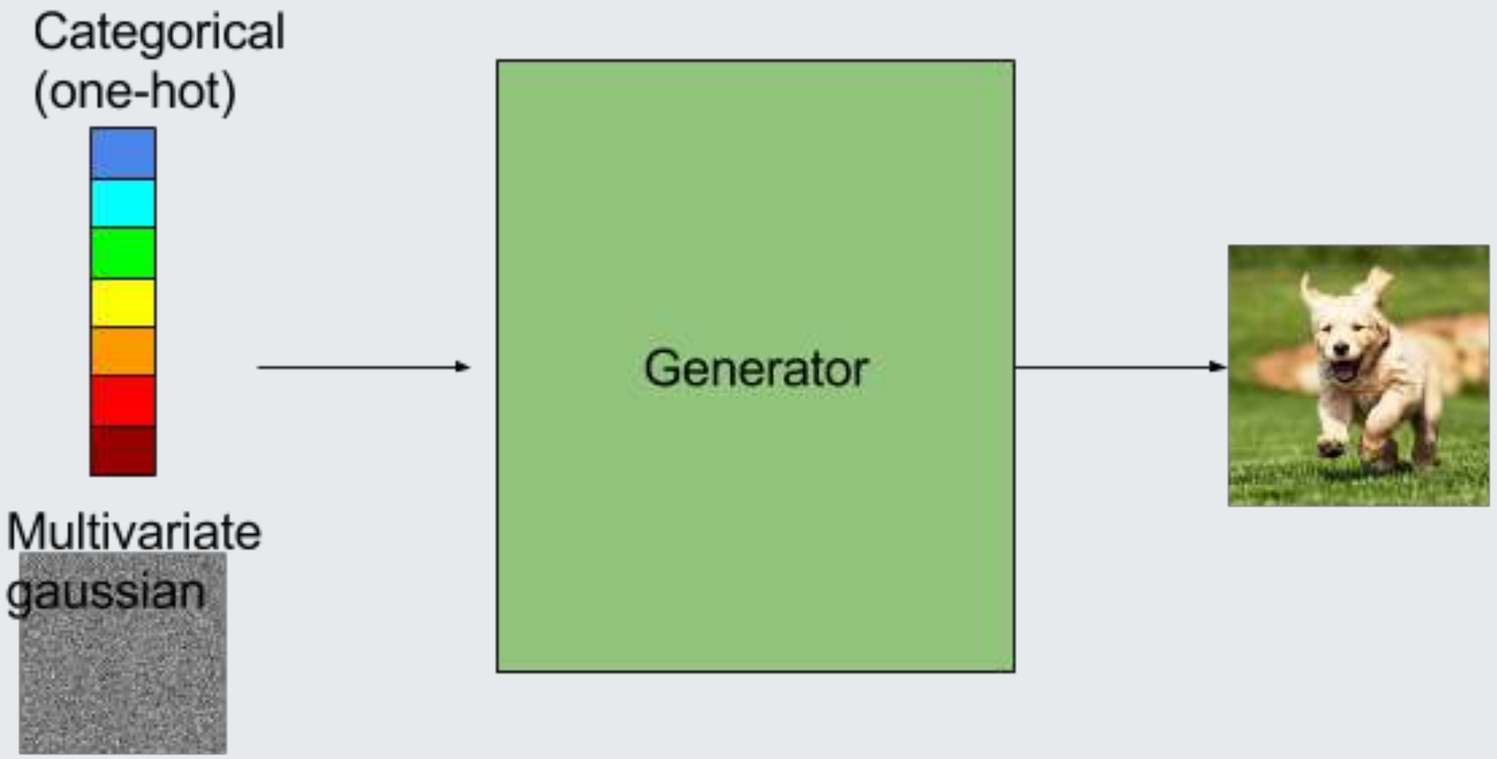
Text-conditional GANs

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	

Disentangling representations



Disentangling representations



Stability and Representation Reuse

Improved Techniques for Training GANs

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen

(Submitted on 10 Jun 2016)

We present a variety of new architectural features and training procedures that we apply to the generative adversarial networks (GANs) framework. We focus on two applications of GANs: semi-supervised learning, and the generation of images that humans find visually realistic. Unlike most work on generative models, our primary goal is not to train a model that assigns high likelihood to test data, nor do we require the model to be able to learn well without using any labels. Using our new techniques, we achieve state-of-the-art results in semi-supervised classification on MNIST, CIFAR-10 and SVHN. The generated images are of high quality as confirmed by a visual Turing test: our model generates MNIST samples that humans cannot distinguish from real data, and CIFAR-10 samples that yield a human error rate of 21.3%. We also present ImageNet samples with unprecedented resolution and show that our methods enable the model to learn recognizable features of ImageNet classes.

Subjects: **Learning (cs.LG)**; Computer Vision and Pattern Recognition (cs.CV); Neural and Evolutionary Computing (cs.NE)
Cite as: **arXiv:1606.03498 [cs.LG]**
(or **arXiv:1606.03498v1 [cs.LG]** for this version)

Stability and Representation Reuse

- Feature matching
- Minibatch discrimination
- Label smoothing
- What’s next?

Stability and Representation Reuse

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	

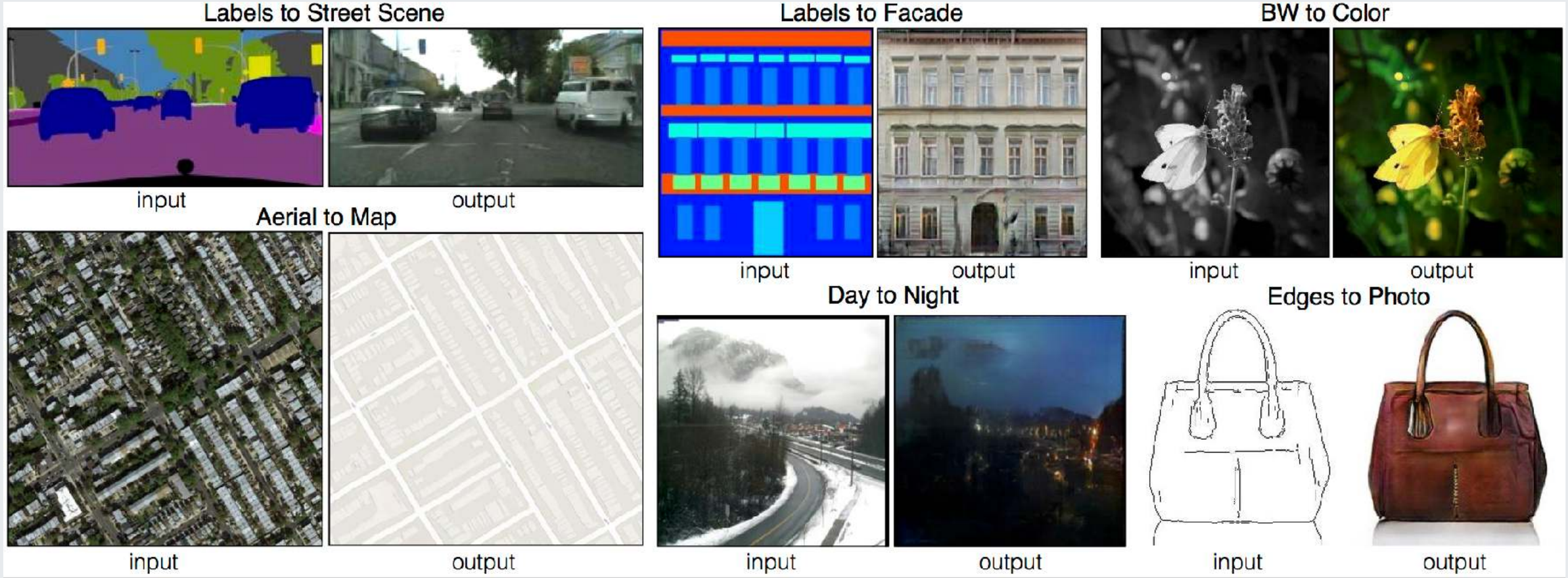
Figure 5: *(Left)* Error rate on SVHN. *(Right)* Samples from the generator for SVHN.

Stability and Representation Reuse

Model	Test error rate for a given number of labeled samples			
	1000	2000	4000	8000
Ladder network [24]			20.40 ± 0.47	
CatGAN [14]			19.58 ± 0.46	
Our model	21.83 ± 2.01	19.61 ± 2.09	18.63 ± 2.32	17.72 ± 1.82
Ensemble of 10 of our models	19.22 ± 0.54	17.25 ± 0.66	15.59 ± 0.47	14.87 ± 0.89

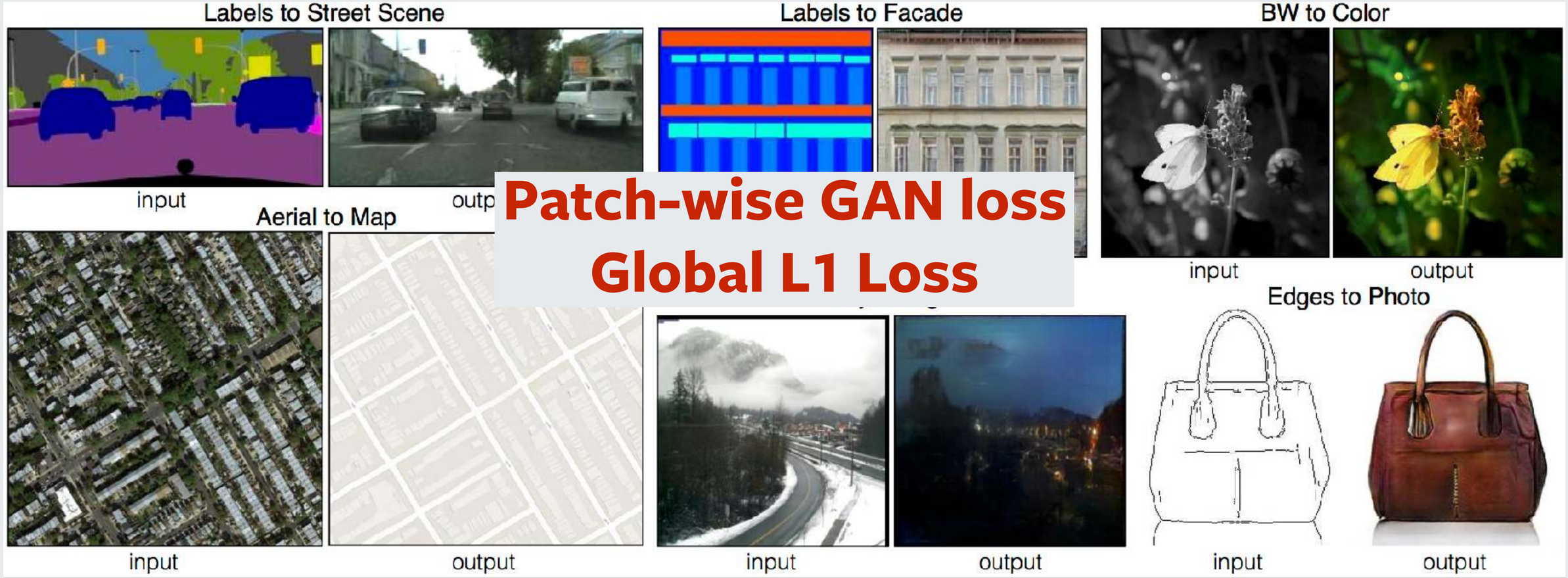
Table 2: Test error on semi-supervised CIFAR-10. Results are averaged over 10 splits of data.

Pix2Pix: Image-to-Image Translation with Conditional Adversarial Nets



https://phillipi.github.io/pix2pix/

Pix2Pix: Image-to-Image Translation with Conditional Adversarial Nets



https://phillipi.github.io/pix2pix/

GANs are Unstable

Comparison to Classification ConvNets

- Throw things at the wall and see what sticks
- Intuition is poorer
- No objective validation

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

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

- The gra

$D^*_G(\boldsymbol{x})$ ... $\boldsymbol{x})$, (3)

and its de

(4)

can be un ... lition to
the deriva

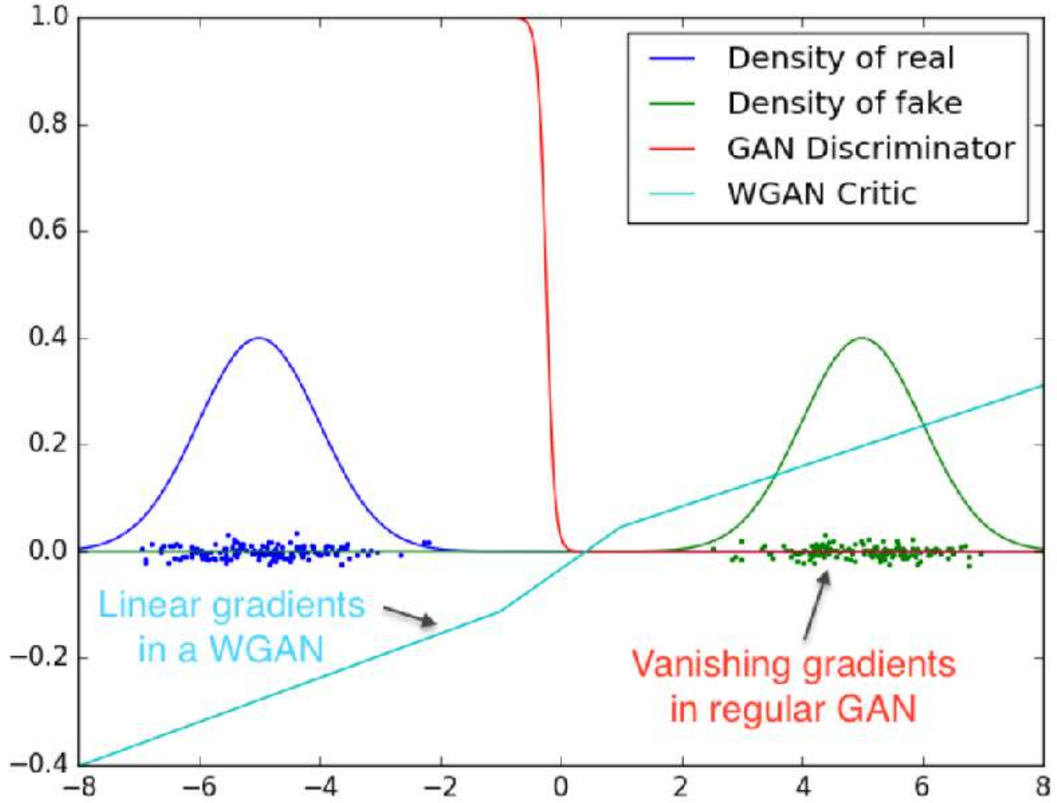


Figure 2: Optimal discriminator and critic when learning to differentiate two Gaussians. As we can see, the traditional GAN discriminator saturates and results in vanishing gradients. Our WGAN critic provides very clean gradients on all parts of the space.

Miyato e ... s" (2018)

Central problems with the GAN objective

- Based on f-divergence
- Needs overlap between real and generated distributions

The *Kullback-Leibler* (KL) divergence

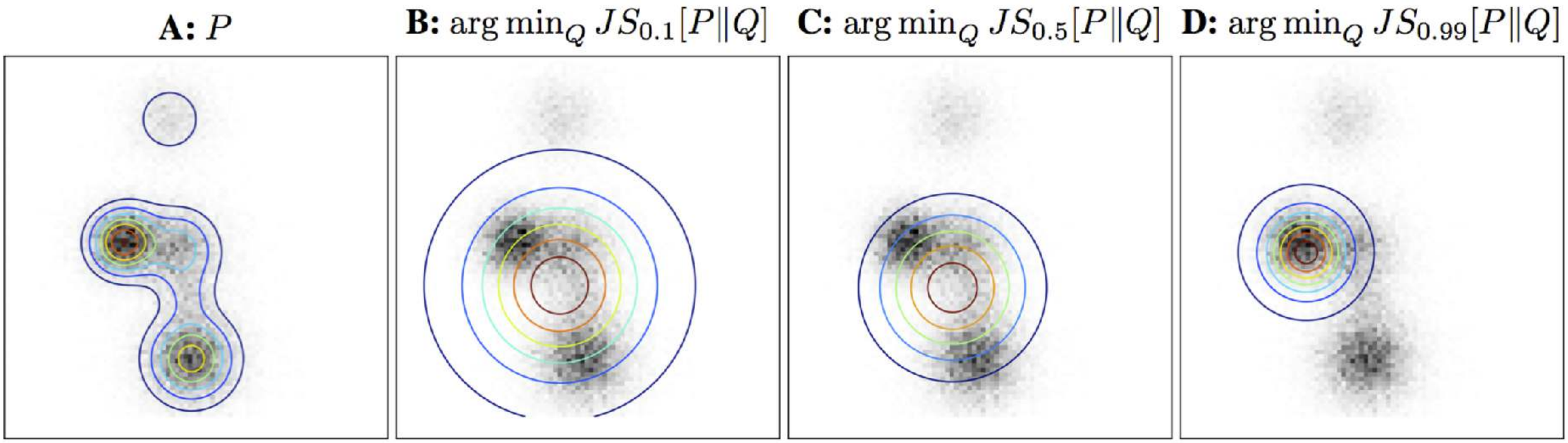
$$KL(\mathbb{P}_r \| \mathbb{P}_g) = \int \log \left(\frac{P_r(x)}{P_g(x)} \right) P_r(x) d\mu(x) \ ,$$

The *Jensen-Shannon* (JS) divergence

$$JS(\mathbb{P}_r, \mathbb{P}_g) = KL(\mathbb{P}_r \| \mathbb{P}_m) + KL(\mathbb{P}_g \| \mathbb{P}_m) \ ,$$

Central problems with the GAN objective

- Based on f-divergence
- Needs overlap between real and generated distributions



http://www.inference.vc/how-to-train-your-generative-models-why-generative-adversarial-networks-work-so-well-2/

Regularizing GANs

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

Using EarthMover's distance

- An alternative to f-divergence
- think of the probability distributions as mounds of dirt
  - the EM distance describes how much effort it takes to transform one mound of dirt so it is the same as the other using an optimal transport plan

The *Earth-Mover* (EM) distance or Wasserstein-1

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma} \left[\|x - y\| \right] ,$$

where $\Pi(\mathbb{P}_r, \mathbb{P}_g)$ denotes the set of all joint distributions $\gamma(x, y)$ whose marginals are respectively $\mathbb{P}_r$ and $\mathbb{P}_g$.

Paper review: https://paper.dropbox.com/doc/Wasserstein-GAN-GvU0p2V9ThzdwY3BbhoP7

Using EarthMover's distance

- Considers all possible "configurations" of pairing up points from the two distributions
- Calculates the mean distance of pairs in each configuration
- Returns the smallest mean distance across all of the configurations
- Intractable, can't compute directly

The *Earth-Mover* (EM) distance or Wasserstein-1

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}\big[\, \|x - y\| \,\big] \,,$$

Using EarthMover's distance

- Tractable alternate definition
- Using Kantorovich-Rubinstein duality

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x \sim \mathbb{P}_r}[f(x)] - \mathbb{E}_{x \sim \mathbb{P}_\theta}[f(x)]$$

where the supremum is over all the 1-Lipschitz functions $f : \mathcal{X} \to \mathbb{R}$.

- Notation: think of the supremum as a maximum
- Intuitively we are finding some function with greatest margin between the mean value over real examples and the mean value over generated examples

WGANs

- Simple algorithm
  - Use Wasserstein-1 distance as objective
  - Clip weights of neural network to maintain 1-lipschitz

Algorithm 1 WGAN, our proposed algorithm. All experiments in the paper used the default values $\alpha = 0.00005$, $c = 0.01$, $m = 64$, $n_{\text{critic}} = 5$.

Require: : α, the learning rate. c, the clipping parameter. m, the batch size. n_{critic}, the number of iterations of the critic per generator iteration.

Require: : w_0, initial critic parameters. θ_0, initial generator's parameters.

while θ has not converged **do**
  for $t = 0, ..., n_{\text{critic}}$ **do**
    Sample $\{x^{(i)}\}_{i=1}^{m} \sim \mathbb{P}_r$ a batch from the real data.
    Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a batch of prior samples.
    $g_w \leftarrow \nabla_w \left[\frac{1}{m}\sum_{i=1}^{m} f_w(x^{(i)}) - \frac{1}{m}\sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))\right]$
    $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$
    $w \leftarrow \text{clip}(w, -c, c)$
  end for
  Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a batch of prior samples.
  $g_\theta \leftarrow -\nabla_\theta \frac{1}{m}\sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))$
  $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$
end while

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:

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

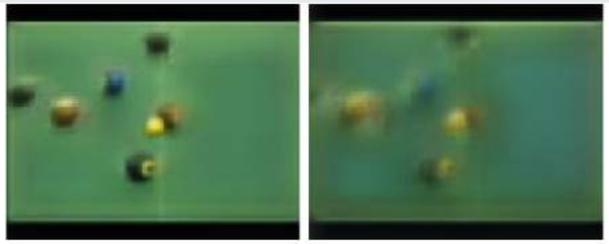
  3. Update W^l with SGD on mini-batch dataset $\mathcal{D}_M$ with a learning rate α:

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

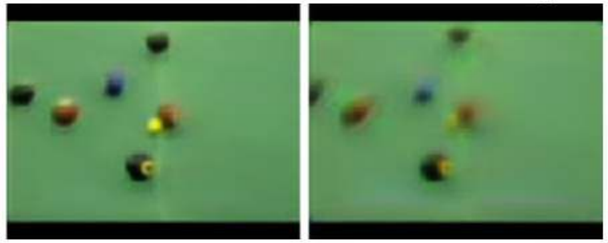
GANs as forward models

Input frames

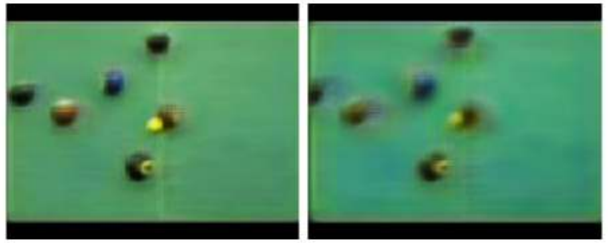
Ground truth



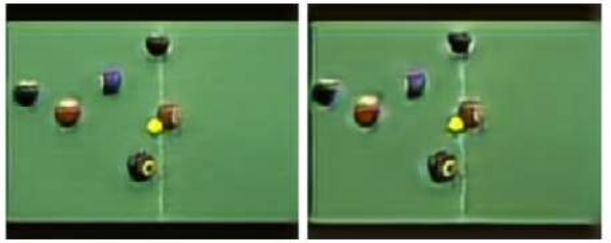
ℓ_2 result



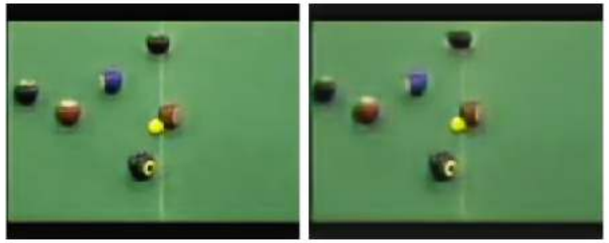
ℓ_1 result



GDL ℓ_1 result



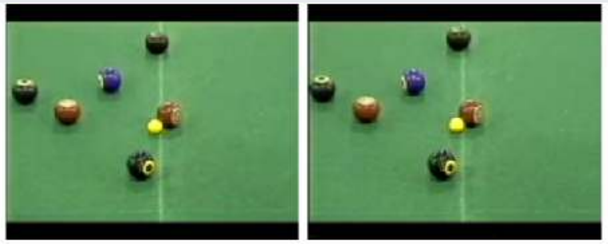
Adversarial result



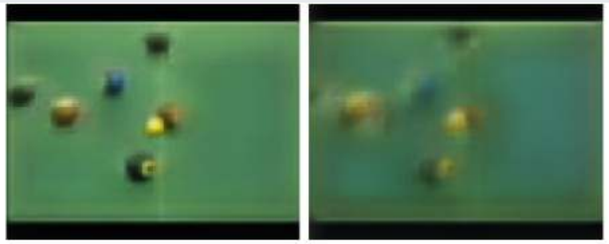
Adversarial+GDL result

GANs as forward models

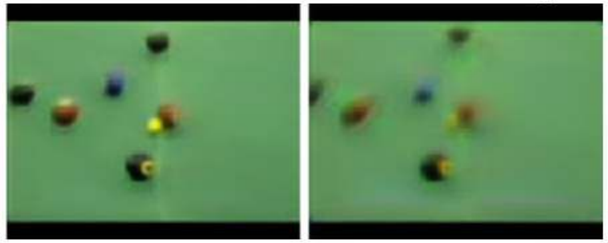
Input frames



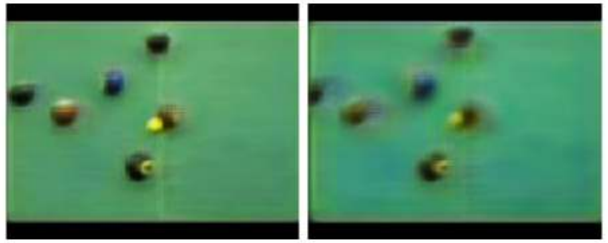
Ground truth



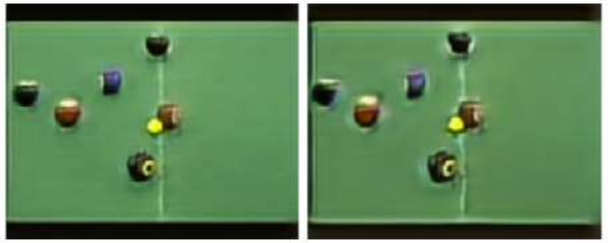
ℓ_2 result



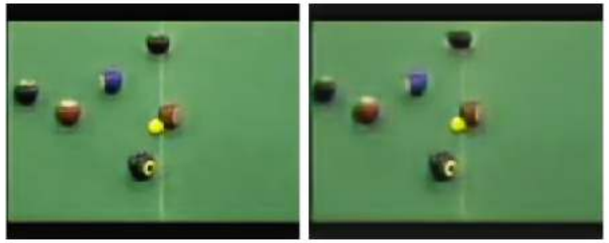
ℓ_1 result



GDL ℓ_1 result



Adversarial result



Adversarial+GDL result

Do we really need to predict in pixel space?

Generative models as forward models

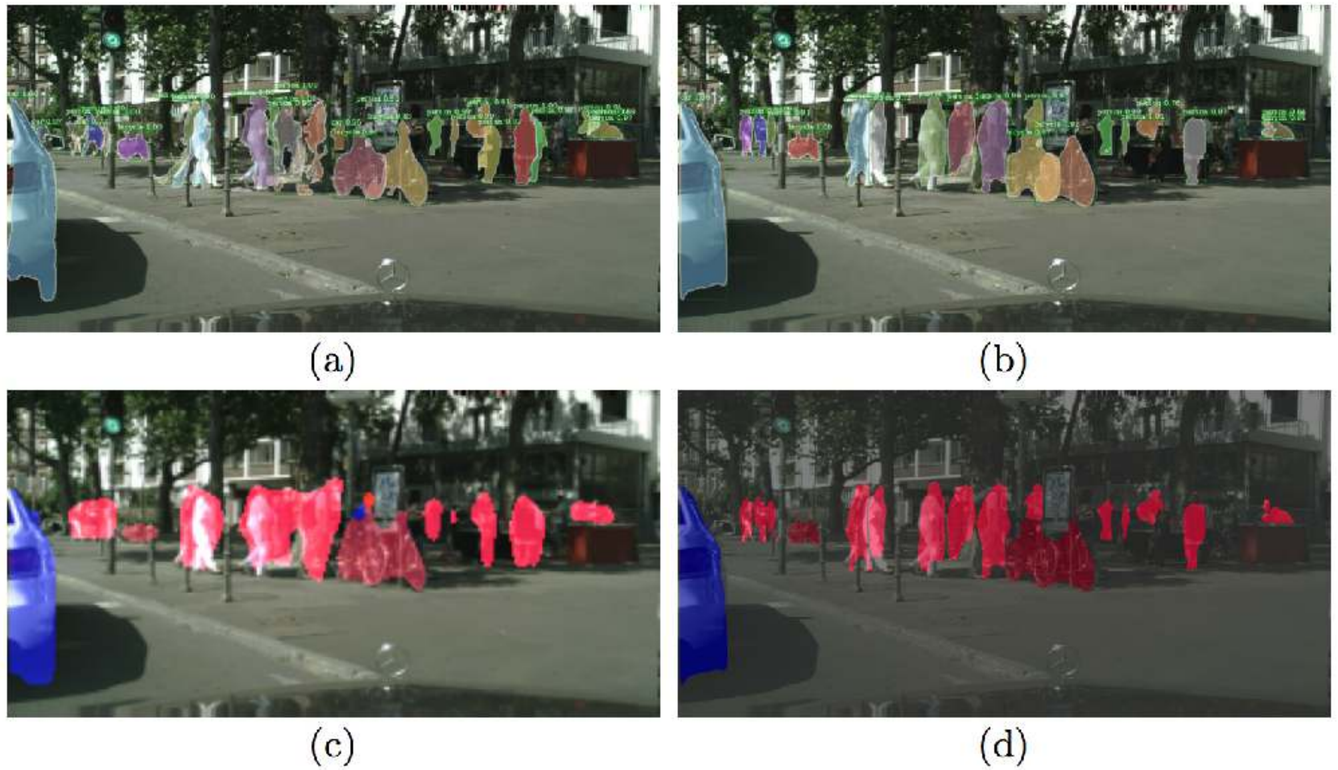


Fig. 1: Predicting 0.5 sec. into the future. Instance segmentations with (a) optical flow baseline and (b) our approach. Semantic segmentation (c) from [8] and (d) derived from our instance semantic segmentation approach. Instance modeling significantly improves the segmentation accuracy of the individual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

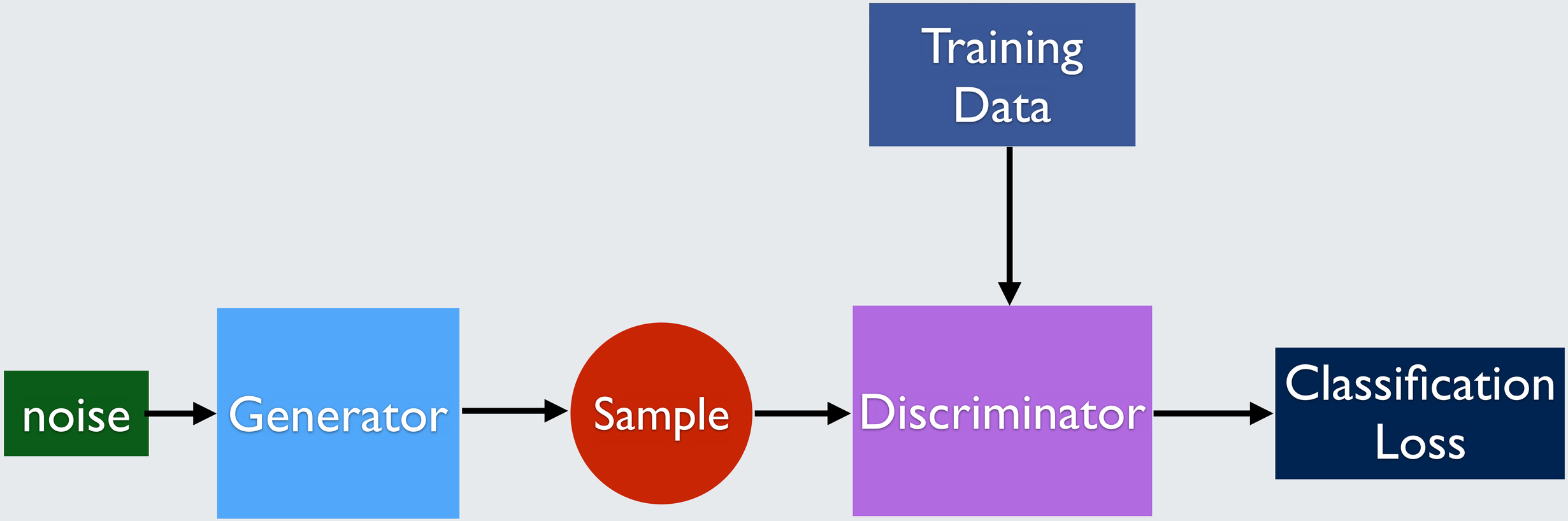
Michael Mathieu, Camille Couprie, Yann LeCun

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

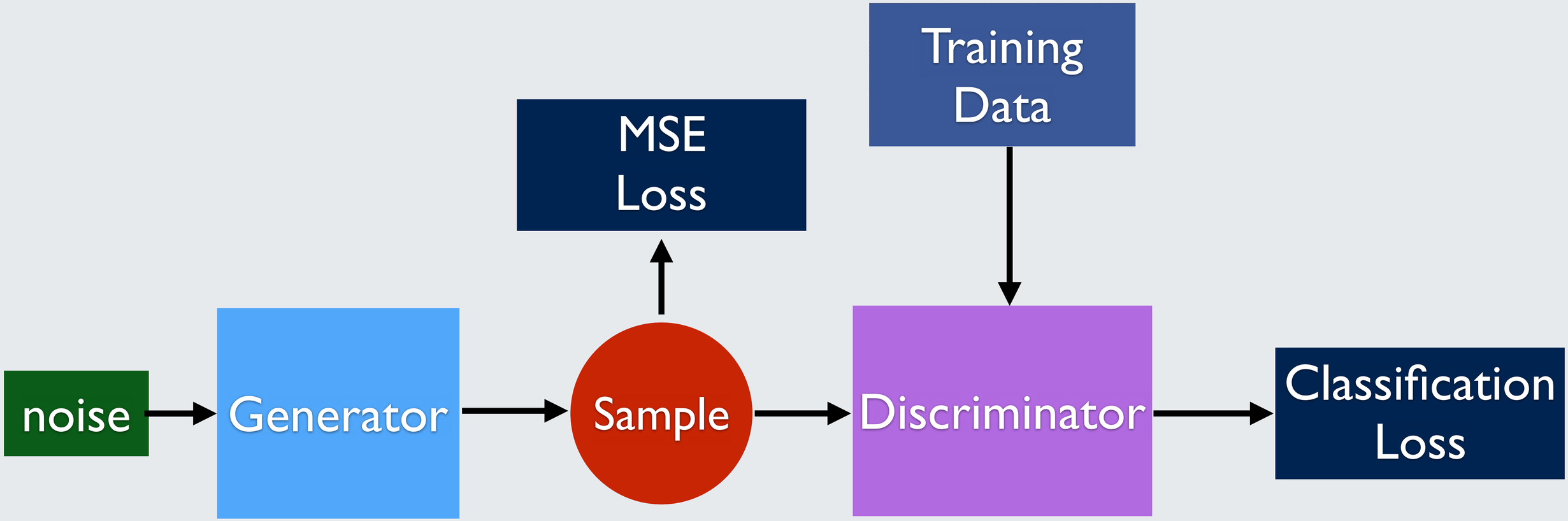
Learning to predict future images from a video sequence involves the construction of an internal representation that models the image evolution accurately, and therefore, to some degree, its content and dynamics. This is why pixel-space video prediction may be viewed as a promising avenue for unsupervised 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



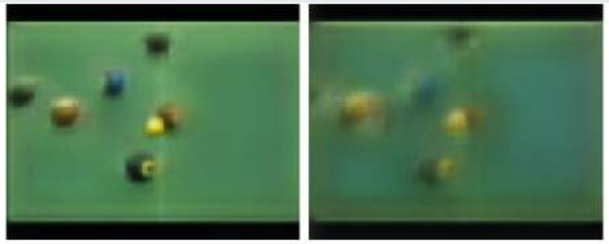
Video Prediction GANs



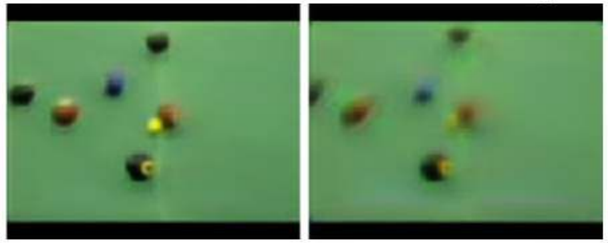
Video Prediction GANs

Input frames

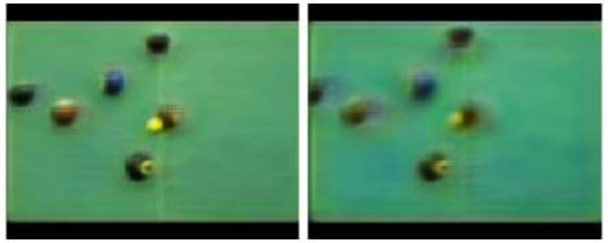
Ground truth



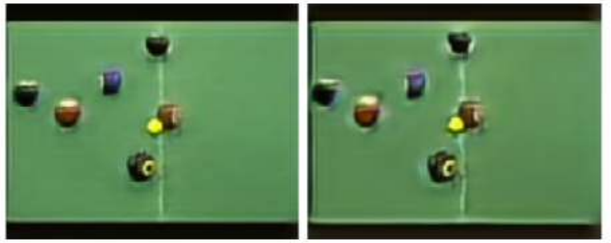
ℓ_2 result



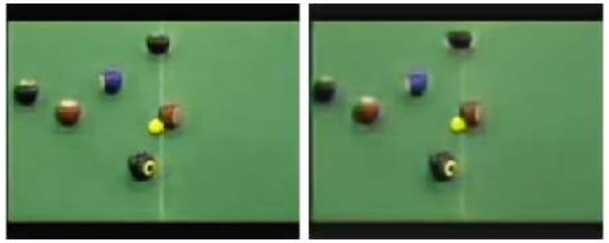
ℓ_1 result



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

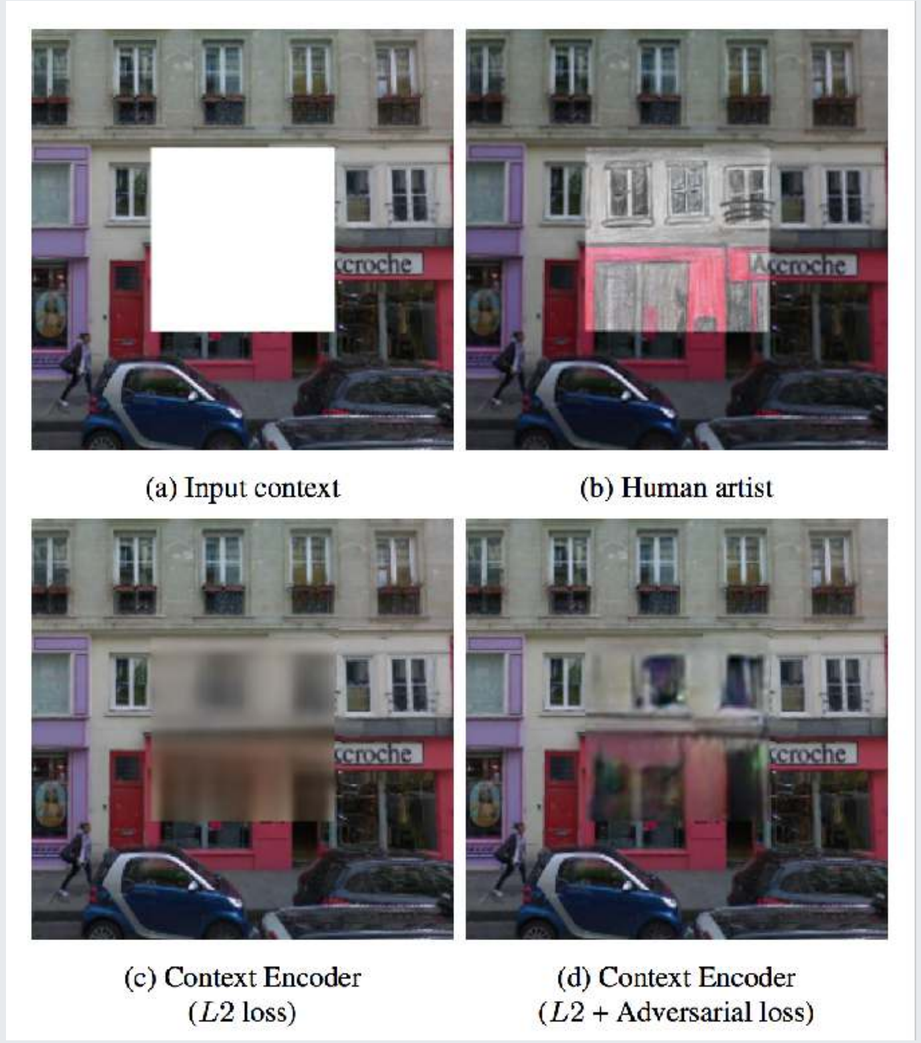
(Submitted on 25 Apr 2016)

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

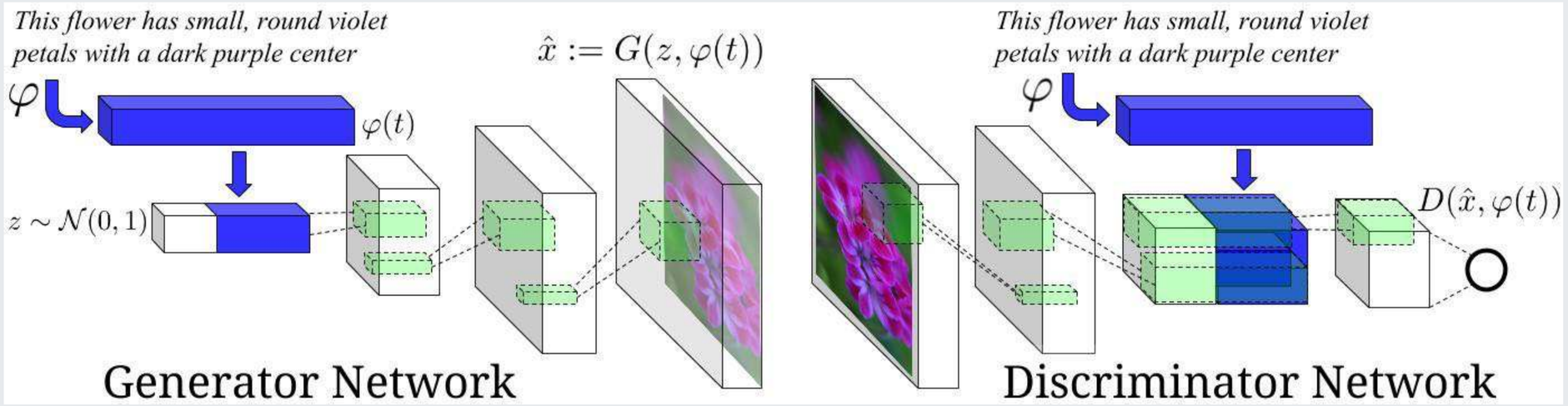


Figure from Reed et. al. 2016

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	

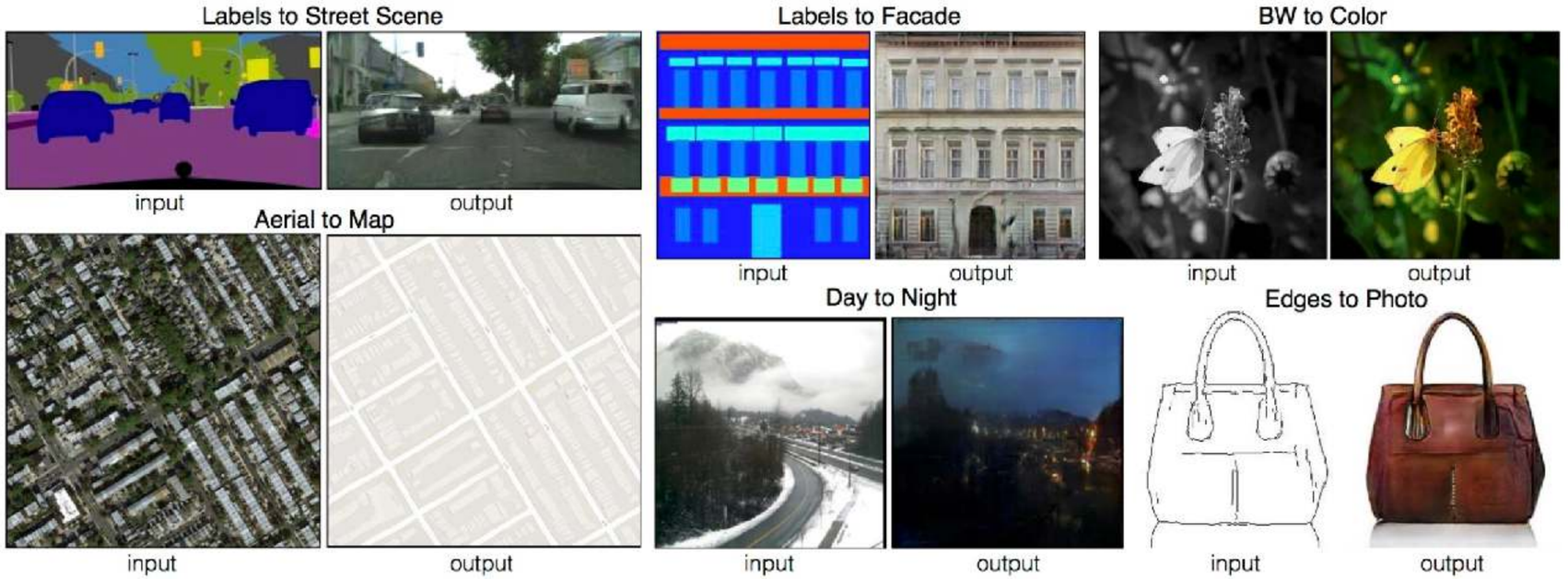
Image-to-image translation

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!!!!!!

Autoencoders

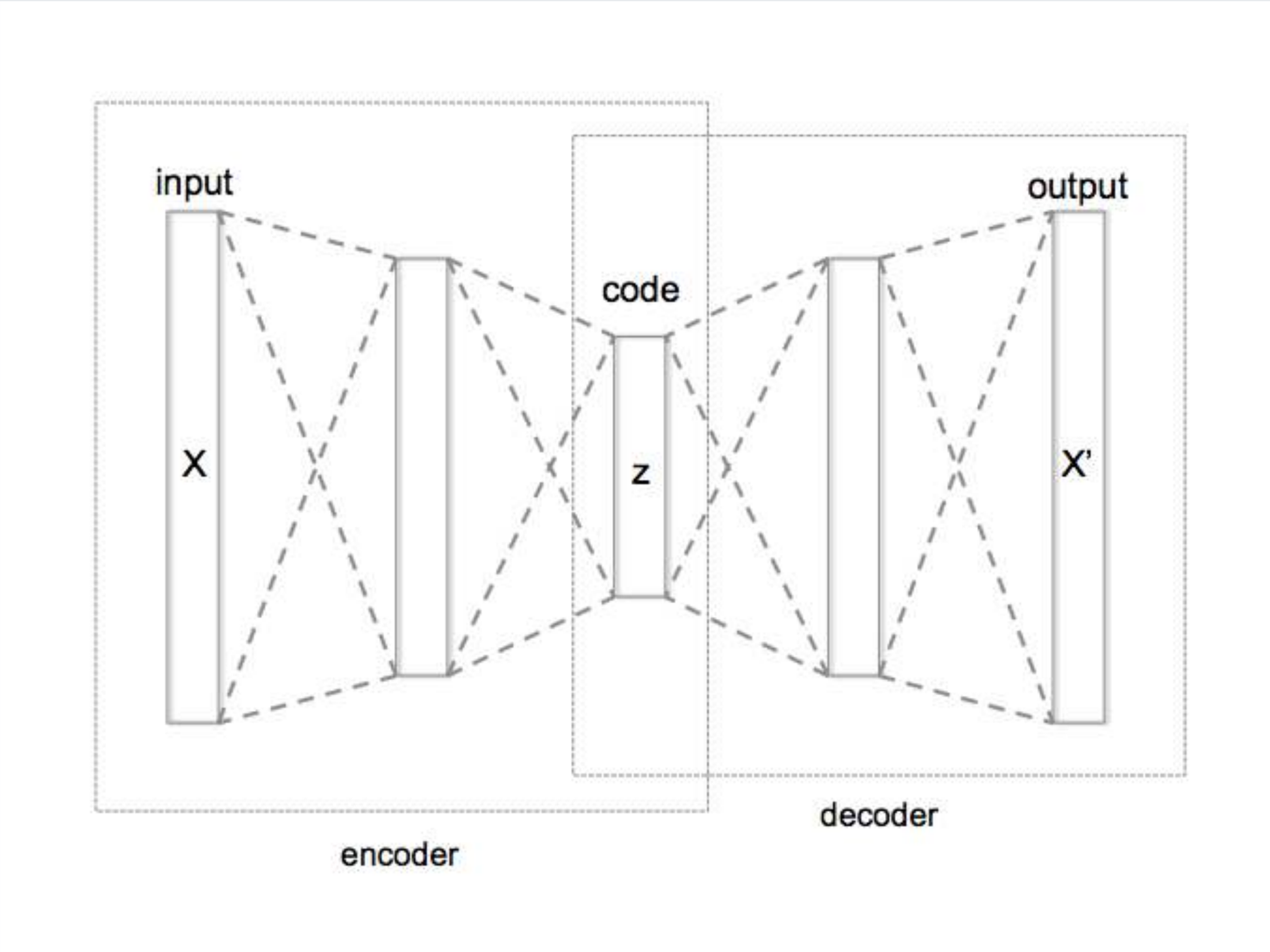


image from wikipedia

Variational Autoencoders

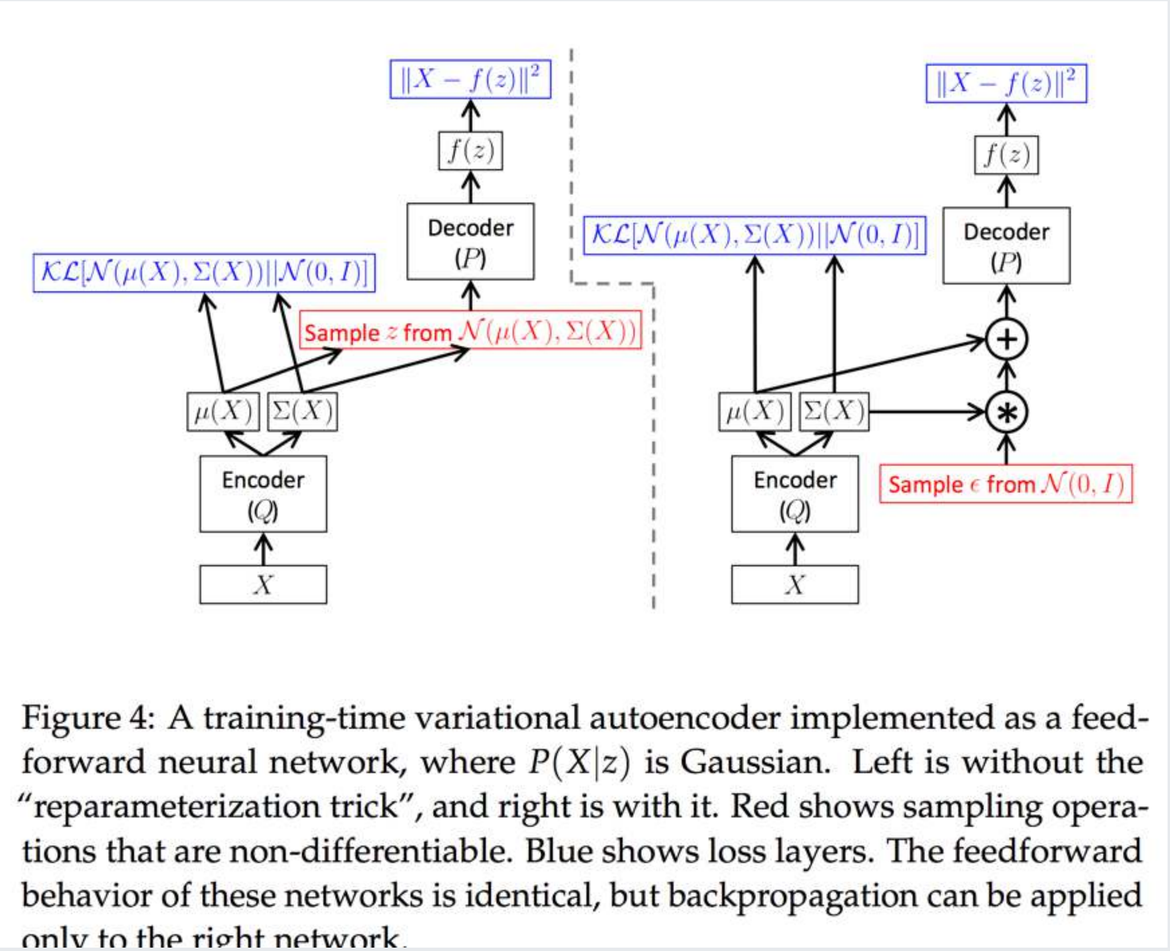
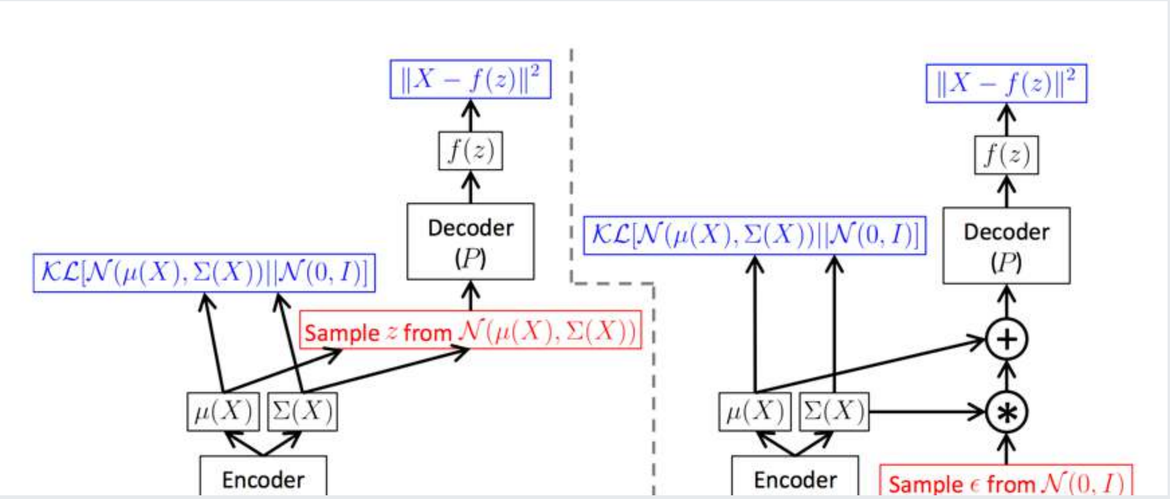


Figure 4: A training-time variational autoencoder implemented as a feedforward neural network, where $P(X|z)$ is Gaussian. Left is without the "reparameterization trick", and right is with it. Red shows sampling operations that are non-differentiable. Blue shows loss layers. The feedforward behavior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

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

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

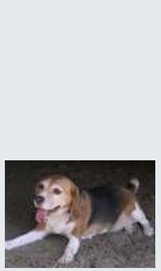
First, we consider a large set of images $\{x_1, \ldots, x_N\}$, where each image $x_i \in \mathcal{X}$ has dimensions $3 \times w \times h$. Second, we initialize a set of d-dimensional random vectors $\{z_1, \ldots, z_N\}$, where $z_i \in \mathcal{Z} \subseteq \mathbb{R}^d$ for all $i = 1, \ldots N$. 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!

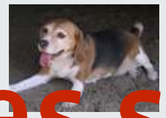





Bojanowski et. al. 2017

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