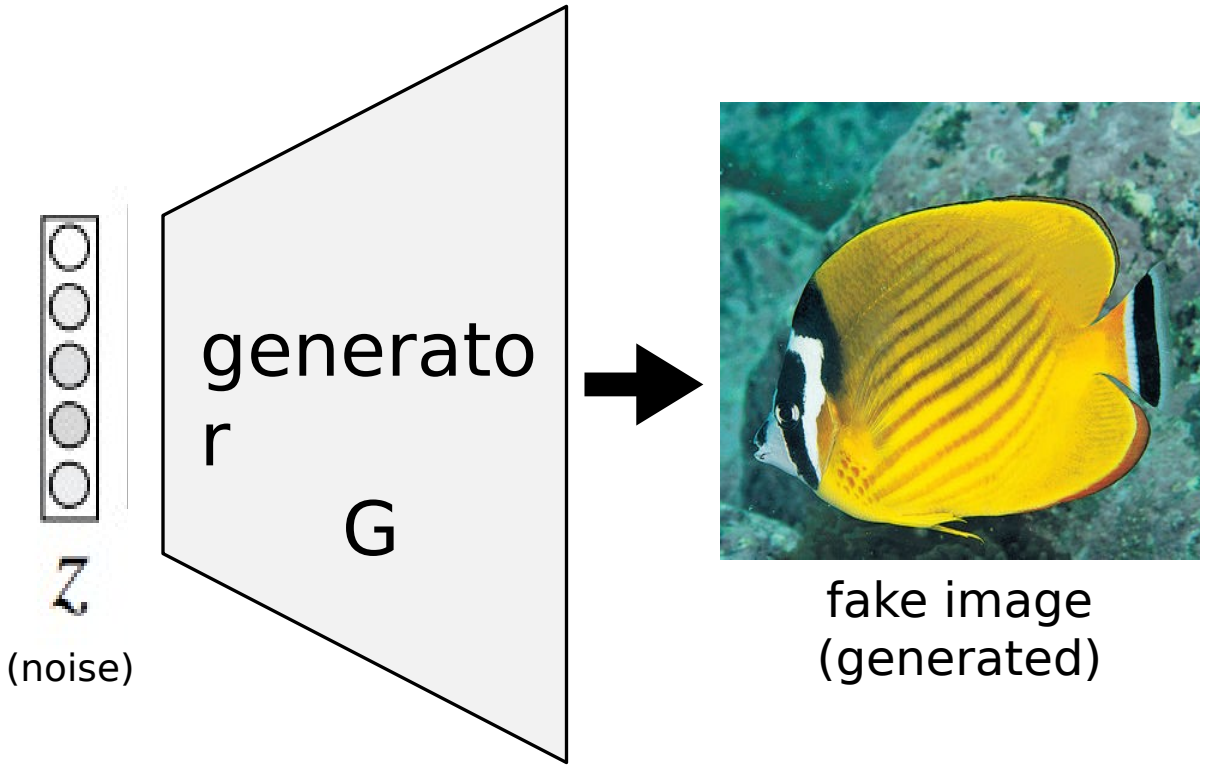
Generative Adversarial Nets

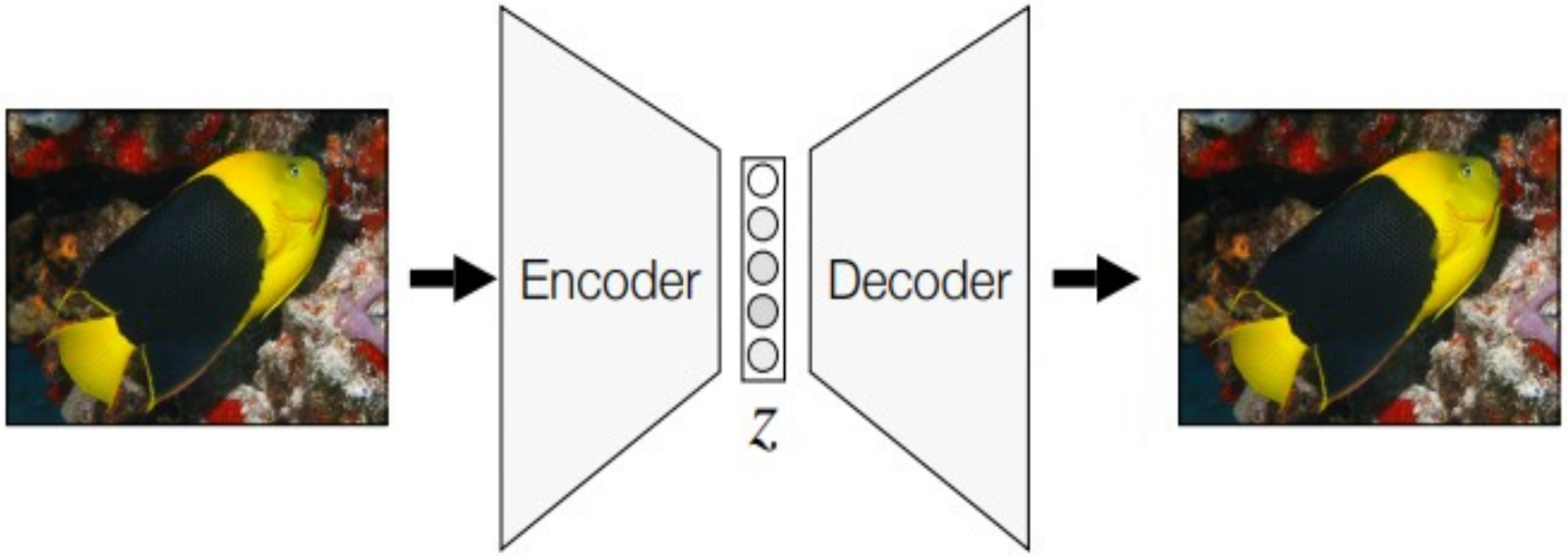
Background

Chia-Ming Wang

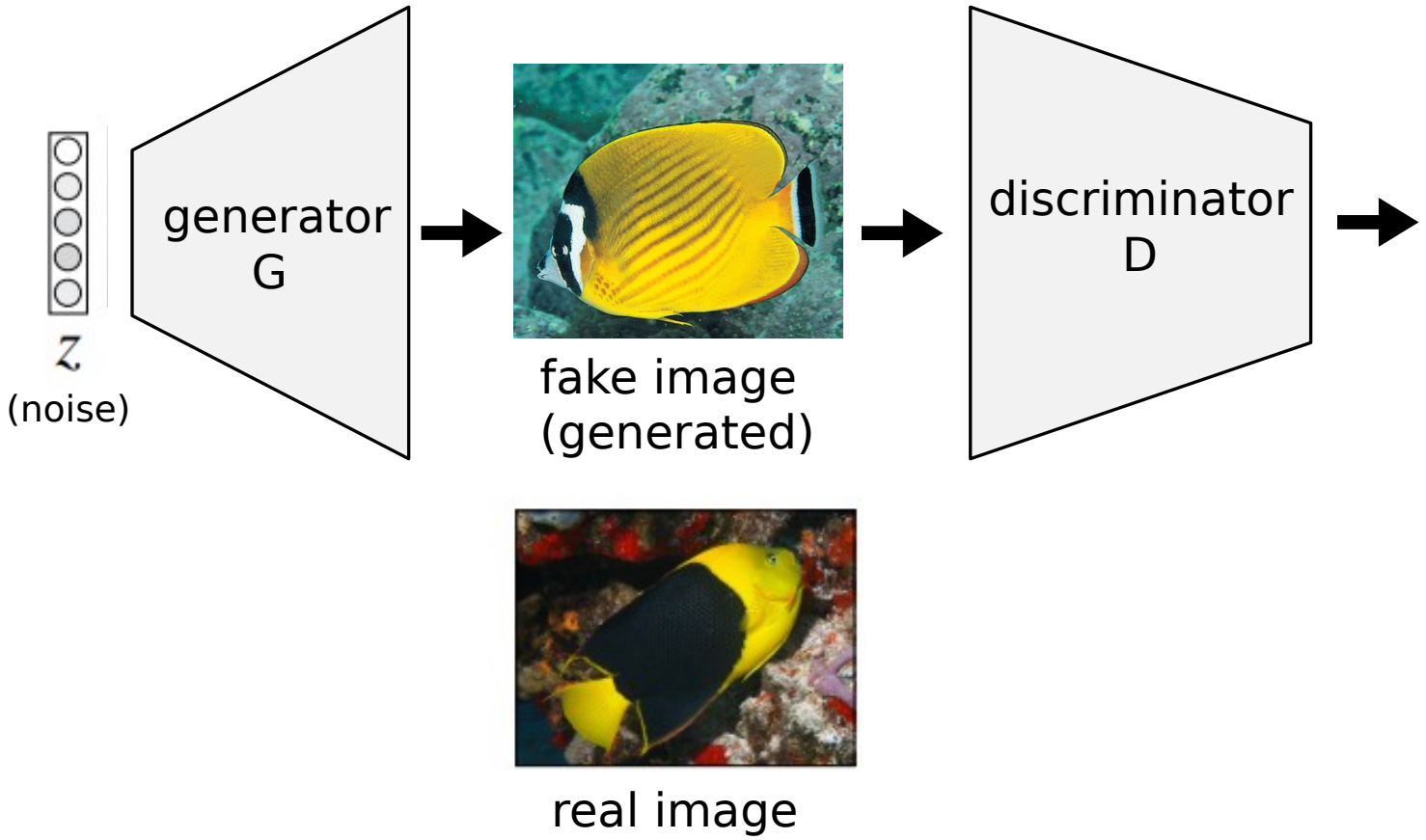
The purpose of GAN



Recall Variational Autoencoders

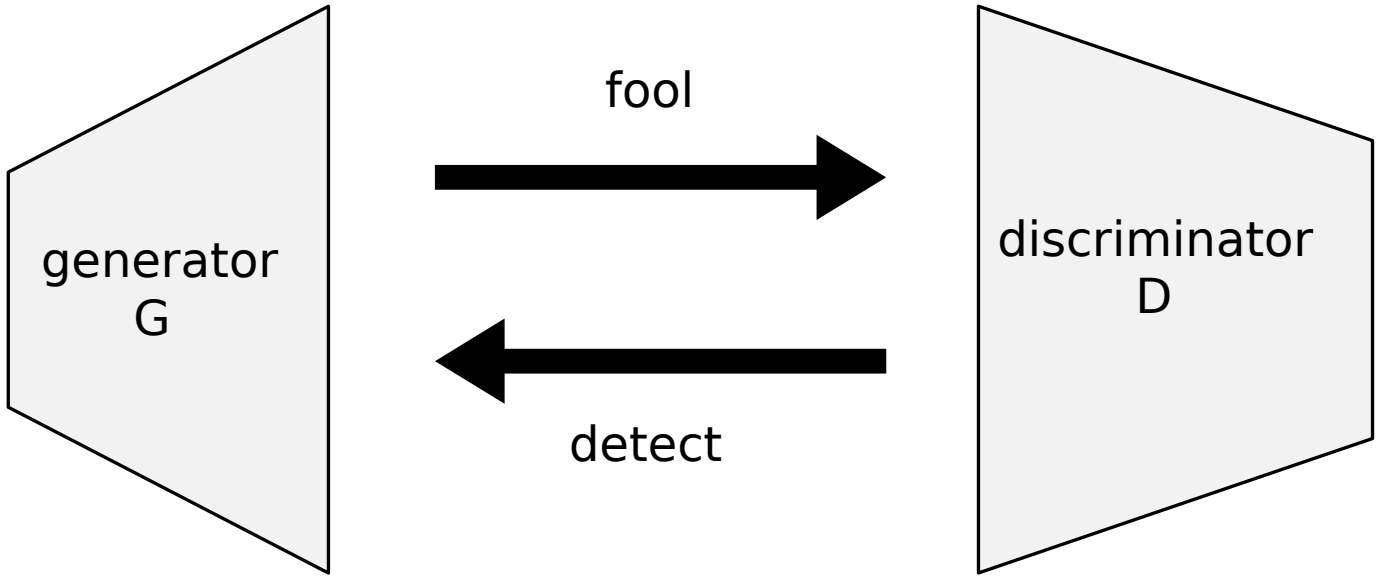


The structure of GAN



meaning of “Adversarial”

- Generator want to fool the discriminator.
- Discriminator wants to figure out the fake images.
- What is the accuracy of D we looking for?
  - 0.5



The loss functions (minmax loss)

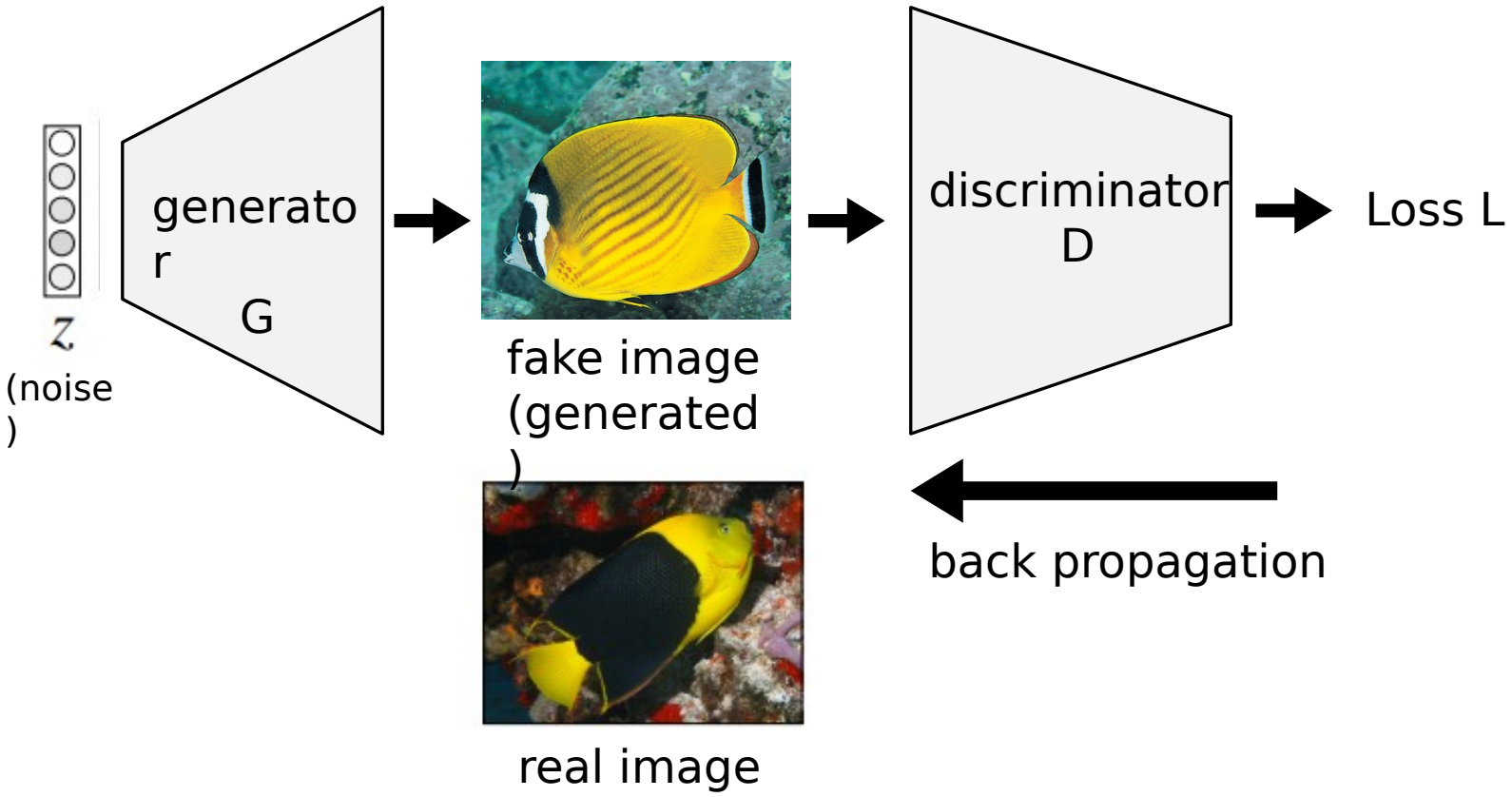
- Loss of Discriminator

$$\arg\max_{\theta}\left(\frac{1}{m}\sum_{i=1}^{m}\left[\log D\left(x^{(i)}\right)+\log\left(1-D\left(G\left(z^{(i)}\right)\right)\right)\right]\right)$$

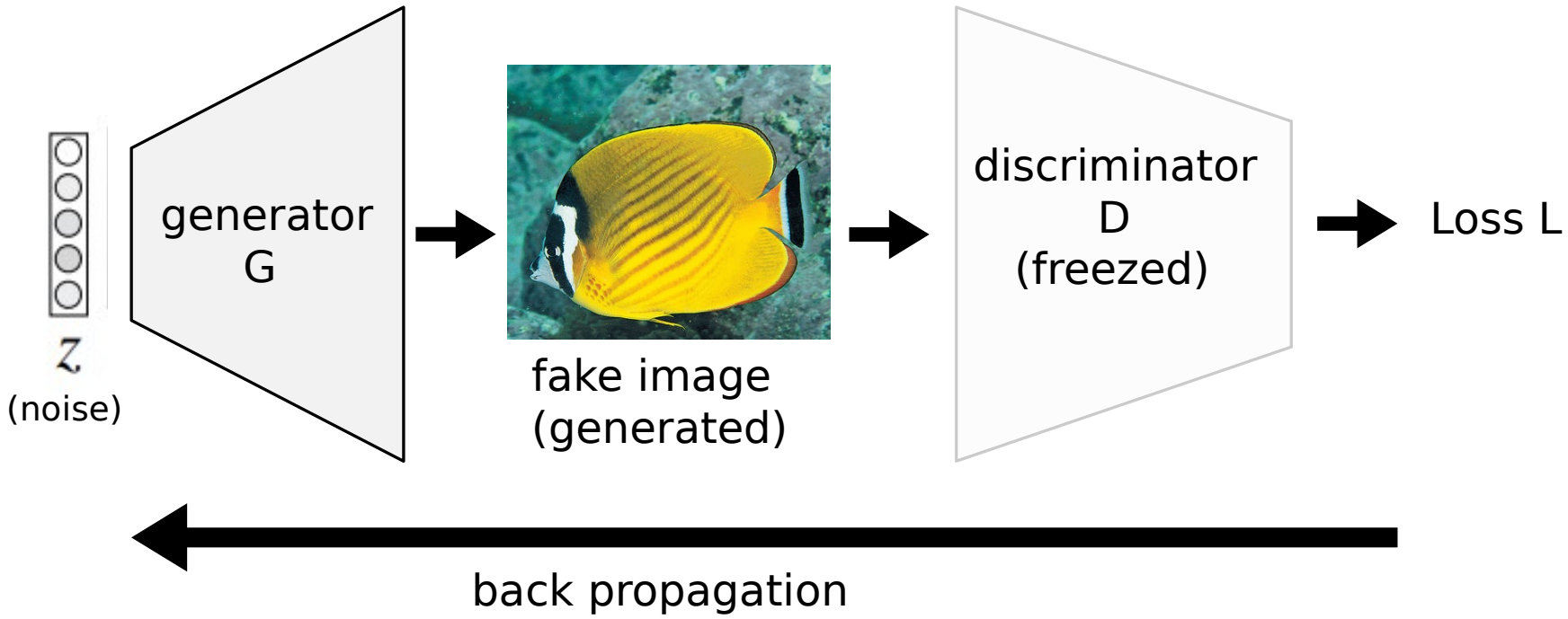
- Loss of Generator

$$\arg\min_{\theta}\left(\frac{1}{m}\sum_{i=1}^{m}\left[\log\left(1-D\left(G\left(z^{(i)}\right)\right)\right)\right]\right)$$

Train Discriminator D



Train Generator G



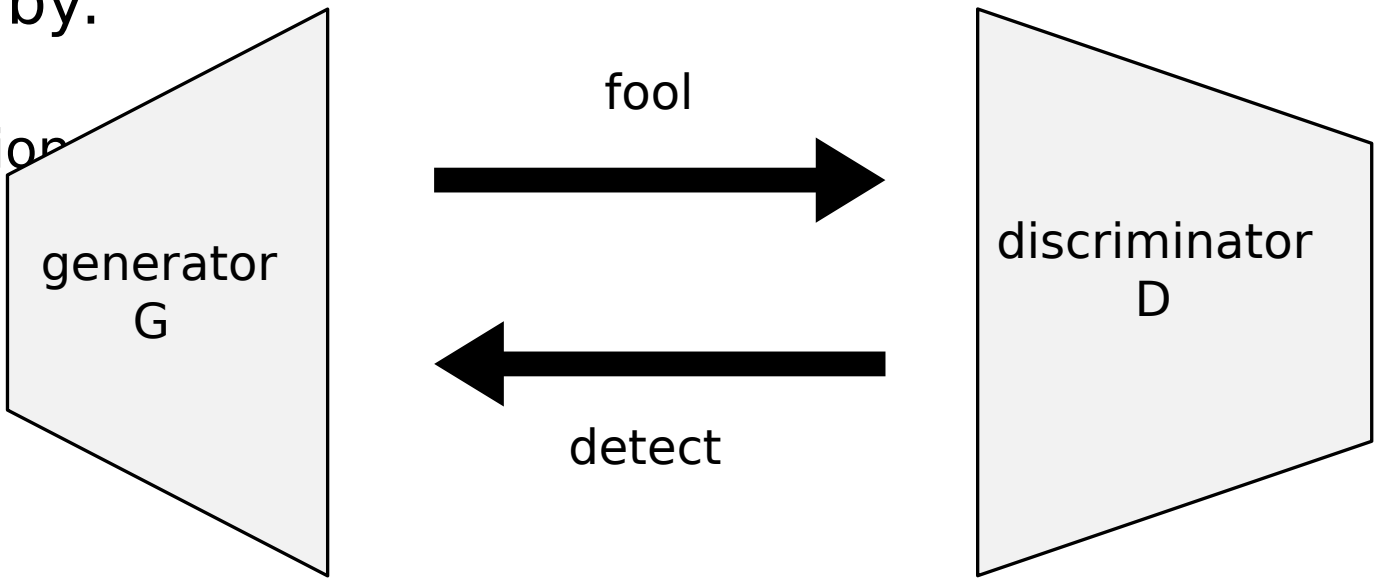
Uses for GANs

- Generate Examples for Image Datasets
- Generate Photographs of Human Faces
- Generate Realistic Photographs
- Generate Cartoon Characters
- Image-to-Image Translation
- Text-to-Image Translation
- Semantic-Image-to-Photo Translation
- Face Frontal View Generation
- Generate New Human Poses
- Photos to Emojis
- Photograph Editing
- Face Aging
- Photo Blending
- Super Resolution
- Photo Inpainting
- Clothing Translation
- Video Prediction
- 3D Object Generation

ref: https://machinelearningmastery.com/impressive-applications-of-generative-adversarial-networks/

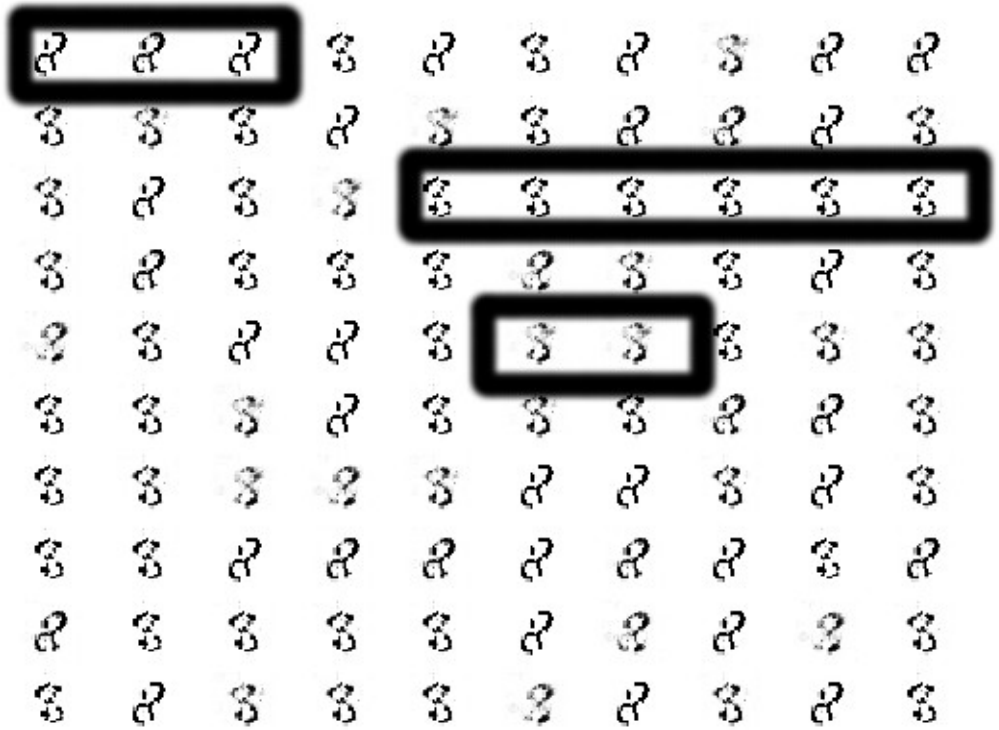
Limits of GANs

- Collapse
  - Imbalance between G & D (usually, D is more powerful)
  - Could be improved by:
    - WGAN/DCGAN
    - spectral normalization



Limits of GANs

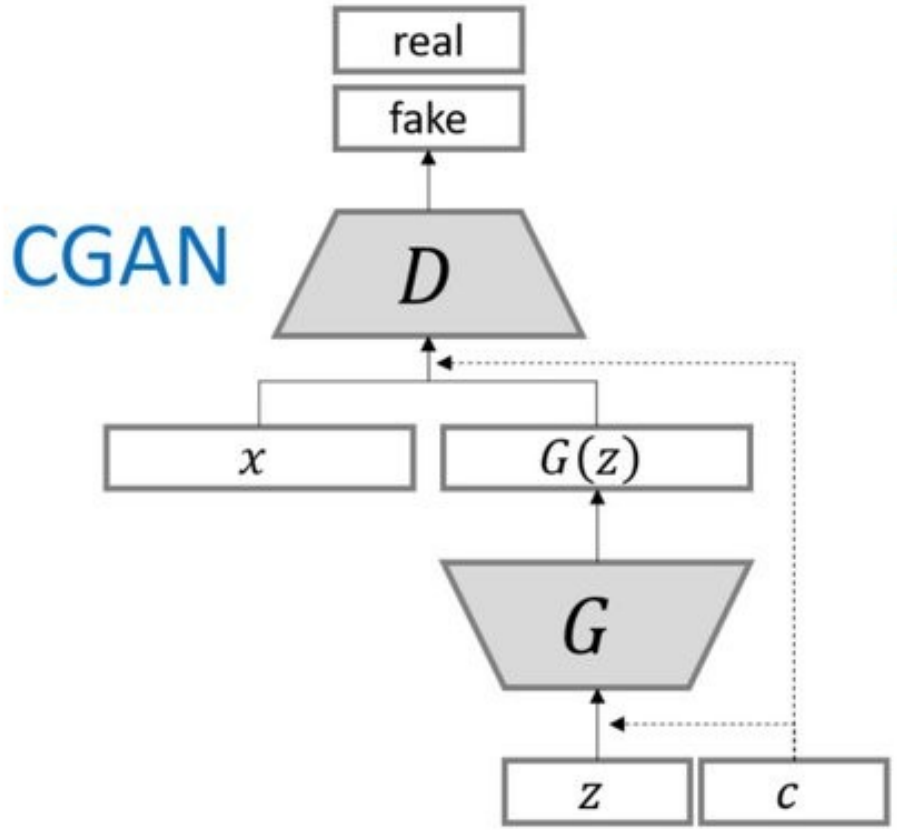
- Collapse
  - Imbalance between G & D
- Mode Collapse
  - No diversity



Limits of GANs

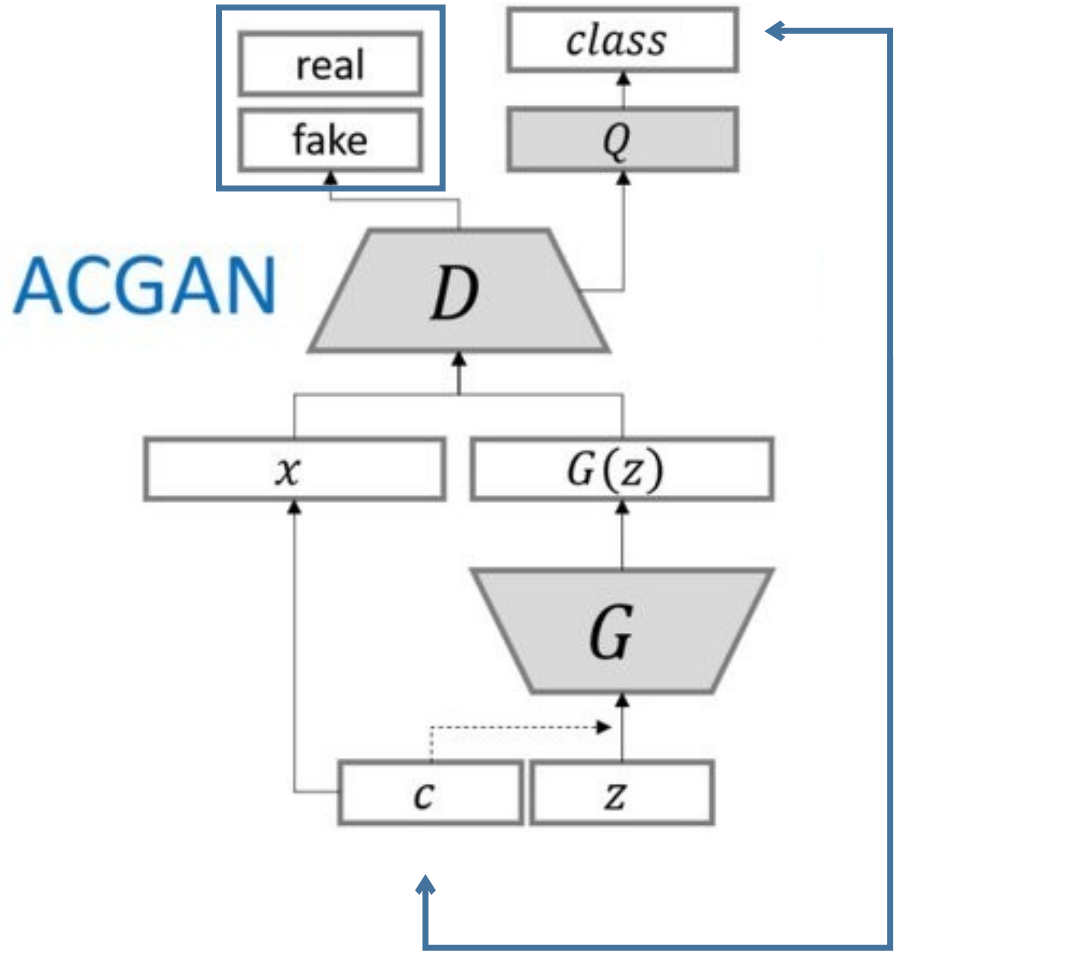
- Collapse
  - Imbalance between G & D
- Mode Collapse
  - No diversity
- Blurred output

Conditional GANs (cGAN/ACGAN)



Minmax Loss

Categorical
Cross Entropy Loss



The loss functions of ACGAN

- L_S: Minmax loss

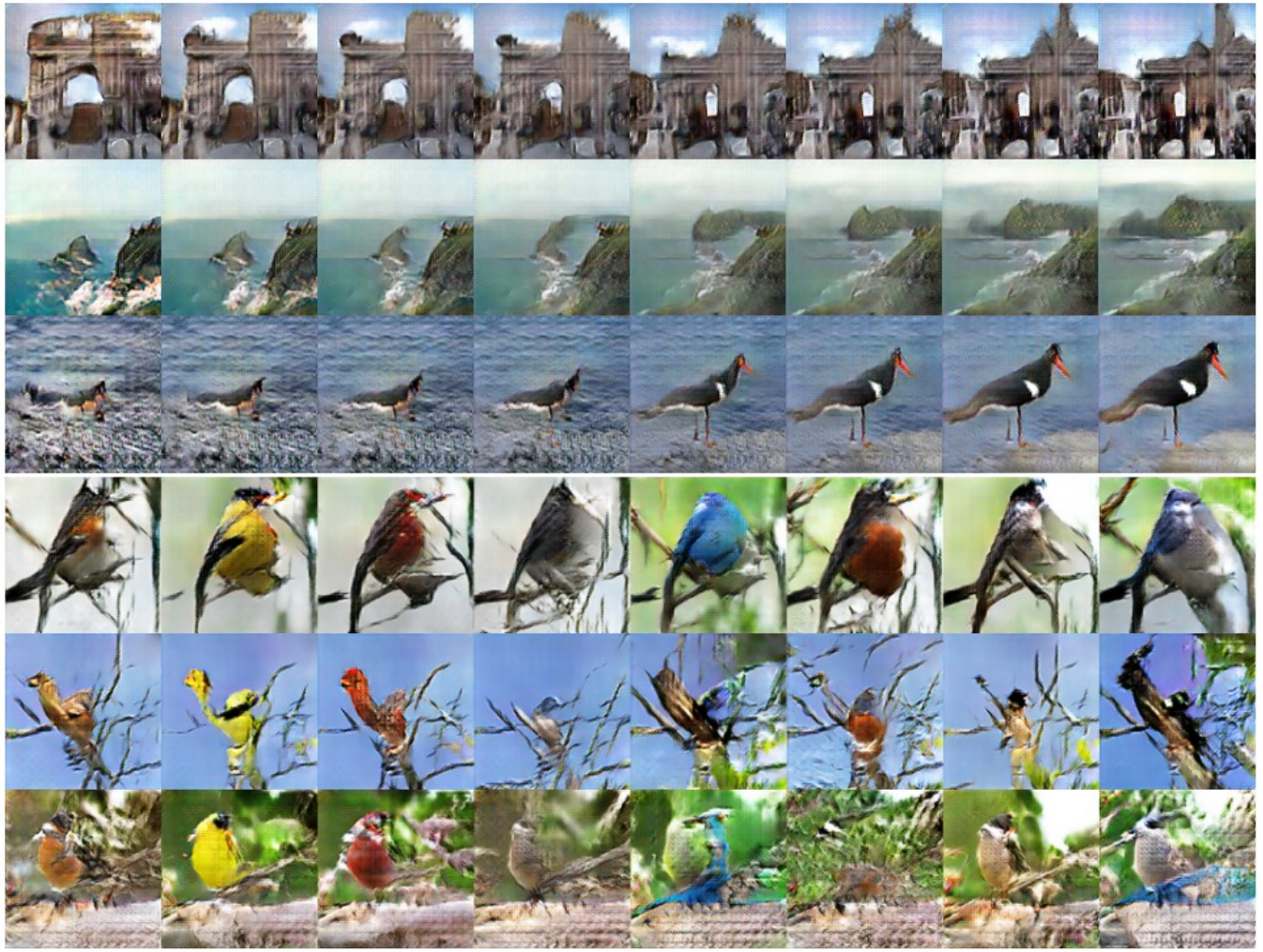
$$L_S = \frac{1}{m}\sum_{i=1}^{m}\left[\log D\left(x^{(i)}\right) + \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)\right]$$

- L_C: Categorical Cross Entropy Loss

$$L_C = E\left[\log P(C = c \mid X_{real})\right] + E\left[\log P(C = c \mid X_{fake})\right]$$

- Train D: argmax(L_s + L_C)
- Train G: argmax(L_C - L_s)

Experiment of ACGAN



Discussion of ‘Rewriting a Deep Generative Model’

Authors: David Bau, Steven Liu, Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba

how to GAN realistic images?

- More params (BigGAN[1]...)
- New layers (AdaIN[2]...)
- New topology (CycleGAN[3], BiGAN[4]...)
- Improve stability: gradient penalty[5], spectral normalization[6] ...

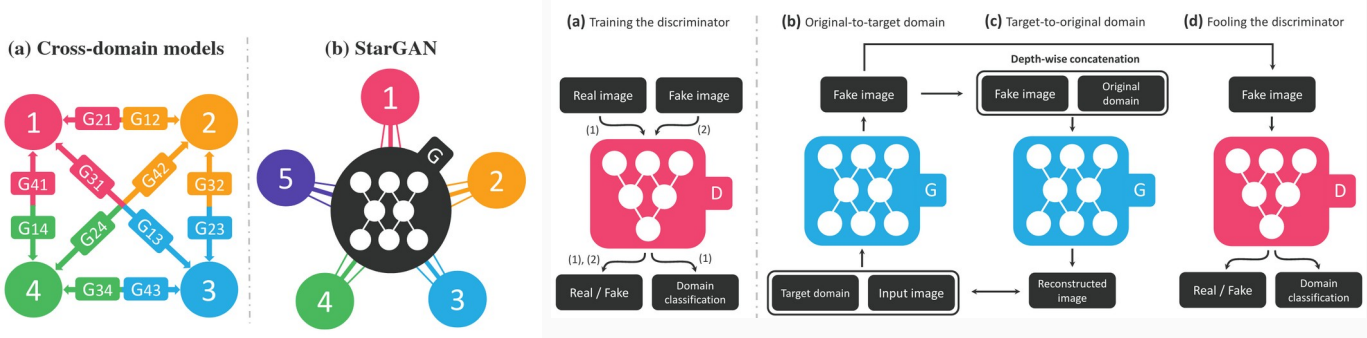
[1] arXiv:1809.11096
[2] arXiv:1703.06868
[3] arXiv:1703.10593
[4] arXiv:1605.09782
[5] arXiv:1704.00028
[6] arXiv:1802.05957

from [1]

how to GAN multi-class images?

- Star-GAN: one-hot label vector[1]

[1] arXiv:1711.09020

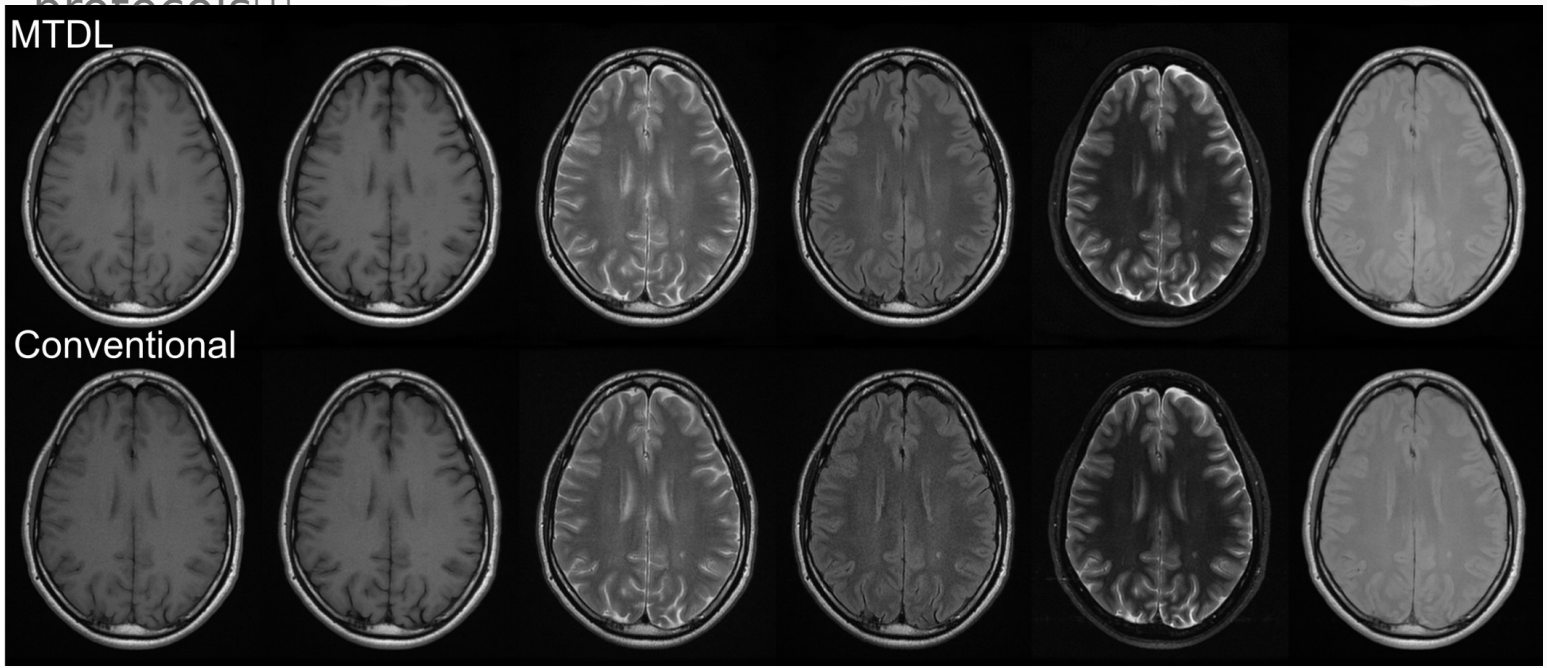


from [1]

how to GAN multi-class images?

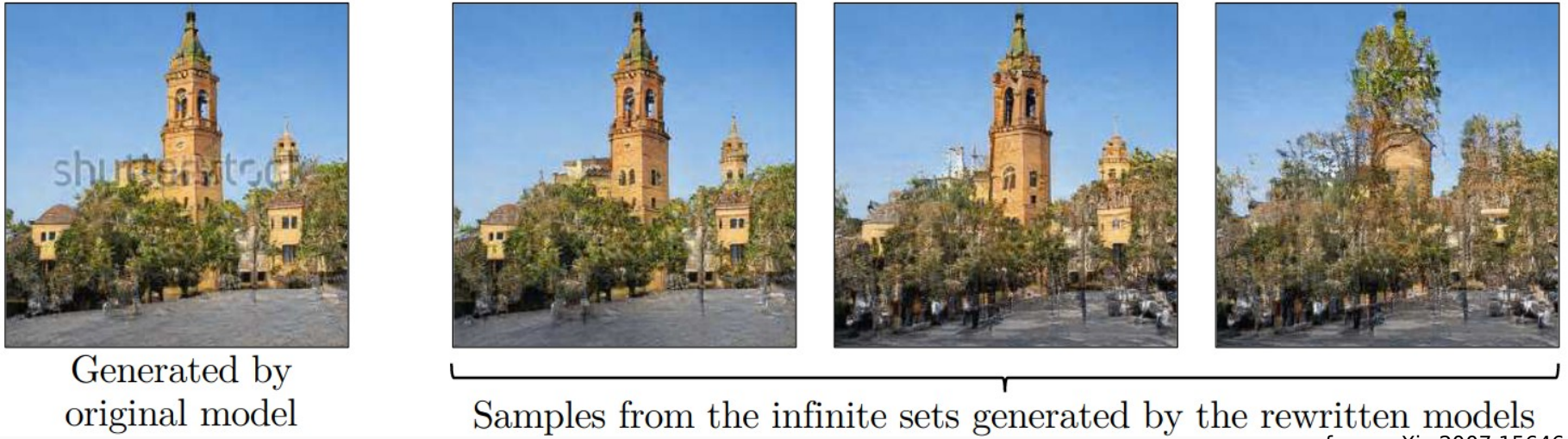
- Example: generate multiple MRI protocols[1]

[1] DOI: 10.1109/TMI.2020.2987026



how to GAN user-defined images?

- Few annotations
- Semantic information



from arXiv:2007.15646

Photoshop 101: copy and paste

from ECCV slides:https://www.youtube.com/watch?v=iBpdJ2CopIE

Update params with Minimal Collateral Damage

$$\theta_1 = \arg\min_{\theta} \mathcal{L}_{\text{smooth}}(\theta) + \lambda \mathcal{L}_{\text{constraint}}(\theta)$$

$$\mathcal{L}_{\text{smooth}}(\theta) \triangleq \mathbb{E}_z[\ell(G(z;\theta_0), G(z;\theta))]$$

$$\mathcal{L}_{\text{constraint}}(\theta) \triangleq \sum_i \ell(x_{*i}, G(z_i;\theta))$$

Update params with Minimal Collateral Damage

- Update only one layer.

$$W_1 = \arg\min_W \mathcal{L}_{\text{smooth}}(W) + \lambda \mathcal{L}_{\text{constraint}}(W)$$

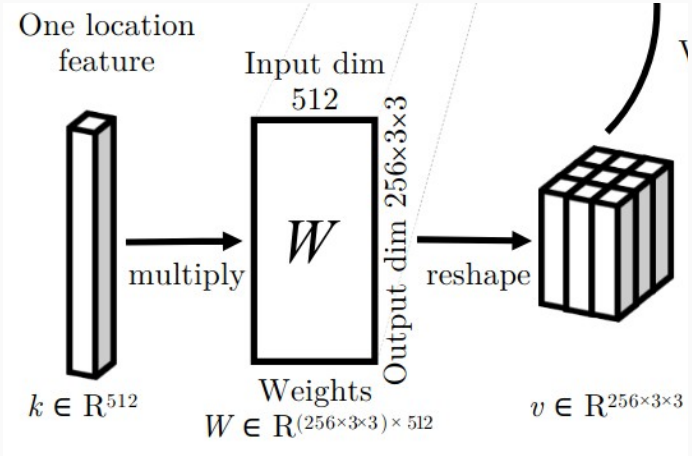
$$\mathcal{L}_{\text{smooth}}(W) \triangleq \mathbb{E}_k \left[\| f(k; W_0) - f(k; W) \|^2 \right]$$

$$\mathcal{L}_{\text{constraint}}(W) \triangleq \sum_i \| v_{*i} - f(k_{*i}; W) \|^2$$

Convolutional layers as Associative Memory

- Convolutional layers as Associative Memory

$$v_i \approx W k_i$$



Update the associative memory

- Initial values

$$W_0 \triangleq \arg\min_W \sum \|v_i - W k_i\|^2$$

$$W_0 = V K^+$$

- Constrained Least-squares

$$W_1 = \arg\min_W \|V - WK\|^2 \qquad \text{smoothness}$$

$$\text{subject to } v_* = W_1 k_* \qquad \text{constraint}$$

$$W_1 = W_0 + \Lambda\left(C^{-1}k_*\right)^T \quad C \triangleq KK^T$$

Update the associative memory

- Initial values

$$W_0 \triangleq \arg\min_W \sum \|v_i - W k_i\|^2$$

$$W_0 = V K^+$$

- Constrained Least-squares

$$W_1 = \arg\min_W \|V - WK\|^2 \quad \text{smoothness}$$

$$\text{subject to } v_* = W_1 k_* \quad \text{constraint}$$

$$W_1 = W_0 + \Lambda\left(C^{-1}k_*\right)^T \qquad C \triangleq KK^T$$

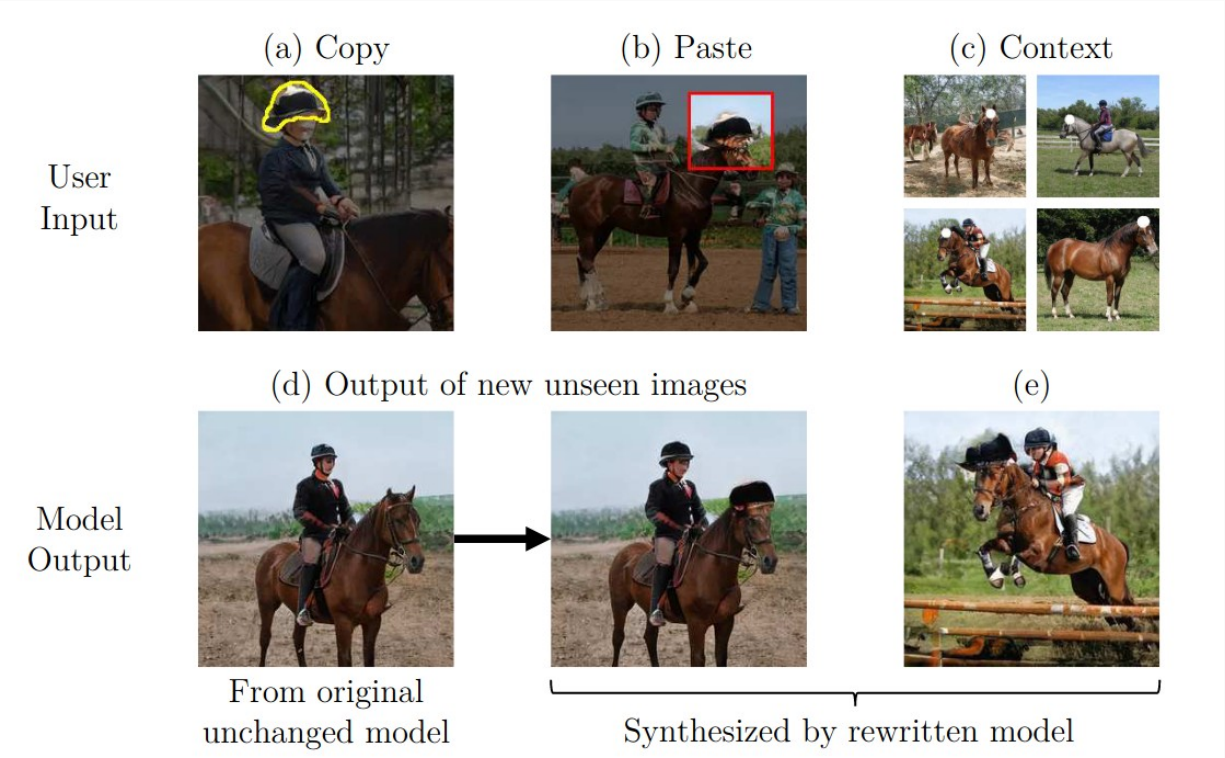
$$\left[\begin{array}{c|c} W_1 & \Lambda \end{array}\right] = \left[\begin{array}{c|c} W_0 & v_* \end{array}\right] \left[\begin{array}{c|c} I & k_* \\ \hline -d^T & 0 \end{array}\right]^{-1}$$

Update the associative memory

- Consider non-linearity

$$\Lambda_1 = \underset{\Lambda \in \mathbb{R}^M}{\arg\min} \left\| v_* - f\left(k_*; W_0 + \Lambda d^T\right) \right\| \quad d \triangleq C^{-1} k_*$$

Copy and paste again



Copy: defining the target value V*

Paste: K* -> V*

Context: establishes the updated direction d (requires dimension reduction by ZCA)

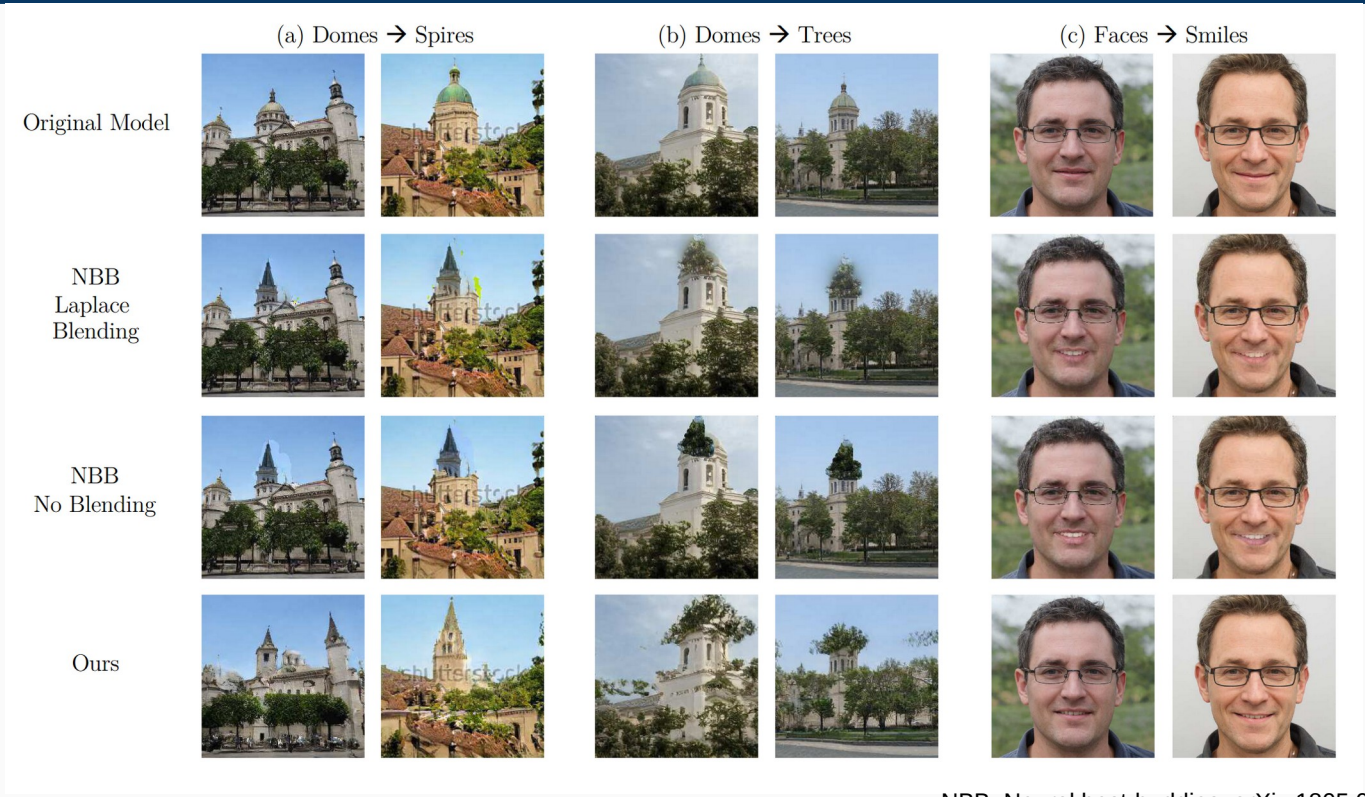
Experiment:

Basic Networks: StyleGAN[1] and Progressive GAN[2]

Image quality evaluation: Amazon Mechanical Turk (AMT) and LPIPS[3]

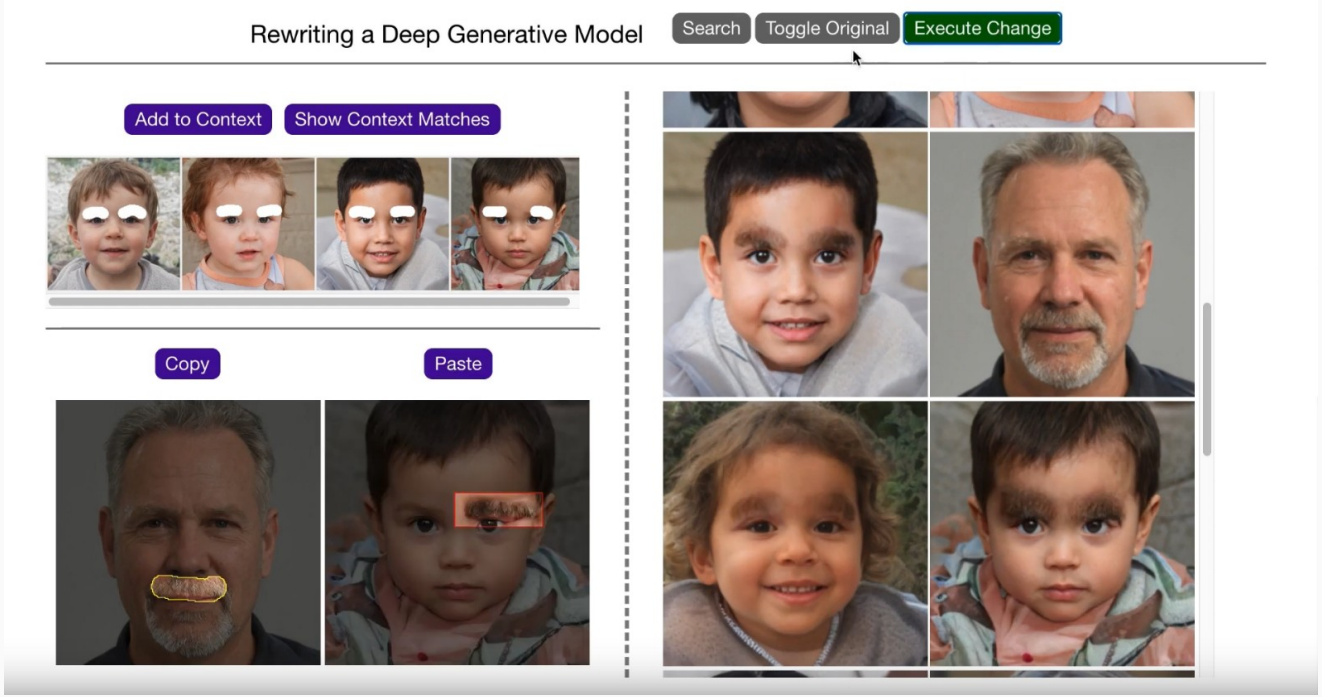
[1] arXiv:1710.10196
[2] arXiv:1812.04948
[3] arXiv:1801.03924

Experiment:



NBB: Neural best-buddies, arXiv:1805.04140

Experiment:



from ECCV slides:https://www.youtube.com/watch?v=iBpdJ2CopIE

Discussion

- Which layer to rewrite?
- Compression of context (rank)
- Dependence on the dataset
- Interpretation of the associative memory

Large Scale Adversarial Representation Learning

Authors: Jeff Donahue and Karen Simonyan

Presented by: Nathan Louis

What is Representation Learning?

- (i.e feature learning) Automatically discover features useful for downstream tasks directly from raw data
  - Classification
  - Detection

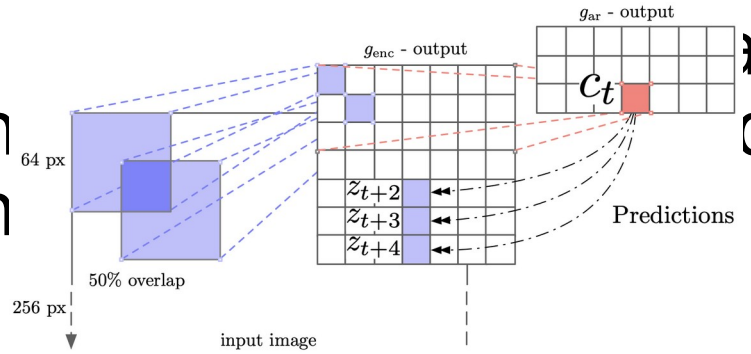
Common self-supervised methods

- This work proposes to use generators inste
- **Assumption:** If yo a in a ist h ders f th

Colorization

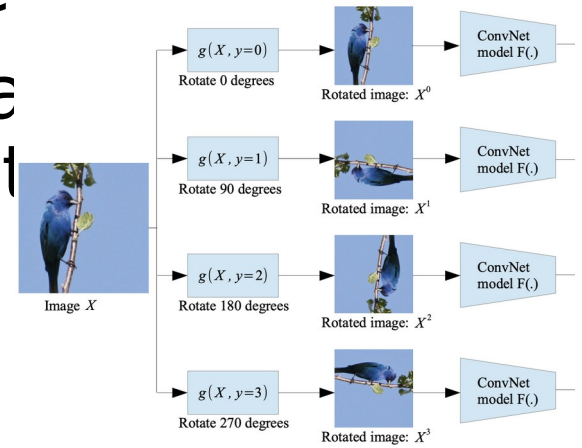
Zhang et al. ECCV 2016

Contrastive Predictive Coding (CPC)



Oord et al. arXiV 2018

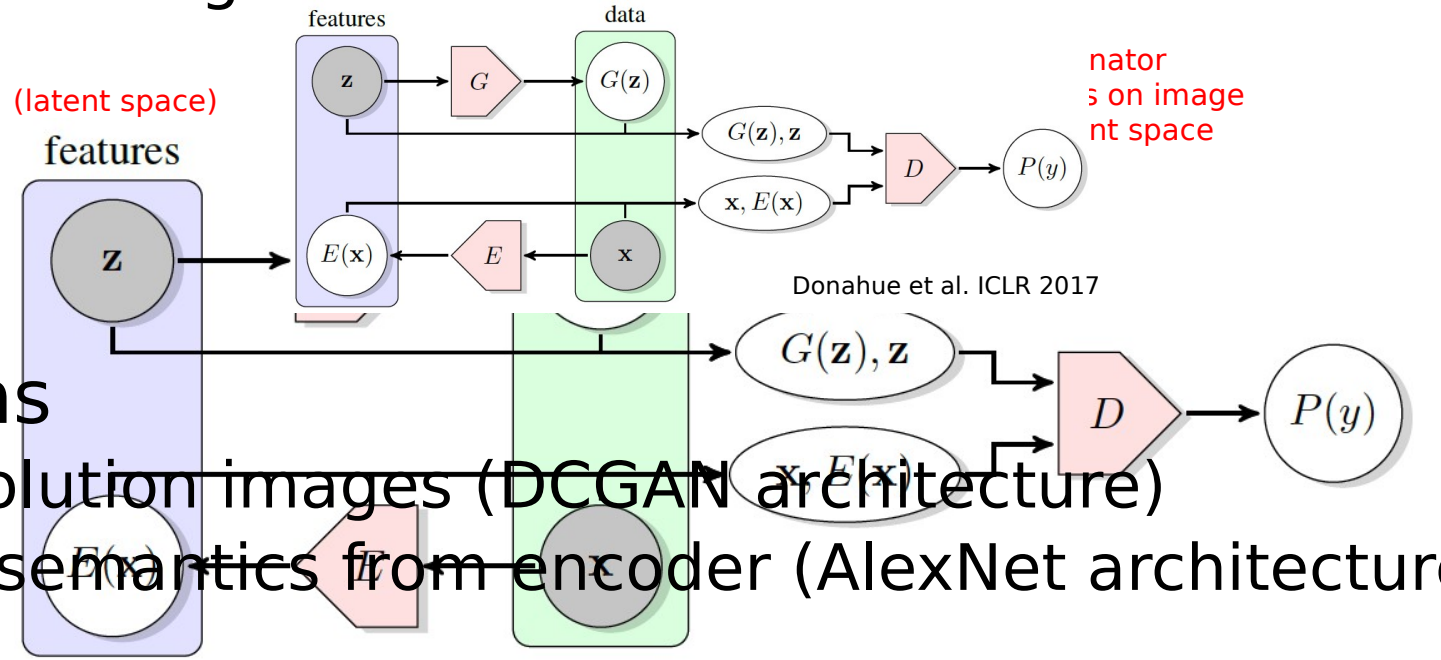
Image Rotations



Gidaris et al. arXiV 2018

Prior work from author

- BiGAN
  - Main contribution: Simultaneously learn an encoder as an inverse to the generator in GANs



Structure of BiGAN

Donahue et al. ICLR 2017

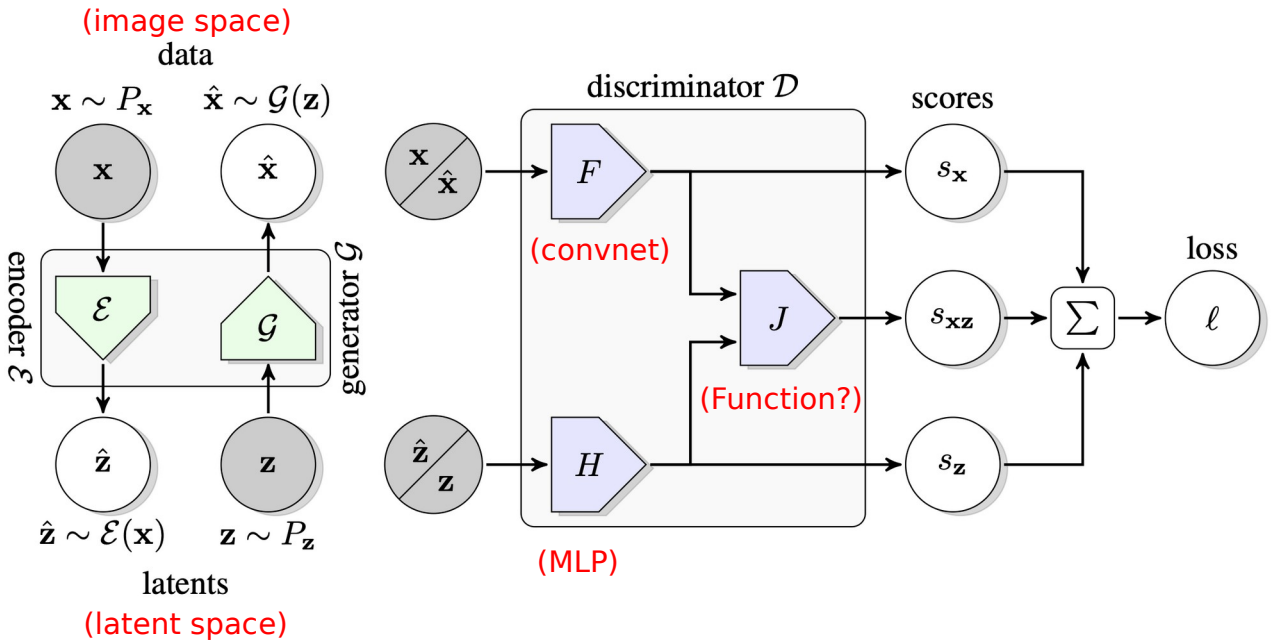
- Limitations
  - Low resolution images (DCGAN architecture)
  - Limited semantics from encoder (AlexNet architecture)

Contributions

- Presents BigBiGAN
  - BiGAN approach combined with BigGAN (more powerful) architecture
- Empirical analyses
- State of the art results in unsupervised representation learning from GANs
  - In comparison to both self-supervised and unsupervised methods

Model Overview

- Updated discriminator structure
- Unary terms in loss function



Scores

$$s_{\mathbf{x}}(\mathbf{x}) = \theta_{\mathbf{x}}^{\intercal} F_{\Theta}(\mathbf{x})$$
$$s_{\mathbf{z}}(\mathbf{z}) = \theta_{\mathbf{z}}^{\intercal} H_{\Theta}(\mathbf{z})$$
$$s_{\mathbf{xz}}(\mathbf{x}, \mathbf{z}) = \theta_{\mathbf{xz}}^{\intercal} J_{\Theta}(F_{\Theta}(\mathbf{x}), H_{\Theta}(\mathbf{z}))$$

$y \in \{-1, +1\}$

Encoder-Generator Loss

$$\ell_{\mathcal{EG}}(\mathbf{x}, \mathbf{z}, y) = y\,(s_{\mathbf{x}}(\mathbf{x}) + s_{\mathbf{z}}(\mathbf{z}) + s_{\mathbf{xz}}(\mathbf{x}, \mathbf{z}))$$
$$\mathcal{L}_{\mathcal{EG}}(P_{\mathbf{x}}, P_{\mathbf{z}}) = \mathbb{E}_{\mathbf{x} \sim P_{\mathbf{x}}, \hat{\mathbf{z}} \sim \mathcal{E}_{\Phi}(\mathbf{x})}\left[\ell_{\mathcal{EG}}(\mathbf{x}, \hat{\mathbf{z}}, +1)\right] + \mathbb{E}_{\mathbf{z} \sim P_{\mathbf{z}}, \hat{\mathbf{x}} \sim \mathcal{G}_{\Phi}(\mathbf{z})}\left[\ell_{\mathcal{EG}}(\hat{\mathbf{x}}, \mathbf{z}, -1)\right]$$

[real image] [generated image]

Discriminator Loss

$$\ell_{\mathcal{D}}(\mathbf{x}, \mathbf{z}, y) = h(y s_{\mathbf{x}}(\mathbf{x})) + h(y s_{\mathbf{z}}(\mathbf{z})) + h(y s_{\mathbf{xz}}(\mathbf{x}, \mathbf{z}))$$
$$\mathcal{L}_{\mathcal{D}}(P_{\mathbf{x}}, P_{\mathbf{z}}) = \mathbb{E}_{\mathbf{x} \sim P_{\mathbf{x}}, \hat{\mathbf{z}} \sim \mathcal{E}_{\Phi}(\mathbf{x})}\left[\ell_{\mathcal{D}}(\mathbf{x}, \hat{\mathbf{z}}, +1)\right] + \mathbb{E}_{\mathbf{z} \sim P_{\mathbf{z}}, \hat{\mathbf{x}} \sim \mathcal{G}_{\Phi}(\mathbf{z})}\left[\ell_{\mathcal{D}}(\hat{\mathbf{x}}, \mathbf{z}, -1)\right]$$

Evaluation and Experiments

- ImageNet Classification
  - Train a linear classifier on encoder outputs
  - Report top-1 classification accuracy
- Image Generation
  - Inception Score (IS)
    - Measure of quality and variety of generated images
  - Fréchet Inception Distance (FID)
    - Measures feature similarity between real and generated images
- Plenty of Ablations

Ablations

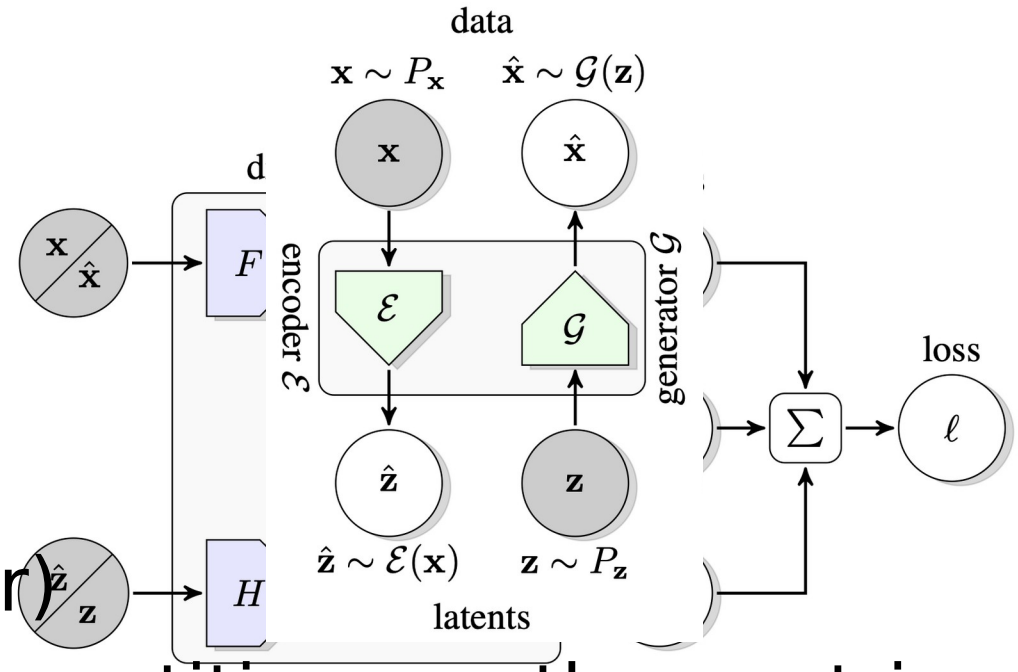
- Stochastic Encoder vs Deterministic Encoder
  - The encoder output is non-deterministic and reparametrized as a gaussian random variable
  - Compared to a deterministic encoder and BiGAN encoder (-1,1)

	Encoder ($\mathcal{E}$)						Gen. ($\mathcal{G}$)		Loss $\mathcal{L}_*$				Results		
	A.	D.	C.	R.	Var.	η	C.	R.	$s_{\mathbf{xz}}$	$s_{\mathbf{x}}$	$s_{\mathbf{z}}$	$P_{\mathbf{z}}$	IS (↑)	FID (↓)	Cls. (↑)
Base	S	50	1	128	✓	1	96	128	✓	✓	✓	$\mathcal{N}$	22.66 ± 0.18	31.19 ± 0.37	48.10 ± 0.13
Deterministic $\mathcal{E}$	S	50	1	128	(-)	1	96	128	✓	✓	✓	$\mathcal{N}$	22.79 ± 0.27	31.31 ± 0.30	46.97 ± 0.35
Uniform $P_{\mathbf{z}}$	S	50	1	128	(-)	1	96	128	✓	✓	✓	($\mathcal{U}$)	22.83 ± 0.24	31.52 ± 0.28	45.11 ± 0.93

Why does the non-deterministic encoder provide better classification results?

Ablations

- Unary loss term
- Generator capacity
- Standard GAN (no encoder)
- Good for classification, competitive on other metrics



	Encoder ($\mathcal{E}$)						Gen. ($\mathcal{G}$)		Loss $\mathcal{L}_*$			$P_\mathbf{z}$	Results		
	A.	D.	C.	R.	Var.	η	C.	R.	$s_{\mathbf{xz}}$	$s_\mathbf{x}$	$s_\mathbf{z}$		IS (↑)	FID (↓)	Cls. (↑)
Base	S	50	1	128	✓	1	96	128	✓	✓	✓	$\mathcal{N}$	22.66 ± 0.18	31.19 ± 0.37	48.10 ± 0.13
$\mathbf{x}$ Unary Only	S	50	1	128	✓	1	96	128	✓	✓	(-)	$\mathcal{N}$	23.19 ± 0.28	31.99 ± 0.30	47.74 ± 0.20
$\mathbf{z}$ Unary Only	S	50	1	128	✓	1	96	128	✓	(-)	✓	$\mathcal{N}$	19.52 ± 0.39	39.48 ± 1.00	47.78 ± 0.28
No Unaries (BiGAN)	S	50	1	128	✓	1	96	128	✓	(-)	(-)	$\mathcal{N}$	19.70 ± 0.30	42.92 ± 0.92	46.71 ± 0.88
Small $\mathcal{G}$ (32)	S	50	1	128	✓	1	(32)	128	✓	✓	✓	$\mathcal{N}$	3.28 ± 0.18	247.30 ± 10.31	43.59 ± 0.34
Small $\mathcal{G}$ (64)	S	50	1	128	✓	1	(64)	128	✓	✓	✓	$\mathcal{N}$	19.96 ± 0.15	38.93 ± 0.39	47.54 ± 0.33
No $\mathcal{E}$ (GAN) *	(-)						96	128	(-)	✓	(-)	$\mathcal{N}$	23.56 ± 0.37	30.91 ± 0.23	-

Ablations

- Varying input and output image resolutions
- Encoder architecture
- Decoupling optimization between Generator and Encoder

	Encoder ($\mathcal{E}$)						Gen. ($\mathcal{G}$)		Loss $\mathcal{L}_*$				Results		
	A.	D.	C.	R.	Var.	η	C.	R.	s_{xz}	s_x	s_z	P_z	IS (↑)	FID (↓)	Cls. (↑)
Base	S	50	1	128	✓	1	96	128	✓	✓	✓	$\mathcal{N}$	22.66 ± 0.18	31.19 ± 0.37	48.10 ± 0.13
High Res $\mathcal{E}$ (256)	S	50	1	(256)	✓	1	96	128	✓	✓	✓	$\mathcal{N}$	23.45 ± 0.14	27.86 ± 0.13	50.80 ± 0.30
Low Res $\mathcal{G}$ (64)	S	50	1	(256)	✓	1	96	(64)	✓	✓	✓	$\mathcal{N}$	19.40 ± 0.19	15.82 ± 0.06	47.51 ± 0.09
High Res $\mathcal{G}$ (256)	S	50	1	(256)	✓	1	96	(256)	✓	✓	✓	$\mathcal{N}$	24.70	38.58	51.49
ResNet-101	S	(101)	1	(256)	✓	1	96	128	✓	✓	✓	$\mathcal{N}$	23.29	28.01	51.21
ResNet ×2	S	50	(2)	(256)	✓	1	96	128	✓	✓	✓	$\mathcal{N}$	23.68	27.81	52.66
RevNet	(V)	50	1	(256)	✓	1	96	128	✓	✓	✓	$\mathcal{N}$	23.33 ± 0.09	27.78 ± 0.06	49.42 ± 0.18
RevNet ×2	(V)	50	(2)	(256)	✓	1	96	128	✓	✓	✓	$\mathcal{N}$	23.21	27.96	54.40
RevNet ×4	(V)	50	(4)	(256)	✓	1	96	128	✓	✓	✓	$\mathcal{N}$	23.23	28.15	57.15
ResNet (↑ $\mathcal{E}$ LR)	S	50	1	(256)	✓	(10)	96	128	✓	✓	✓	$\mathcal{N}$	23.27 ± 0.22	28.51 ± 0.44	53.70 ± 0.15
RevNet ×4 (↑ $\mathcal{E}$ LR)	(V)	50	(4)	(256)	✓	(10)	96	128	✓	✓	✓	$\mathcal{N}$	23.08	28.54	60.15

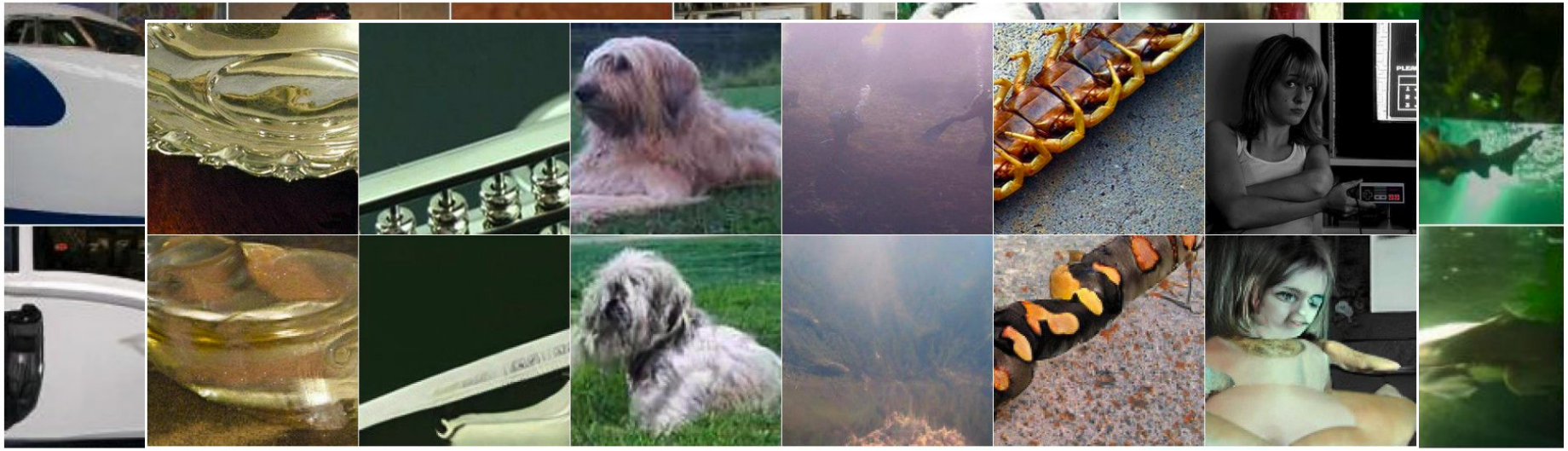
Comparisons to existing work

- ImageNet Classification against unsupervised and self-supervised methods
- Outperforms other methods, but comparable to Efficient CPC

Method	Architecture	Feature	Top-1	Top-5
BiGAN [7, 42]	AlexNet	Conv3	31.0	-
SS-GAN [4]	ResNet-19	Block6	38.3	-
Motion Segmentation (MS) [30, 6]	ResNet-101	AvePool	27.6	48.3
Exemplar (Ex) [8, 6]	ResNet-101	AvePool	31.5	53.1
Relative Position (RP) [5, 6]	ResNet-101	AvePool	36.2	59.2
Colorization (Col) [41, 6]	ResNet-101	AvePool	39.6	62.5
Combination of MS+Ex+RP+Col [6]	ResNet-101	AvePool	-	69.3
CPC [39]	ResNet-101	AvePool	48.7	73.6
Rotation [11, 24]	RevNet-50 ×4	AvePool	55.4	-
Efficient CPC [17]	ResNet-170	AvePool	61.0	83.0
BigBiGAN (ours)	ResNet-50	AvePool	55.4	77.4
	ResNet-50	BN+CReLU	56.6	78.6
	RevNet-50 ×4	AvePool	60.8	81.4
	RevNet-50 ×4	BN+CReLU	61.3	81.9

Image Reconstruction

- How much information is encoded in their latent space?
- Note: Their loss function does not enforce a reconstruction loss, but high-level details and semantics aris



Large Architecture

Unsupervised Generation

- Single-Label: A single “dummy” label is used for all images
- Cl

Method	.. (↓)
BigGA	66.32)
BigGA	0.80)
BigBiG	± 0.12)
BigBiG	± 0.15)
BigBiG	± 0.09)

High Fidelity Natural Image Generator

Lighter Augmentation

Conclusion

- Generators are powerful tools for learning features from unlabeled images
- The biggest bottleneck is model capacity
  - Larger networks = Better results