

Supervised vs unsupervised learning

Supervised Learning

Data: (x, y)
 x is data, y is label

Goal: Learn function to map
 $x \rightarrow y$

Examples: Classification,
regression, object detection,
semantic segmentation, etc.

Unsupervised Learning

Data: x
 x is data, no labels!

Goal: Learn some *hidden* or
underlying structure of the data

Examples: Clustering, feature or
dimensionality reduction, etc.

Supervised vs unsupervised learning

Supervised Learning

Data: (x, y)
 x is data, y is label

Goal: Learn function to map
 $x \rightarrow y$

Examples: Classification,
regression, object detection,
semantic segmentation, etc.

Unsupervised Learning

Data: x
 x is data, no labels!

Goal: Learn the *hidden* or
underlying structure of the data

Examples: Clustering, feature or
dimensionality reduction, etc.

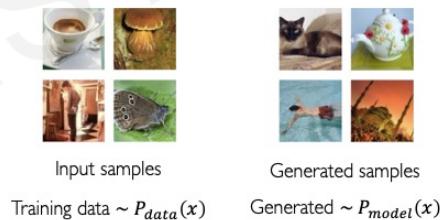
Generative modeling

Goal: Take as input training samples from some distribution
and learn a model that represents that distribution

Density Estimation



Sample Generation



How can we learn $P_{model}(x)$ similar to $P_{data}(x)$?

Why generative models? Debiasing

Capable of uncovering **underlying features** in a dataset



Homogeneous skin color, pose



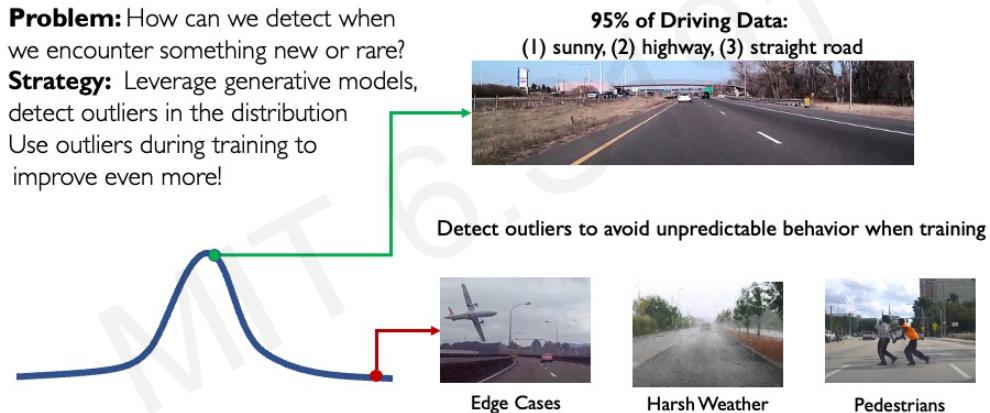
Diverse skin color, pose, illumination

How can we use this information to create fair and representative datasets?



Why generative models? Outlier detection

- Problem:** How can we detect when we encounter something new or rare?
- Strategy:** Leverage generative models, detect outliers in the distribution
- Use outliers during training to improve even more!

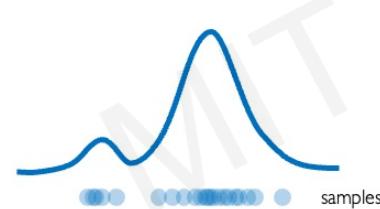


MIT Introduction to Deep Learning
introtodeeplearning.com @MITDeepLearning

Amini+ IROS 2018. 1/7/25

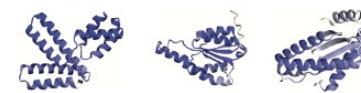
Why generative models? Sample generation

- Generative models learn probability distributions**
- Sampling** from that distribution → new data instances
- Backbone of Generative AI:** generate language, images, and more



Natural Language
6.S191 Guest Lectures!

Images & Videos



Biology
6.S191 Guest Lecture!

MIT Introduction to Deep Learning
introtodeeplearning.com @MITDeepLearning

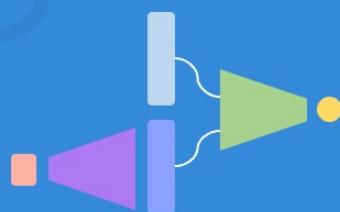
1/7/25

Latent variable models

Autoencoders and Variational Autoencoders (VAEs)



Generative Adversarial Networks (GANs)



Autoencoders

Autoencoders: background

Unsupervised approach for learning a **lower-dimensional** feature representation from unlabeled training data



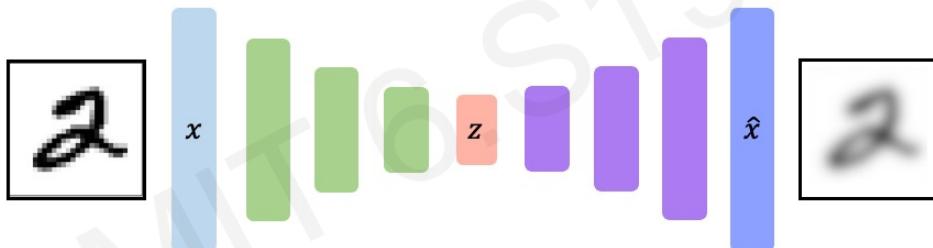
"Encoder" learns mapping from the data, x , to a low-dimensional latent space, z



Autoencoders: background

How can we learn this latent space?

Train the model to use these features to **reconstruct the original data**

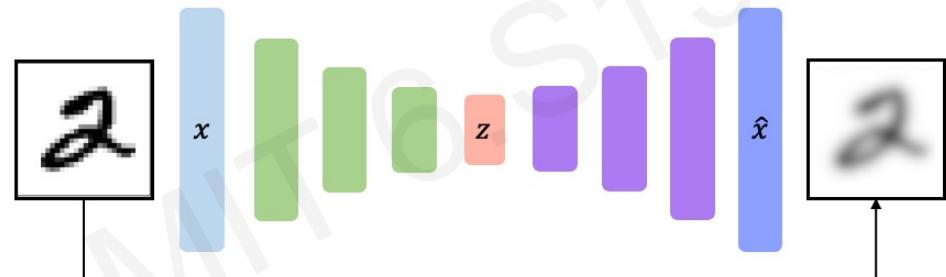


"Decoder" learns mapping back from latent space, z , to a reconstructed observation, \hat{x}

Autoencoders: background

How can we learn this latent space?

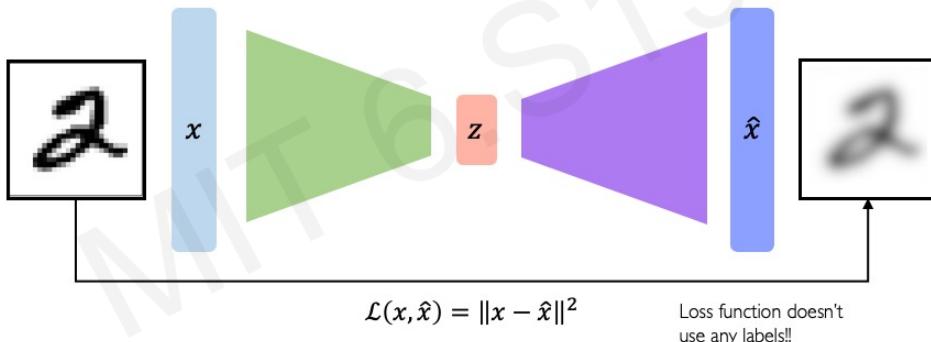
Train the model to use these features to **reconstruct the original data**



Autoencoders: background

How can we learn this latent space?

Train the model to use these features to **reconstruct the original data**



Dimensionality of latent space → reconstruction quality

Autoencoding is a form of compression!

Smaller latent space will force a larger training bottleneck

2D latent space

7	2	1	0	4	1	4	9	9	9
0	6	9	0	1	5	9	7	3	4
9	6	6	5	4	0	7	4	0	1
3	1	3	0	7	2	7	1	2	1
1	7	9	2	3	5	1	2	9	4
6	3	5	5	6	0	4	1	9	8
7	8	4	2	7	9	6	4	3	0
7	0	2	9	1	9	3	2	9	7
9	6	2	7	8	4	7	3	6	1
3	6	9	3	1	4	1	7	6	9

5D latent space

7	2	1	0	4	1	4	9	9	9
0	6	9	0	1	5	9	7	3	4
9	6	6	5	4	0	7	4	0	1
3	1	3	4	7	2	7	1	2	1
1	7	4	4	3	5	1	2	9	4
6	3	5	5	6	0	4	1	9	8
7	8	4	3	7	9	4	6	4	3
7	0	2	9	1	7	3	2	9	7
9	6	2	7	8	4	7	3	6	1
3	6	9	3	1	4	1	7	6	9

Ground Truth

7	2	1	0	4	1	4	9	5	9
0	6	9	0	1	5	9	7	3	4
9	6	6	5	4	0	7	4	0	1
3	1	3	4	7	2	7	1	2	1
1	7	4	4	3	5	1	2	9	4
6	3	5	5	6	0	4	1	9	5
7	8	5	3	7	4	6	4	3	0
7	0	2	9	1	7	3	2	9	7
9	6	2	7	8	4	7	3	6	1
3	6	9	3	1	4	1	7	6	9

Autoencoders for representation learning

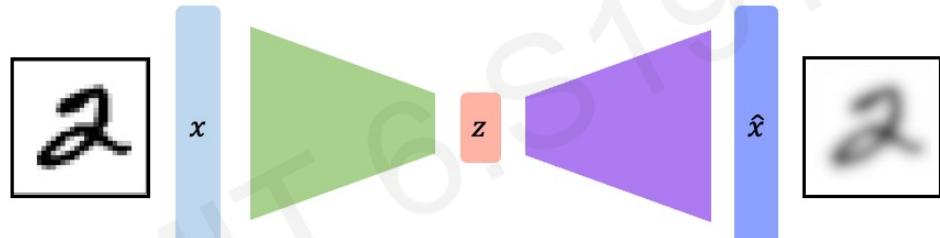
Bottleneck hidden layer forces network to learn a compressed latent representation

Reconstruction loss forces the latent representation to capture (or encode) as much "information" about the data as possible

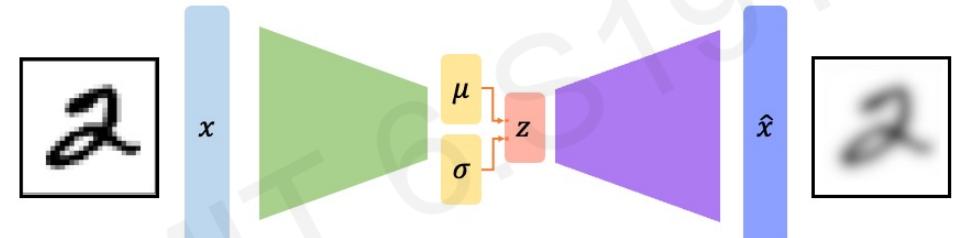
Autoencoding = **A**utomatically **e**ncoding data; "Auto" = **s**elf-encoding

Variational Autoencoders (VAEs)

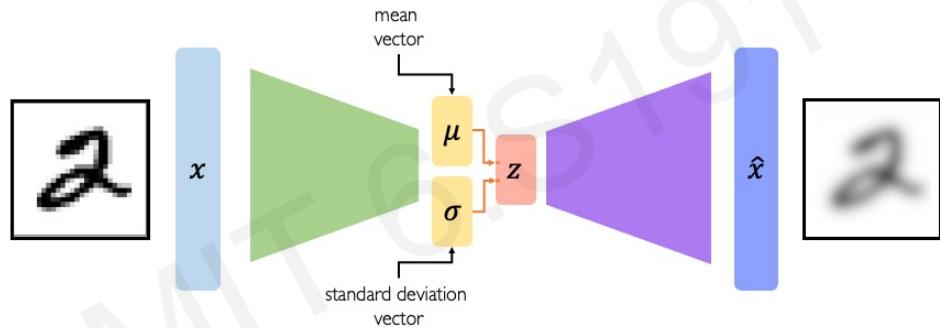
Traditional autoencoders



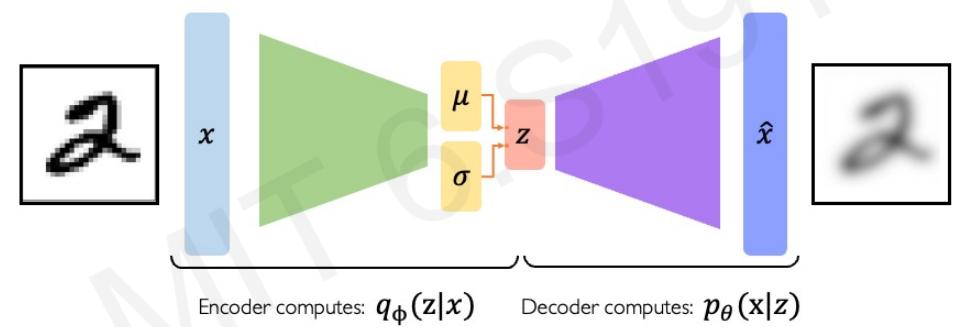
VAEs: key difference with traditional autoencoder



VAEs: key difference with traditional autoencoder



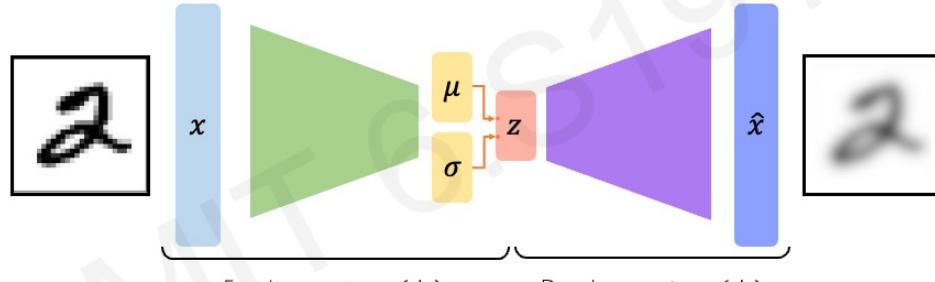
VAE optimization



Variational autoencoders are a probabilistic twist on autoencoders!

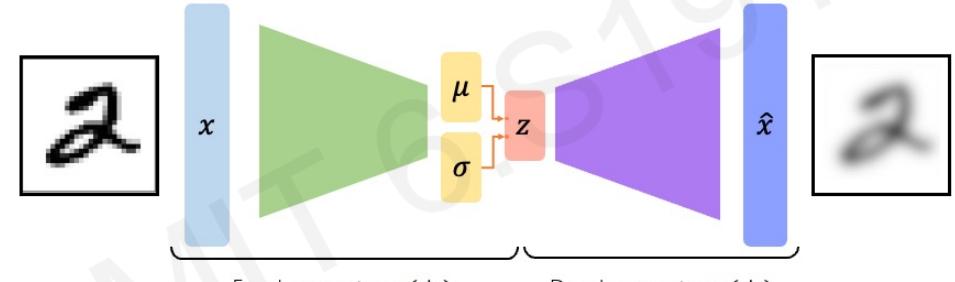
Sample from the mean and standard deviation to compute latent sample

VAE optimization



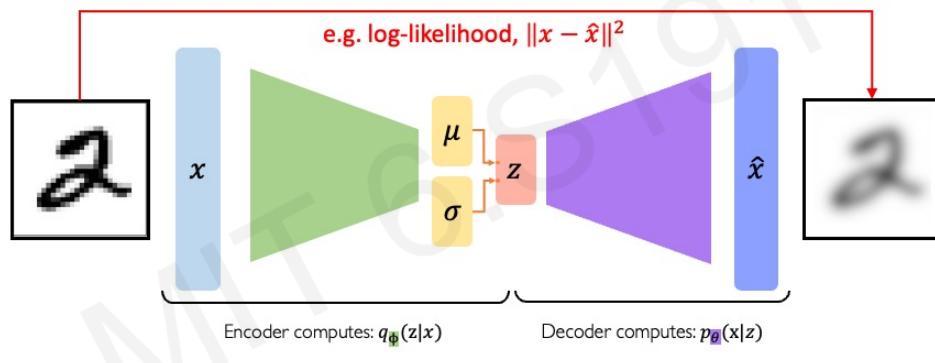
$$\mathcal{L}(\phi, \theta, x) = (\text{reconstruction loss}) + (\text{regularization term})$$

VAE optimization



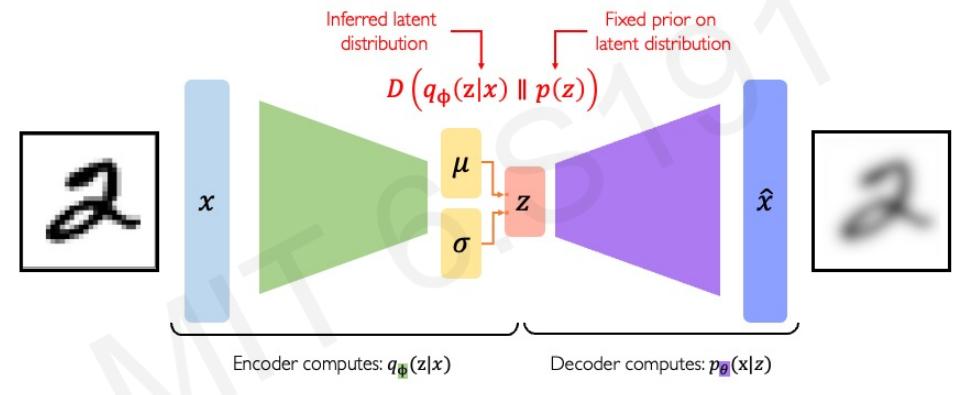
$$\mathcal{L}(\phi, \theta, x) = (\text{reconstruction loss}) + (\text{regularization term})$$

VAE optimization



$$\mathcal{L}(\phi, \theta, x) = (\text{reconstruction loss}) + (\text{regularization term})$$

VAE optimization

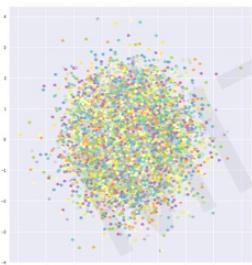


$$\mathcal{L}(\phi, \theta, x) = (\text{reconstruction loss}) + (\text{regularization term})$$

Priors on the latent distribution

$$D(q_\phi(z|x) \parallel p(z))$$

Inferred latent distribution Fixed prior on latent distribution



Common choice of prior – Normal Gaussian:

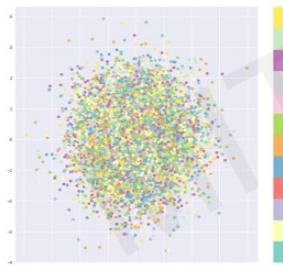
$$p(z) = \mathcal{N}(\mu = 0, \sigma^2 = 1)$$

- Encourages encodings to distribute encodings evenly around the center of the latent space
- Penalize the network when it tries to "cheat" by clustering points in specific regions (i.e., by memorizing the data)

Priors on the latent distribution

$$D(q_\phi(z|x) \parallel p(z)) = -\frac{1}{2} \sum_{j=0}^{k-1} (\sigma_j + \mu_j^2 - 1 - \log \sigma_j)$$

KL-divergence between the two distributions



Common choice of prior – Normal Gaussian:

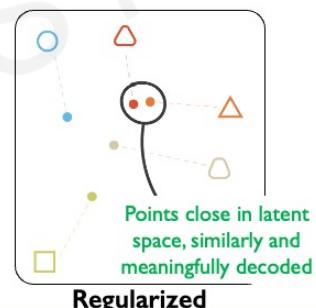
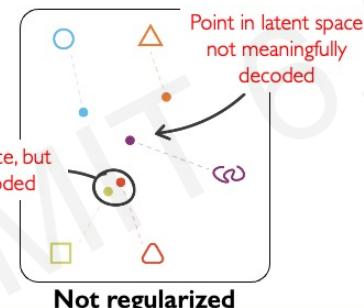
$$p(z) = \mathcal{N}(\mu = 0, \sigma^2 = 1)$$

- Encourages encodings to distribute encodings evenly around the center of the latent space
- Penalize the network when it tries to "cheat" by clustering points in specific regions (i.e., by memorizing the data)

Intuition on regularization and the Normal prior

What properties do we want to achieve from regularization? 🤔

1. **Continuity:** points that are close in latent space → similar content after decoding
2. **Completeness:** sampling from latent space → "meaningful" content after decoding



Intuition on regularization and the Normal prior

1. **Continuity:** points that are close in latent space → similar content after decoding
2. **Completeness:** sampling from latent space → "meaningful" content after decoding

Encoding as a distribution does not guarantee these properties!

Small variances → Pointed distributions

Different means → Discontinuities

Normal prior → continuity + completeness

Not regularized

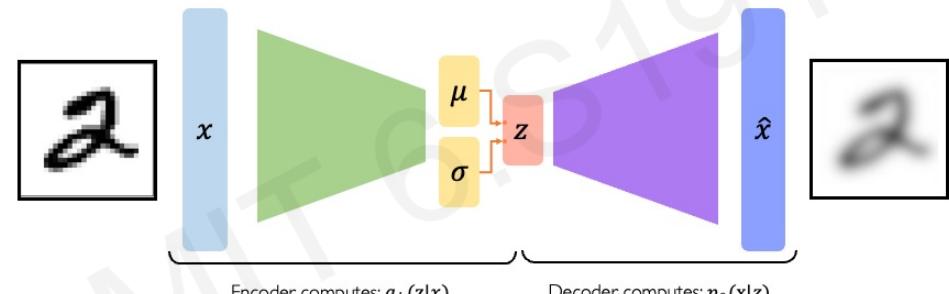
Regularized

Intuition on regularization and the Normal prior

1. **Continuity:** points that are close in latent space \rightarrow similar content after decoding
2. **Completeness:** sampling from latent space \rightarrow "meaningful" content after decoding



VAE computation graph

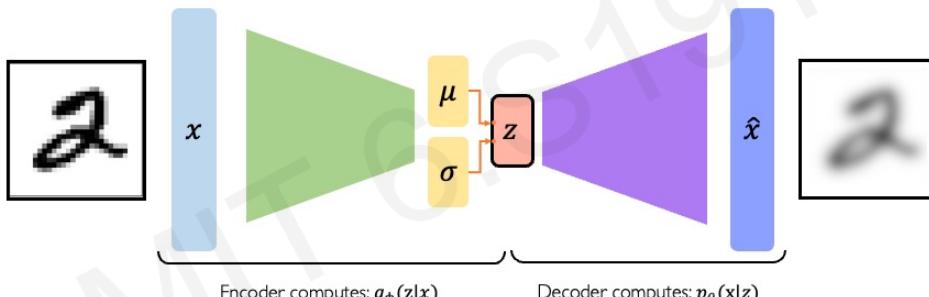


$$\mathcal{L}(\phi, \theta, x) = (\text{reconstruction loss}) + (\text{regularization term})$$

Regularization with Normal prior helps enforce **continuity & completeness** in the latent space.

VAE computation graph

Problem: We cannot backpropagate gradients through sampling layers!



$$\mathcal{L}(\phi, \theta, x) = (\text{reconstruction loss}) + (\text{regularization term})$$

Reparametrizing the sampling layer

Key Idea:

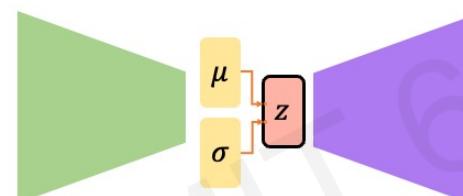
$$z \sim \mathcal{N}(\mu, \sigma^2)$$

Consider the sampled latent vector z as a sum of

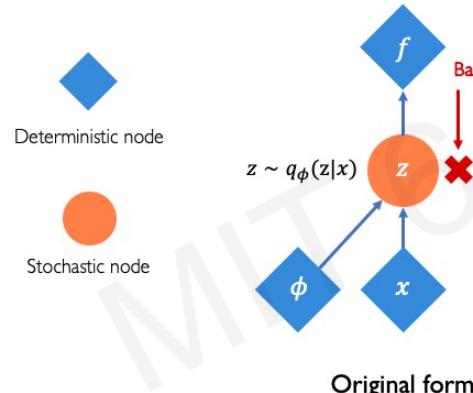
- a fixed μ vector,
- and fixed σ vector, scaled by random constants drawn from the prior distribution

$$\Rightarrow z = \mu + \sigma \odot \varepsilon$$

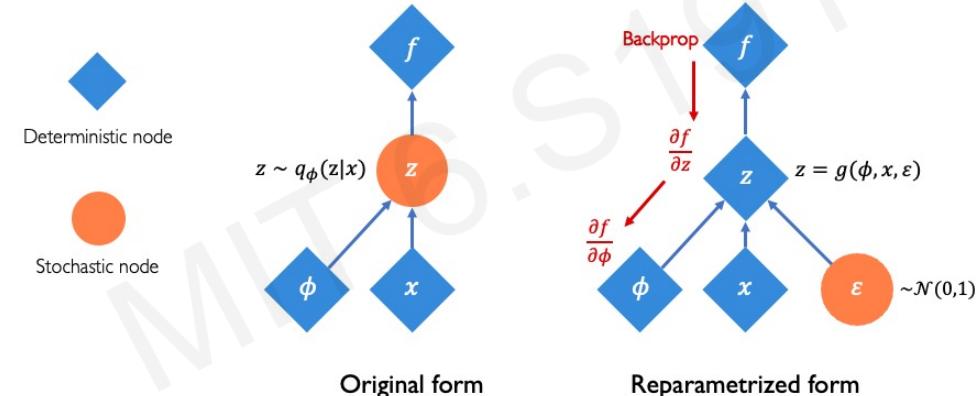
where $\varepsilon \sim \mathcal{N}(0,1)$



Reparametrizing the sampling layer

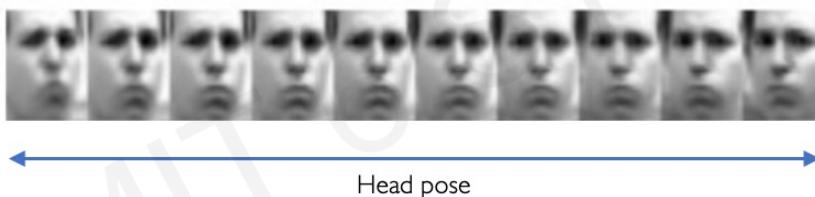


Reparametrizing the sampling layer



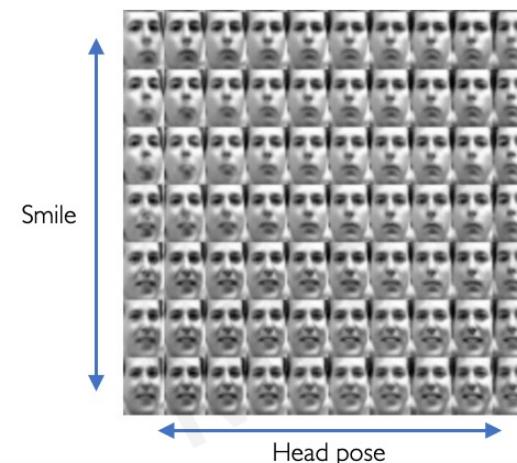
VAEs: Latent perturbation

Slowly increase or decrease a **single latent variable**
Keep all other variables fixed



Different dimensions of z encode **different interpretable latent features**

VAEs: Latent perturbation



Ideally, we want latent variables that are uncorrelated with each other

Enforce diagonal prior on the latent variables to encourage independence

Disentanglement

Latent space disentanglement with β -VAEs

Standard VAE loss:

$$\mathcal{L}(\theta, \phi; \mathbf{x}, \mathbf{z}, \beta) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z}))$$

Reconstruction term

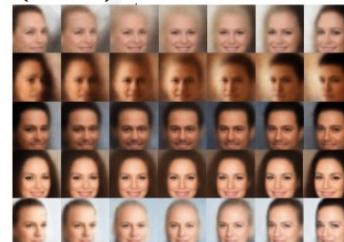
Regularization term

$\beta > 1$: constrain latent bottleneck, encourage efficient latent encoding \rightarrow disentanglement

Head rotation (azimuth)



Standard VAE ($\beta = 1$)



β -VAE ($\beta = 250$)

MIT Introduction to Deep Learning
introtodeeplearning.com @MITDeepLearning

Higgins+ ICLR 2017. 1/7/25

MIT Massachusetts Institute of Technology

Why latent variable models? Debiasing

Capable of uncovering **underlying latent variables** in a dataset



Homogeneous skin color, pose



Diverse skin color, pose, illumination

vs

How can we use latent distributions to create fair and representative datasets?



Software Lab!

MIT Introduction to Deep Learning
introtodeeplearning.com @MITDeepLearning

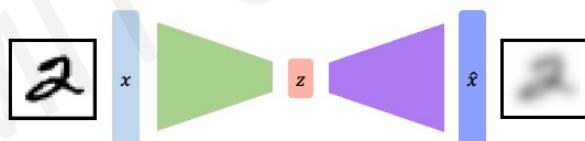
Amini/Soleimany+ AAAI/AIES 2019. 1/7/25

VAE summary

1. Compress representation of world to something we can use to learn

VAE summary

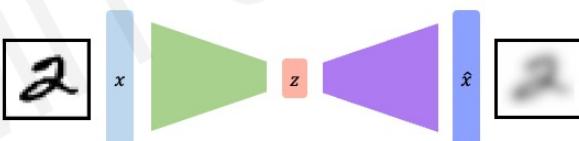
1. Compress representation of world to something we can use to learn
2. Reconstruction allows for unsupervised learning (no labels!)



MIT Introduction to Deep Learning
introtodeeplearning.com @MITDeepLearning

1/7/25

MIT Massachusetts Institute of Technology

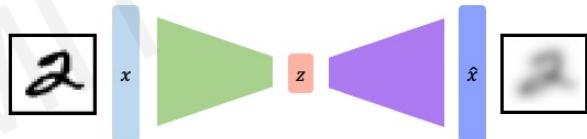


MIT Introduction to Deep Learning
introtodeeplearning.com @MITDeepLearning

1/7/25

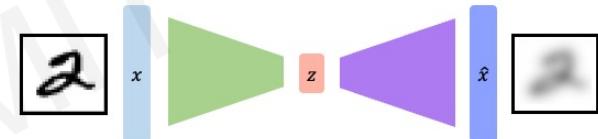
VAE summary

1. Compress representation of world to something we can use to learn
2. Reconstruction allows for unsupervised learning (no labels!)
3. Reparameterization trick to train end-to-end



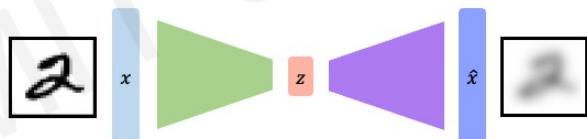
VAE summary

1. Compress representation of world to something we can use to learn
2. Reconstruction allows for unsupervised learning (no labels!)
3. Reparameterization trick to train end-to-end
4. Interpret hidden latent variables using perturbation



VAE summary

1. Compress representation of world to something we can use to learn
2. Reconstruction allows for unsupervised learning (no labels!)
3. Reparameterization trick to train end-to-end
4. Interpret hidden latent variables using perturbation
5. Generating new examples



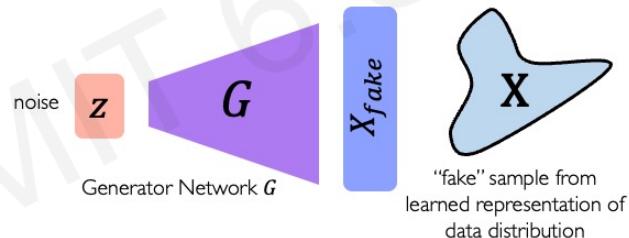
Generative Adversarial Networks (GANs)

What if we just want to sample?

Idea: don't explicitly model density, and instead just sample to generate new instances.

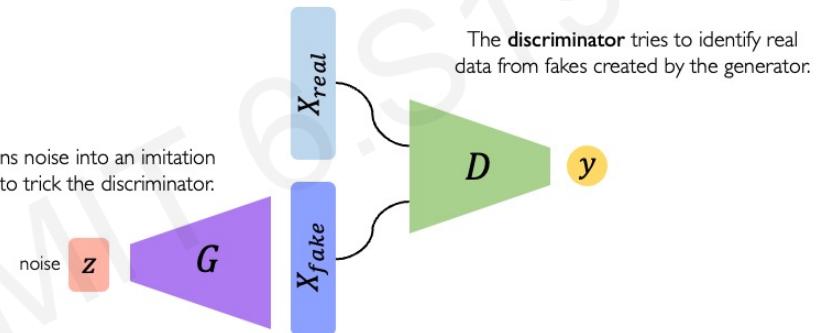
Problem: want to sample from complex distribution – can't do this directly!

Solution: sample from something simple (e.g., noise), learn a transformation to the data distribution.



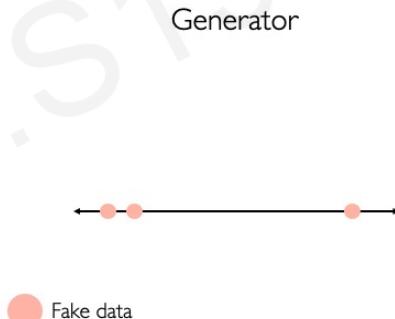
Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) are a way to make a generative model by having two neural networks compete with each other.



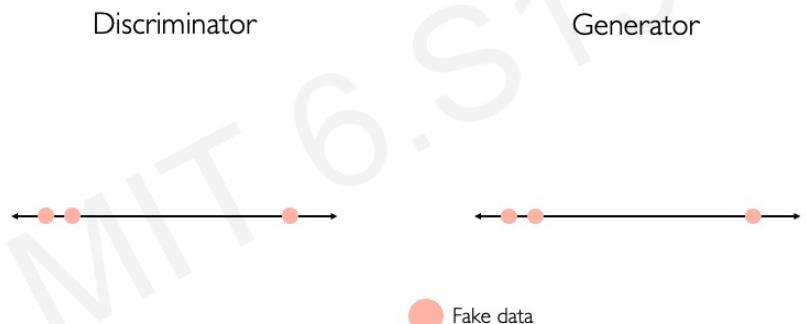
Intuition behind GANs

Generator starts from noise to try to create an imitation of the data.



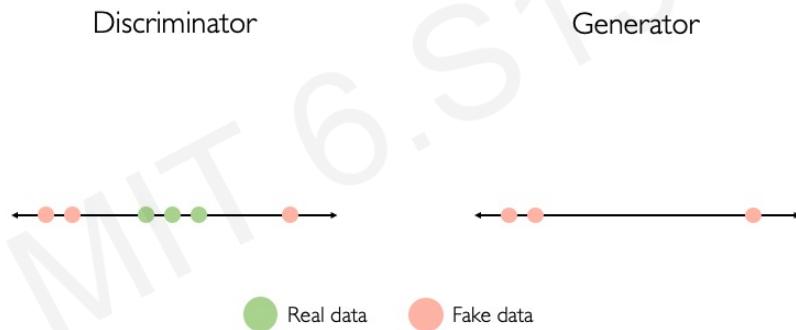
Intuition behind GANs

Discriminator looks at both real data and fake data created by the generator.



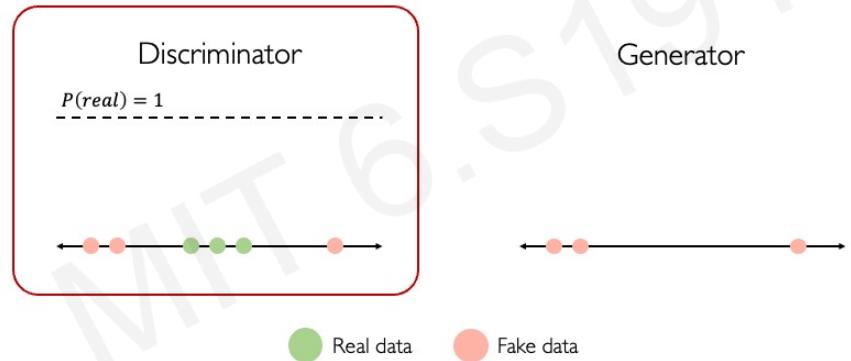
Intuition behind GANs

Discriminator looks at both real data and fake data created by the generator.



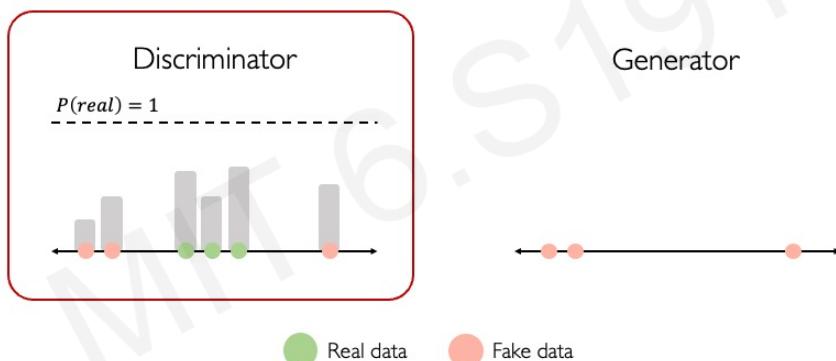
Intuition behind GANs

Discriminator tries to predict what's real and what's fake.



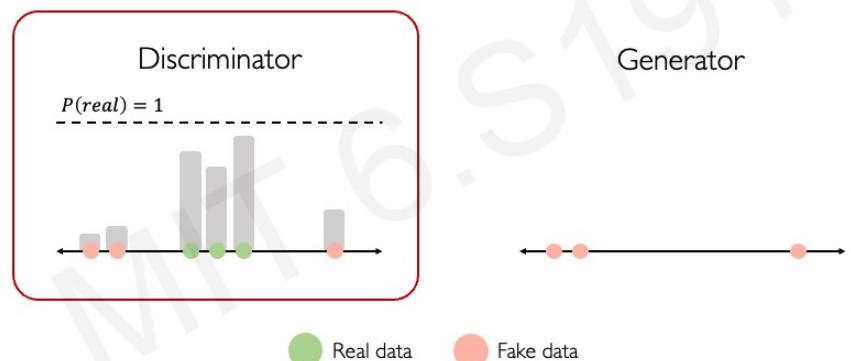
Intuition behind GANs

Discriminator tries to predict what's real and what's fake.



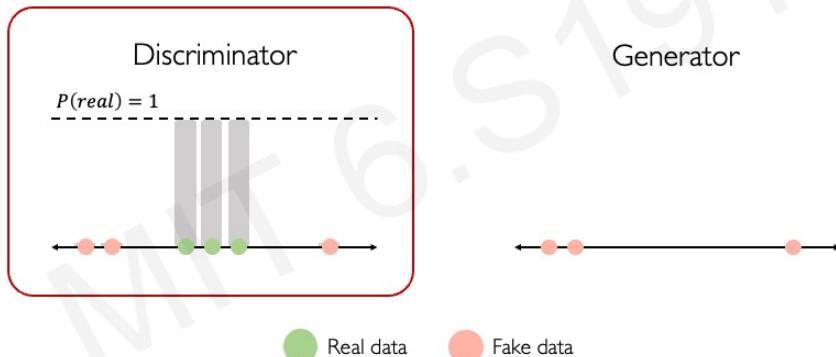
Intuition behind GANs

Discriminator tries to predict what's real and what's fake.



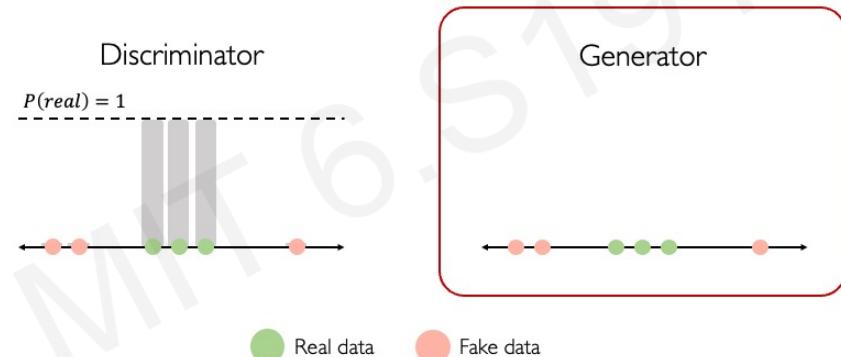
Intuition behind GANs

Discriminator tries to predict what's real and what's fake.



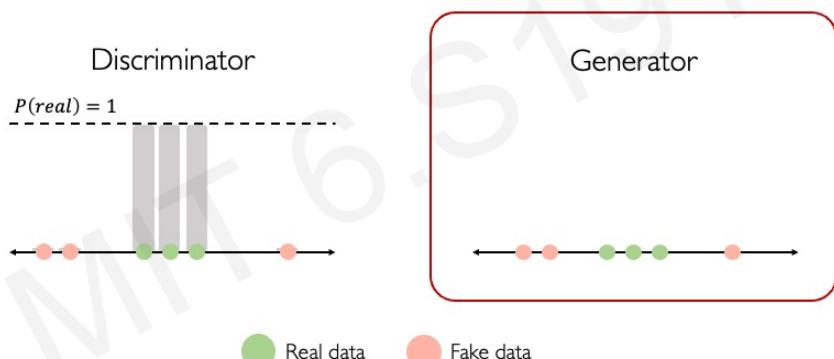
Intuition behind GANs

Generator tries to improve its imitation of the data.



Intuition behind GANs

Generator tries to improve its imitation of the data.



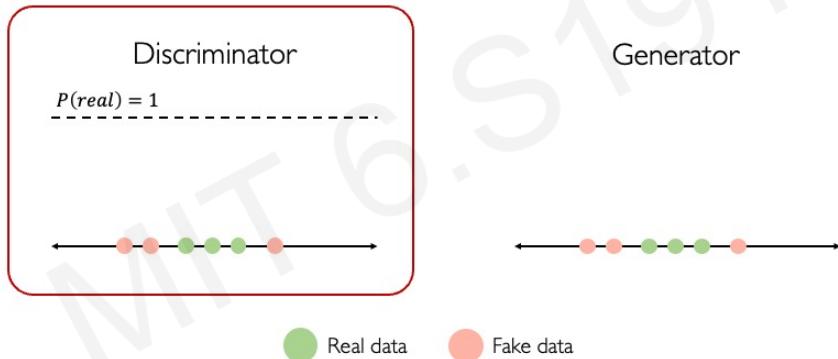
Intuition behind GANs

Generator tries to improve its imitation of the data.



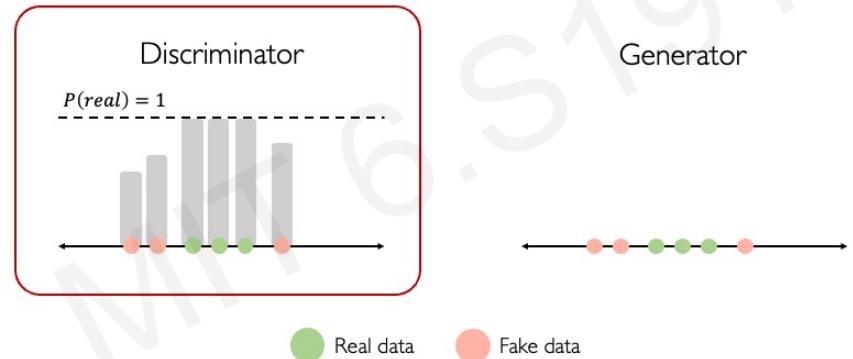
Intuition behind GANs

Discriminator tries to predict what's real and what's fake.



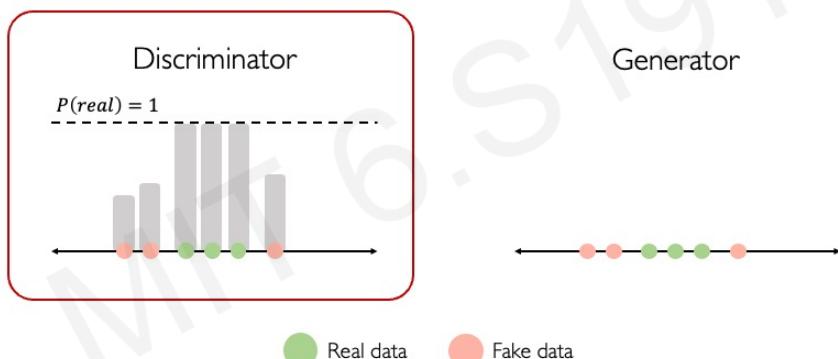
Intuition behind GANs

Discriminator tries to predict what's real and what's fake.



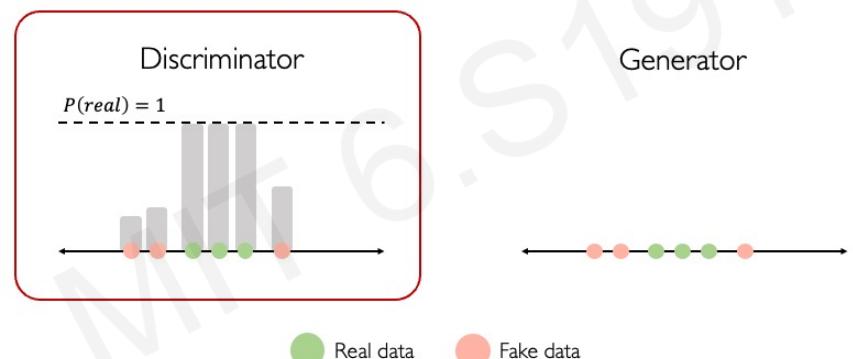
Intuition behind GANs

Discriminator tries to predict what's real and what's fake.



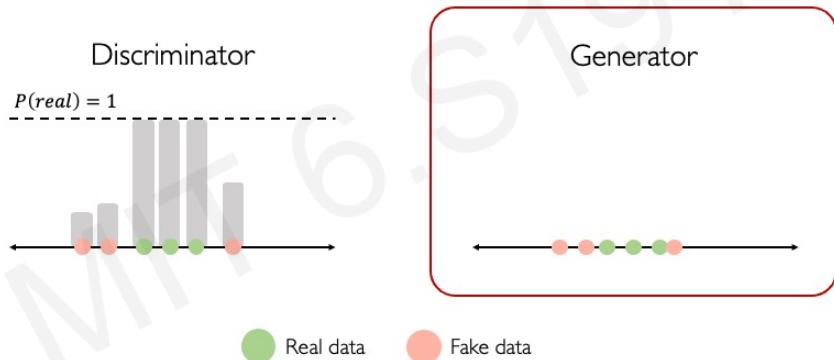
Intuition behind GANs

Discriminator tries to predict what's real and what's fake.



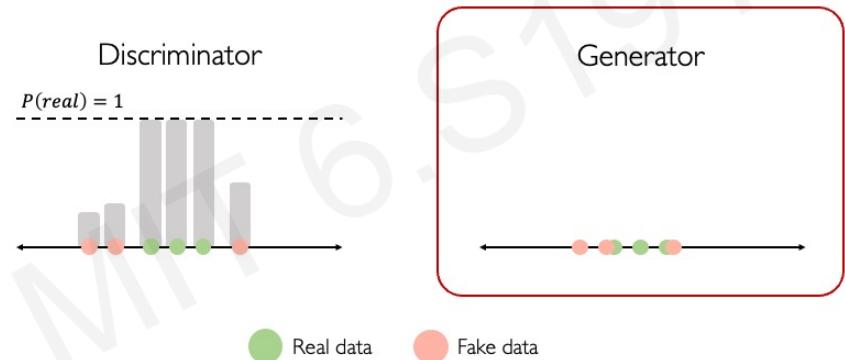
Intuition behind GANs

Generator tries to improve its imitation of the data.



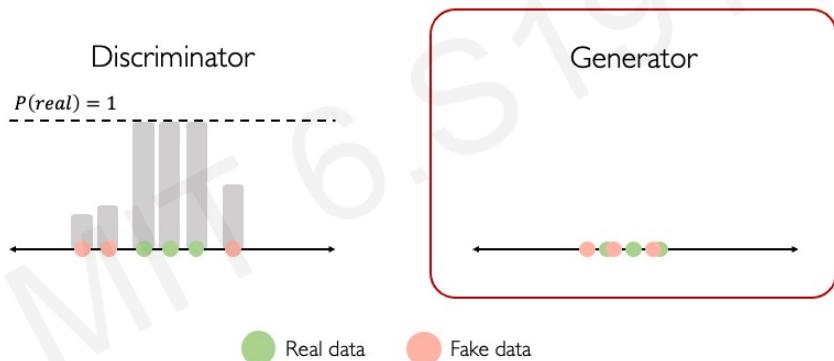
Intuition behind GANs

Generator tries to improve its imitation of the data.



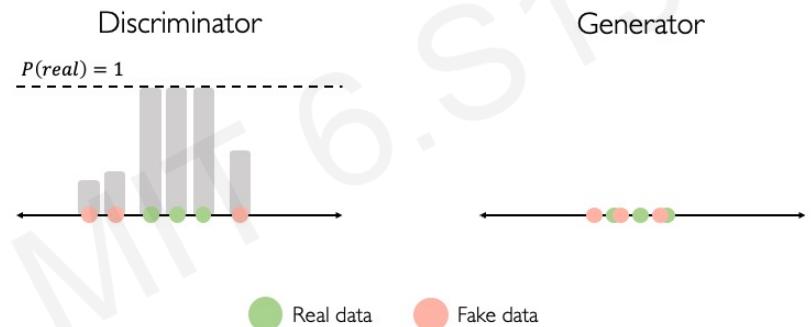
Intuition behind GANs

Generator tries to improve its imitation of the data.

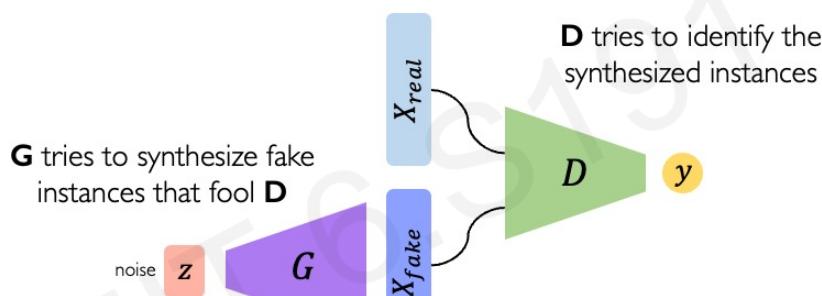


Intuition behind GANs

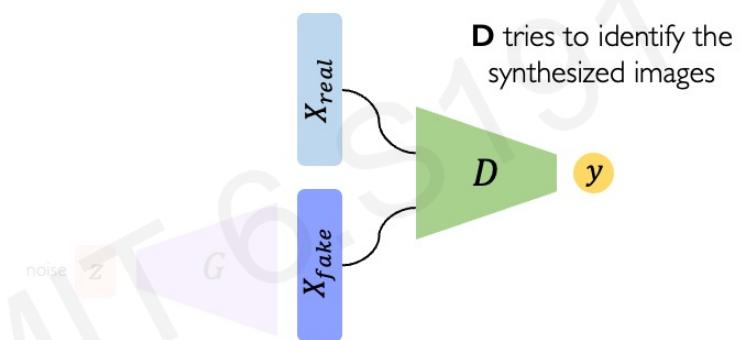
Discriminator tries to identify real data from fakes created by the generator.
Generator tries to create imitations of data to trick the discriminator.



Training GANs



Training GANs: loss function

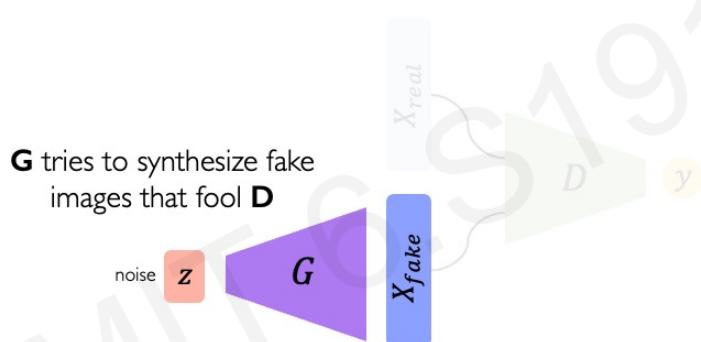


Training: adversarial objectives for **D** and **G**

Global optimum: **G** reproduces the true data distribution

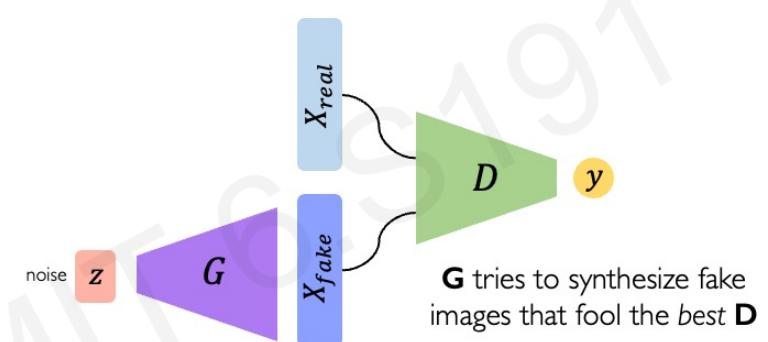
$$\arg \max_D \mathbb{E}_{\mathbf{z}, \mathbf{x}} [\underbrace{\log D(G(\mathbf{z}))}_{\text{Fake}} + \underbrace{\log (1 - D(\mathbf{x}))}_{\text{Real}}]$$

Training GANs: loss function



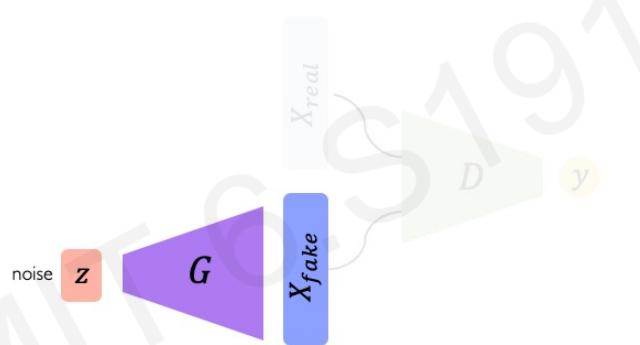
$$\arg \min_G \mathbb{E}_{\mathbf{z}, \mathbf{x}} [\log D(G(\mathbf{z})) + \log (1 - D(\mathbf{x}))]$$

Training GANs: loss function



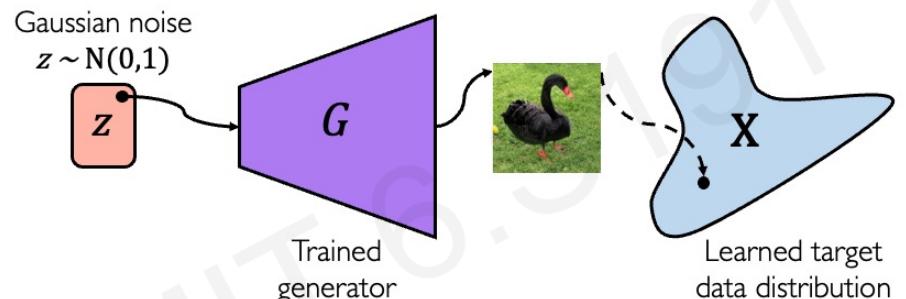
$$\arg \min_G \max_D \mathbb{E}_{\mathbf{z}, \mathbf{x}} [\log D(G(\mathbf{z})) + \log (1 - D(\mathbf{x}))]$$

Generating new data with GANs

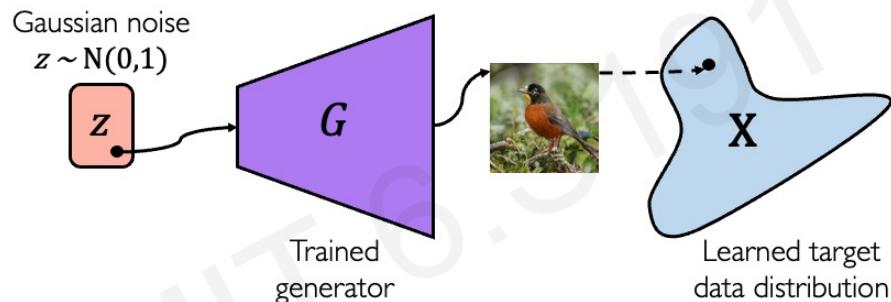


After training, use generator network to create **new data** that's never been seen before.

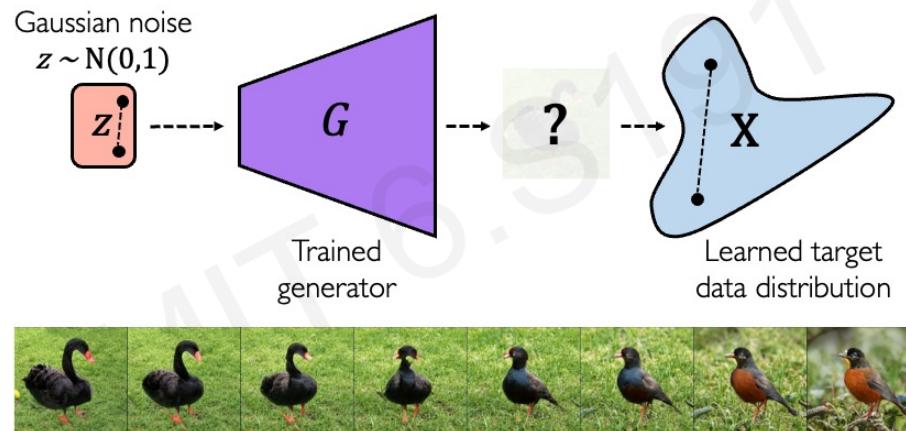
GANs are distribution transformers



GANs are distribution transformers

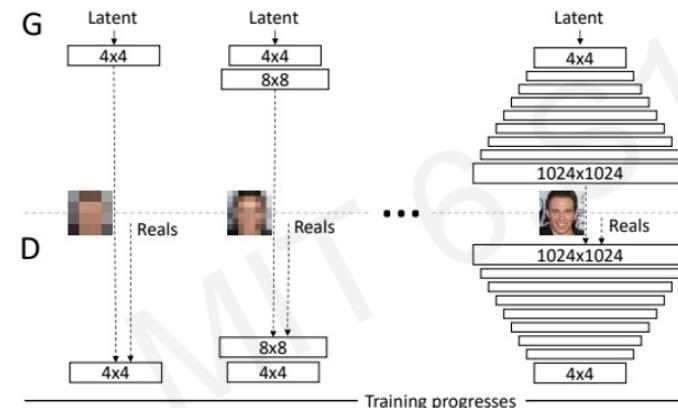


GANs are distribution transformers



GANs: Advances and Applications

Progressive growing of GANs



MIT Introduction to Deep Learning
introtodeeplearning.com @MITDeepLearning

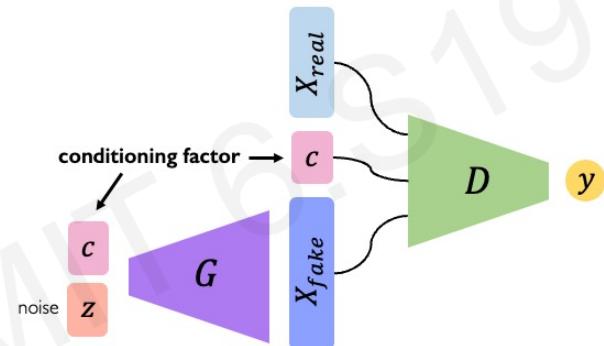
Karras+ ICLR 2018; Karras+ arXiv 2019. 1/7/25

Progressive growing of GANs: results

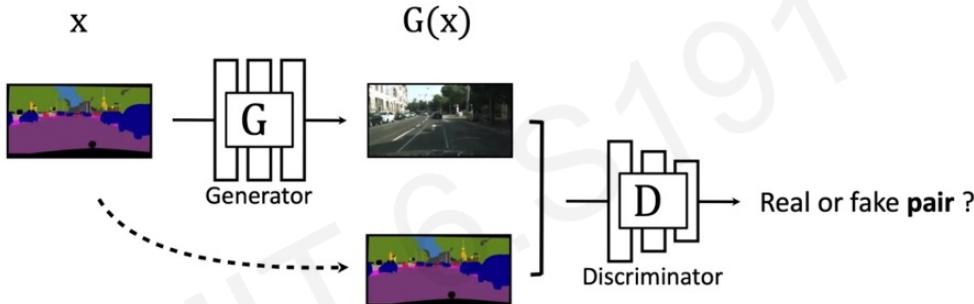


Conditional GANs

What if we want to control the nature of the output, by **conditioning** on a label?



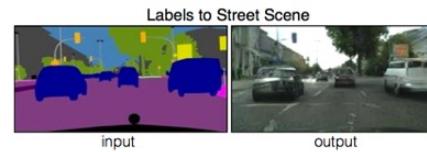
Conditional GANs and pix2pix: paired translation



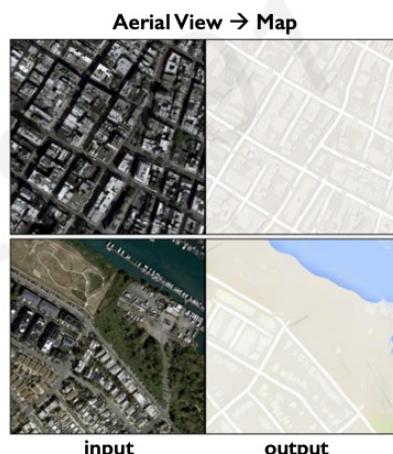
The discriminator, D , classifies between fake and real **pairs**.

The generator, G , learns to fool the discriminator.

Applications of paired translation

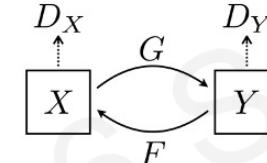


Paired translation: results



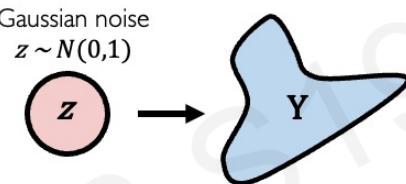
CycleGAN: domain transformation

CycleGAN learns transformations across domains with unpaired data.



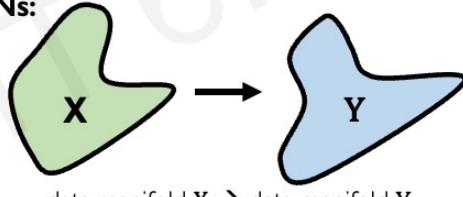
Distribution transformations

GANs: Gaussian noise



Gaussian noise → target data manifold

CycleGANs:



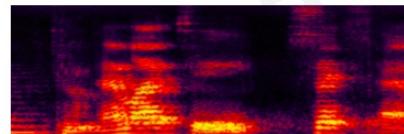
data manifold X → data manifold Y

CycleGAN: transforming speech

Audio waveform (A)



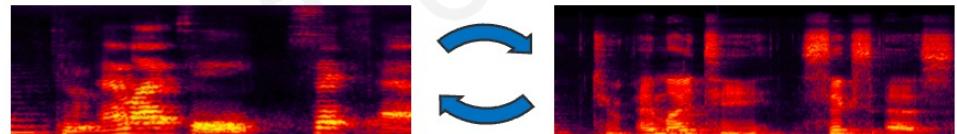
Spectrogram image (A)



Audio waveform (B)



Spectrogram image (B)



Original (Amini)



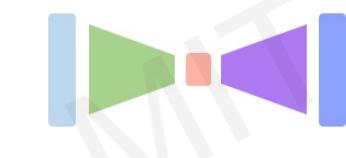
Synthesized (Obama)



Deep Generative Modeling: Summary

Autoencoders and Variational Autoencoders (VAEs)

Learn lower-dimensional **latent space** and **sample** to generate input reconstructions



Generative Adversarial Networks (GANs)

Competing **generator** and **discriminator** networks

