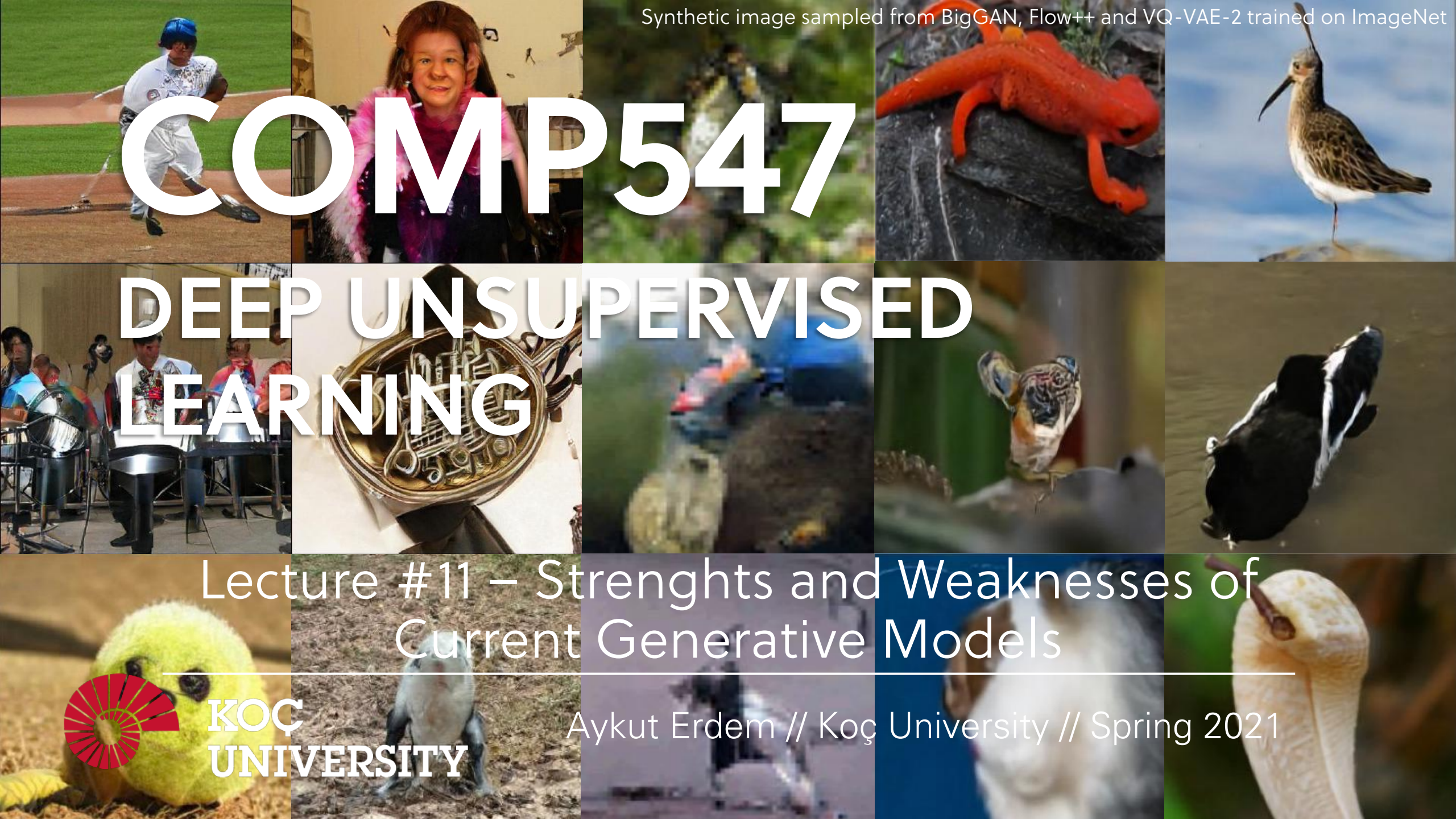


COMP547

DEEP UNSUPERVISED LEARNING

Lecture #11 – Strenghts and Weaknesses of Current Generative Models

KOÇ UNIVERSITY

Aykut Erdem // Koç University // Spring 2021

Previously on COMP547

- Motivation
- REINFORCE, Gumbel-Softmax, Straight-through estimator
- Sparse Coding
- Vector Quantization VAE (VQ-VAE), VQ-VAE-2, VQGAN
- Discrete Flows
- GANs for Text: SeqGAN, MaskGAN, ScratchGAN

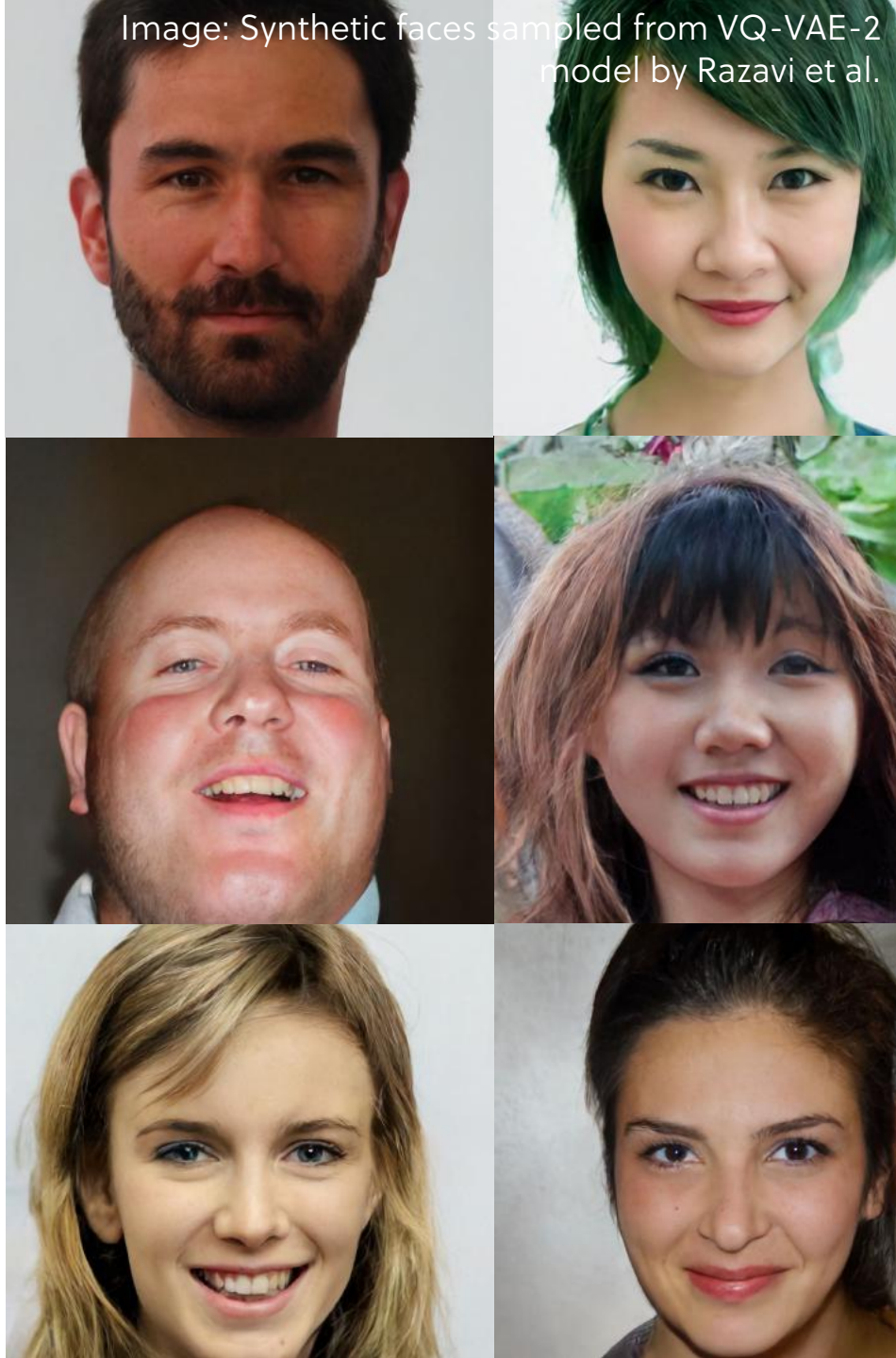


Image: Synthetic faces samoled from VQ-VAE-2 model by Razavi et al.

Lecture overview

- Autoregressive models
- Flow models
- Latent Variable models
- Implicit models

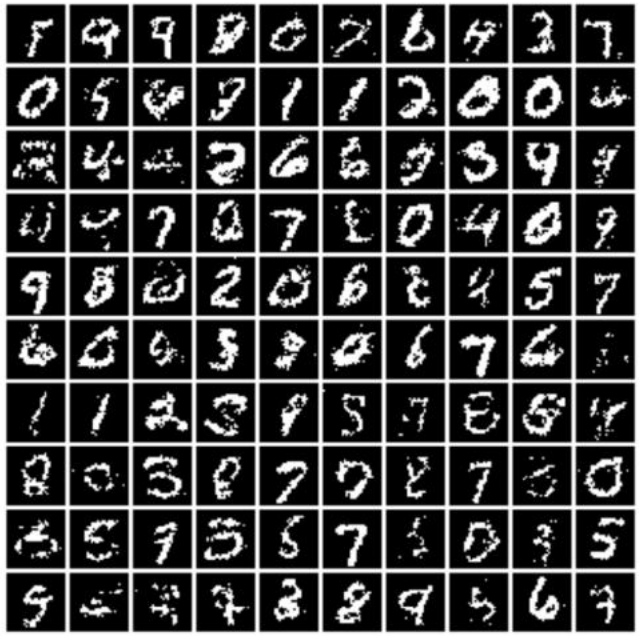
Disclaimer: Much of the material and slides for this lecture were borrowed from

—Pieter Abbeel, Peter Chen, Jonathan Ho, Aravind Srinivas' Berkeley CS294-158 class

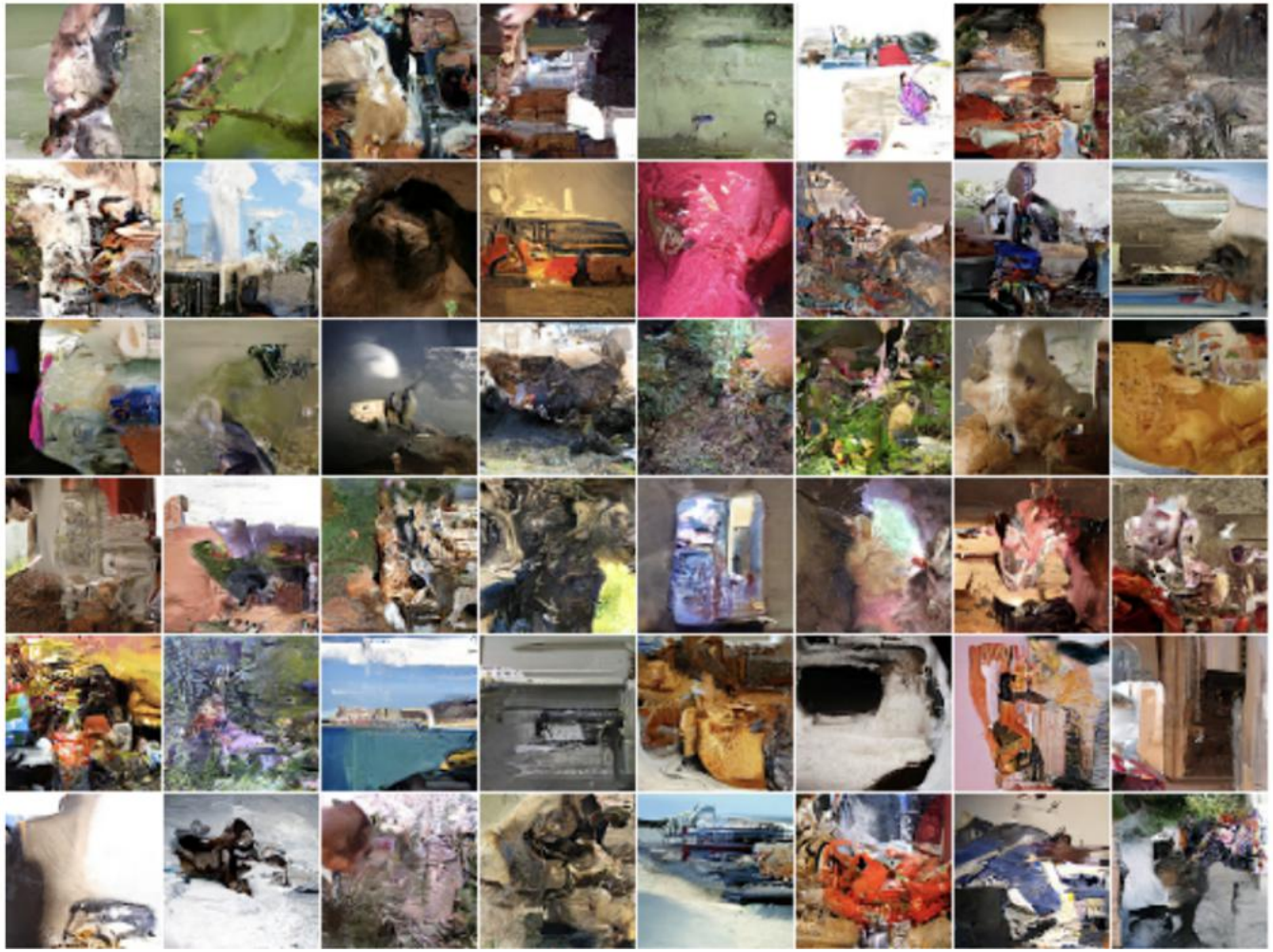
Lecture overview

- Autoregressive models
  - PixelRNN, PixelCNN, PixelCNN++, PixelSNAIL
- Flow models
- Latent Variable models
- Implicit models

Autoregressive Models

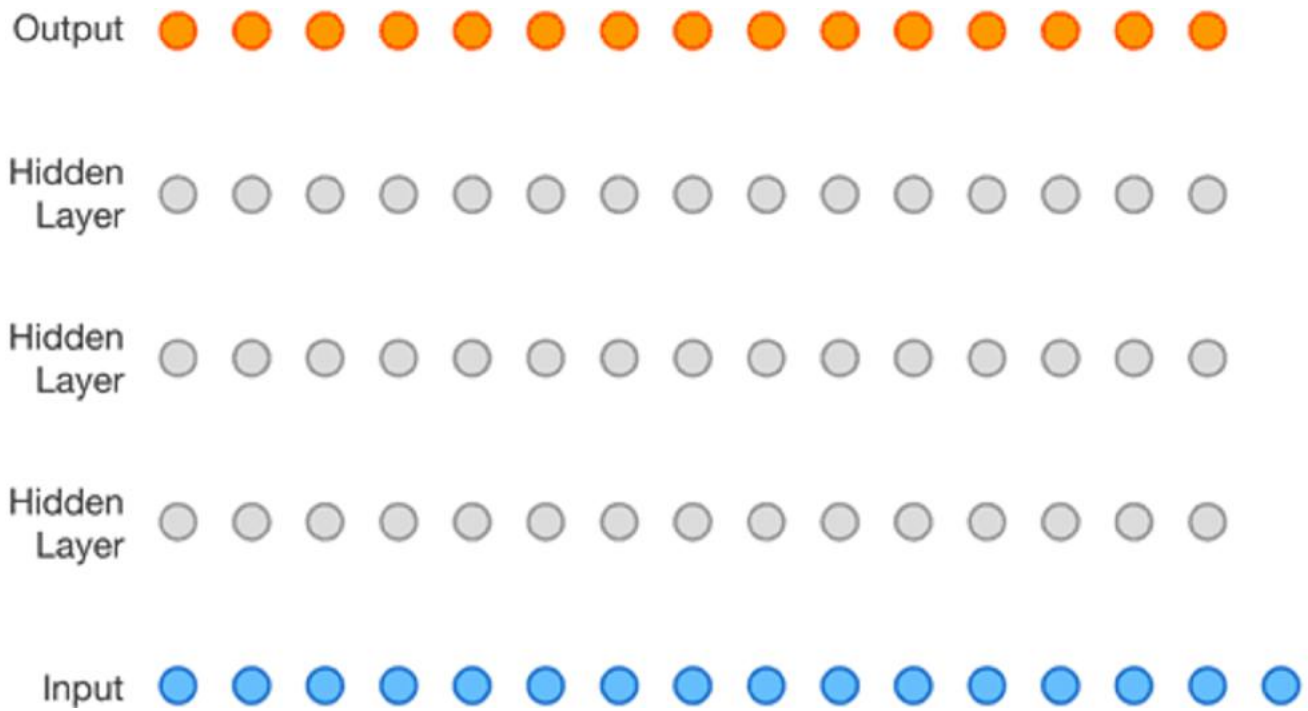


MADE (2015)



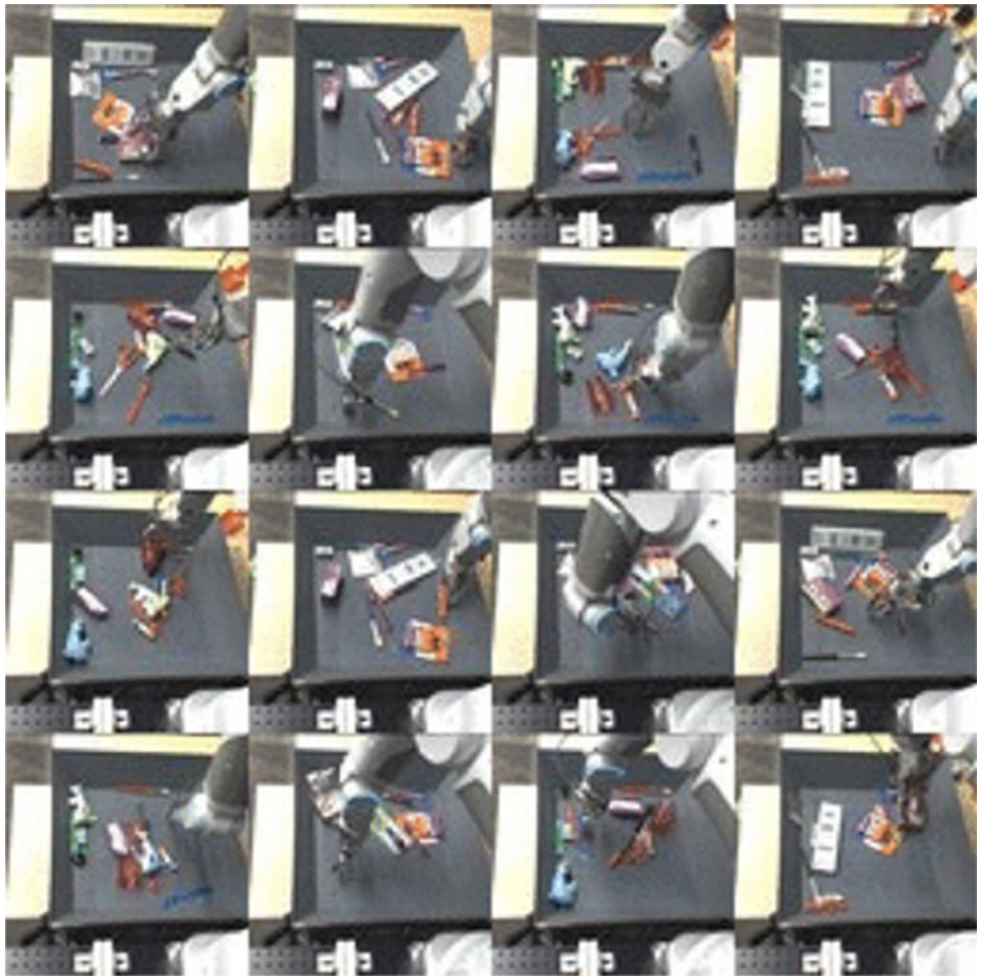
PixelRNN/CNN (2016)

Autoregressive Models



WaveNet

Autoregressive Models



Video Pixel Networks

Autoregressive Models

Subscale Pixel Networks

Hierarchical Autoregressive Image Models with Auxiliary Decoders

Autoregressive Models: History of LMs

SLP book, 2000 (Shannon, 1951), 3-gram

They also point to ninety nine point six billion dollars from two hundred four oh six three percent of the rates of interest stores as Mexico and Brazil on market conditions

Sutskever et al, 2011, RNNs

The meaning of life is the tradition of the ancient human reproduction: it is less favorable to the good boy for when to remove her bigger

Jozefowicz et al, 2016, BIG LSTMs

With even more new technologies coming onto the market quickly during the past three years , an increasing number of companies now must tackle the ever-changing and ever-changing environmental challenges online .

Liu et al, 2018, Transformer

==wings over kansas

==wings over kansas is a 2010 dhamma feature film written and directed by brian ig ariyoshi . it premiered on march 17, 2010 the film tells the story of three americans who bravely achieved a victory without expected daknfi .

==Wings Over Kansas Plot

the story begins with the faltering success of egypt 's hungry dakfunctionality when he loses his lives around the time when the embarked [...]

Radford et al, 2019, BIG Transformer

In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Perez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Perez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Perez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Perez.

Perez and his friends were astonished to see the unicorn herd. [...]

Autoregressive Models: OpenAI GPT-2

In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.

Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.

While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic."

Dr. Pérez believes that the unicorns may have originated in Argentina, where the animals were believed to be descendants of a lost race of people who lived there before the arrival of humans in those parts of South America.

While their origins are still unclear, some believe that perhaps the creatures were created when a human and a unicorn met each other in a time before human civilization. According to Pérez, "In South America, such incidents seem to be quite common."

However, Pérez also pointed out that it is likely that the only way of knowing for sure if unicorns are indeed the descendants of a lost alien race is through DNA. "But they seem to be able to communicate in English quite well, which I believe is a sign of evolution, or at least a change in social organization," said the scientist.

Autoregressive Models: OpenAI GPT-3

Context →	The City BY C. P. CAVAFY TRANSLATED BY EDMUND KEELEY [Poem text omitted] SOME TREES John Ashbery [Poem text omitted] Shadows on the Way Wallace Stevens

Nobody will come to this place. It is a road that leads nowhere.
The solitude is deep. The mountains are high.
But they are desolate, and they turn the traveler's face
Towards the North. All the sounds of the world are far away.
When the wind rises above the trees,
The boughs bow to the ground.
Even the birds that inhabit the tangle of weeds
That is the roadside cover, are silent. One listens,
But hears no roar of the forest. One is alone.
One will be taken.
One will be taken.
There is no utterance, there is no conversation,
But one is uneasy all the same....
There is a thin blue mist,
A darkness rising like smoke,
And within that darkness
A possession of the heart.
One will be taken.... It was here, and it will be here again-
Here, under this sky empty and full of light.

Completion from a context suggesting the model compose a poem in the style of Wallace Stevens with the title 'Shadows on the Way'.

Autoregressive Models: OpenAI GPT-3

Context →	Helsinki is the capital and largest city of Finland. It is in the region of Uusimaa, in southern Finland, on the shore of the Gulf of Finland. Helsinki has a population of , an urban population of , and a metropolitan population of over 1.4 million, making it the most populous municipality and urban area in Finland. Helsinki is some north of Tallinn, Estonia, east of Stockholm, Sweden, and west of Saint Petersburg, Russia. Helsinki has close historical connections with these three cities. The Helsinki metropolitan area includes the urban core of Helsinki, Espoo, Vantaa, Kauniainen, and surrounding commuter towns. It is the world's northernmost metro area of over one million people, and the city is the northernmost capital of an EU member state. The Helsinki metropolitan area is the third largest metropolitan area in the Nordic countries after Stockholm and Copenhagen, and the City of Helsinki is the third largest after Stockholm and Oslo. Helsinki is Finland's major political, educational, financial, cultural, and research center as well as one of northern Europe's major cities. Approximately 75% of foreign companies that operate in Finland have settled in the Helsinki region. The nearby municipality of Vantaa is the location of Helsinki Airport, with frequent service to various destinations in Europe and Asia. Q: what is the most populous municipality in Finland? A: Helsinki Q: how many people live there? A: 1.4 million in the metropolitan area Q: what percent of the foreign companies that operate in Finland are in Helsinki? A: 75% Q: what towns are a part of the metropolitan area? A:
Target Completion →	Helsinki, Espoo, Vantaa, Kauniainen, and surrounding commuter towns

Formatted dataset example for CoQA

Autoregressive Models

Huge advances due to:

- Larger batch sizes
- More hidden units
- More layers
- Clever ways to condition on auxiliary variables
- Preprocessing
- Computer power
- Several days / weeks of training

- Fewer assumptions
  - Masked / Causal Convolutions
  - Dilated Convolutions
  - Transformers
- Architectural advances
- Loss functions
  - Relying heavily on well-behaved cross-entropy loss

Autoregressive Models

Huge advances due to:

- Larger batch sizes
- More hidden units
- More layers
- Clever ways to condition on auxiliary variables
- Preprocessing
- Computer power
- Several days / weeks of training
- Fewer assumptions
  - Masked / Causal Convolutions
  - Dilated Convolutions
  - Transformers
- Architectural advances
- Loss functions
  - Relying heavily on well-behaved cross-entropy loss

Autoregressive Models

Huge advances due to:

- Larger batch sizes
- More hidden units
- More layers
- Clever ways to condition on auxiliary variables
- Preprocessing
- Computer power
- Several days / weeks of training
- Fewer assumptions
  - Masked / Causal Convolutions
  - Dilated Convolutions
  - Transformers
- Architectural advances
- Loss functions
  - Relying heavily on well-behaved cross-entropy loss

Autoregressive Models

Huge advances due to:

- Larger batch sizes
- More hidden units
- More layers
- Clever ways to condition on auxiliary variables
- Preprocessing
- Computer power
- Several days / weeks of training
- Fewer assumptions
  - Masked / Causal Convolutions
  - Dilated Convolutions
  - Transformers
- Architectural advances
- Loss functions
  - Relying heavily on well-behaved cross-entropy loss

Autoregressive Models

Huge advances due to:

- Larger batch sizes
- More hidden units
- More layers
- Clever ways to condition on auxiliary variables
- Preprocessing
- Computer power
- Several days / weeks of training

- Fewer assumptions
  - Masked / Causal Convolutions
  - Dilated Convolutions
  - Transformers
- Architectural advances
- Loss functions
  - Relying heavily on well-behaved cross-entropy loss

Autoregressive Models

Huge advances due to:

- Larger batch sizes
- More hidden units
- More layers
- Clever ways to condition on auxiliary variables
- Preprocessing
- Computer power
- Several days / weeks of training
- Fewer assumptions
  - Masked / Causal Convolutions
  - Dilated Convolutions
  - Transformers
- Architectural advances
- Loss functions
  - Relying heavily on well-behaved cross-entropy loss

Autoregressive Models

Huge advances due to:

- Larger batch sizes
- More hidden units
- More layers
- Clever ways to condition on auxiliary variables
- Preprocessing
- Computer power
- Several days / weeks of training

- Fewer assumptions
  - Masked / Causal Convolutions
  - Dilated Convolutions
  - Transformers
- Architectural advances
- Loss functions
  - Relying heavily on well-behaved cross-entropy loss

Autoregressive Models

Huge advances due to:

- Larger batch sizes
- More hidden units
- More layers
- Clever ways to condition on auxiliary variables
- Preprocessing
- Computer power
- Several days / weeks of training

- Fewer assumptions
- Architectural advances
  - Masked / Causal Convolutions
  - Dilated Convolutions
  - Transformers
- Loss functions
  - Relying heavily on well-behaved cross-entropy loss

Autoregressive Models

Huge advances due to:

- Larger batch sizes
- More hidden units
- More layers
- Clever ways to condition on auxiliary variables
- Preprocessing
- Computer power
- Several days / weeks of training

- Fewer assumptions
- Architectural advances
  - Masked / Causal Convolutions
  - Dilated Convolutions
  - Transformers
- Loss functions
  - Relying heavily on well-behaved cross-entropy loss

Autoregressive Models

Huge advances due to:

- Larger batch sizes
- More hidden units
- More layers
- Clever ways to condition on auxiliary variables
- Preprocessing
- Computer power
- Several days / weeks of training

- Fewer assumptions
- Architectural advances
  - Masked / Causal Convolutions
  - Dilated Convolutions
  - Transformers
- Loss functions
  - Relying heavily on well-behaved cross-entropy loss

Autoregressive Models

Huge advances due to:

- Larger batch sizes
- More hidden units
- More layers
- Clever ways to condition on auxiliary variables
- Preprocessing
- Computer power
- Several days / weeks of training

- Fewer assumptions
- Architectural advances
  - Masked / Causal Convolutions
  - Dilated Convolutions
  - Transformers
- Loss functions
  - Relying heavily on well-behaved cross-entropy loss

Autoregressive Models: Future

Still only scratching the surface of what's possible.

- Advances with model-parallelism to come
- Trillion parameter language models trained on all the Internet's text (ex Google Books / Kindle / HackerNews / Reddit / Podcast transcripts, so on). Could compress the Internet's text.
- Same model for both text and pixels (image / video). Share self-attention blocks and compress both of Wikipedia and Youtube / Instagram. Only separate blocks of encoders and decoders.
- Fast sampling with better low-level core engineering - new kernels with sparsity and efficiency for the bottleneck ops. Ex: WaveRNN instead of Parallel Wavenet.
- Hybrid models with weaker autoregressive structure but trained on a larger scale (Ex: Revisiting architectures like Parallel PixelCNN that can provide a good tradeoff between autoregressive structure and sampling time with more independence assumptions.
- New architecture design choices such as self-attention which introduce inductive biases that leverage a lot of computation per parameter introduced.

Autoregressive Models: Future

Still only scratching the surface of what's possible.

- Advances with model-parallelism to come
- Trillion parameter language models trained on all the Internet's text (ex Google Books / Kindle / HackerNews / Reddit / Podcast transcripts, so on). Could compress the Internet's text.
- Same model for both text and pixels (image / video). Share self-attention blocks and compress both of Wikipedia and Youtube / Instagram. Only separate blocks of encoders and decoders.
- Fast sampling with better low-level core engineering - new kernels with sparsity and efficiency for the bottleneck ops. Ex: WaveRNN instead of Parallel Wavenet.
- Hybrid models with weaker autoregressive structure but trained on a larger scale (Ex: Revisiting architectures like Parallel PixelCNN that can provide a good tradeoff between autoregressive structure and sampling time with more independence assumptions.
- New architecture design choices such as self-attention which introduce inductive biases that leverage a lot of computation per parameter introduced.

Autoregressive Models: Future

Still only scratching the surface of what's possible.

- Advances with model-parallelism to come
- Trillion parameter language models trained on all the Internet's text (ex Google Books / Kindle / HackerNews / Reddit / Podcast transcripts, so on). Could compress the Internet's text.
- Same model for both text and pixels (image / video). Share self-attention blocks and compress both of Wikipedia and Youtube / Instagram. Only separate blocks of encoders and decoders.
- Fast sampling with better low-level core engineering - new kernels with sparsity and efficiency for the bottleneck ops. Ex: WaveRNN instead of Parallel Wavenet.
- Hybrid models with weaker autoregressive structure but trained on a larger scale (Ex: Revisiting architectures like Parallel PixelCNN that can provide a good tradeoff between autoregressive structure and sampling time with more independence assumptions.
- New architecture design choices such as self-attention which introduce inductive biases that leverage a lot of computation per parameter introduced.

Autoregressive Models: Future

Still only scratching the surface of what's possible.

- Advances with model-parallelism to come
- Trillion parameter language models trained on all the Internet's text (ex Google Books / Kindle / HackerNews / Reddit / Podcast transcripts, so on). Could compress the Internet's text.
- Same model for both text and pixels (image / video). Share self-attention blocks and compress both of Wikipedia and Youtube / Instagram. Only separate blocks of encoders and decoders.
- Fast sampling with better low-level core engineering - new kernels with sparsity and efficiency for the bottleneck ops. Ex: WaveRNN instead of Parallel Wavenet.
- Hybrid models with weaker autoregressive structure but trained on a larger scale (Ex: Revisiting architectures like Parallel PixelCNN that can provide a good tradeoff between autoregressive structure and sampling time with more independence assumptions.
- New architecture design choices such as self-attention which introduce inductive biases that leverage a lot of computation per parameter introduced.

Autoregressive Models: Future

Still only scratching the surface of what's possible.

- Advances with model-parallelism to come
- Trillion parameter language models trained on all the Internet's text (ex Google Books / Kindle / HackerNews / Reddit / Podcast transcripts, so on). Could compress the Internet's text.
- Same model for both text and pixels (image / video). Share self-attention blocks and compress both of Wikipedia and Youtube / Instagram. Only separate blocks of encoders and decoders.
- Fast sampling with better low-level core engineering - new kernels with sparsity and efficiency for the bottleneck ops. Ex: WaveRNN instead of Parallel Wavenet.
- Hybrid models with weaker autoregressive structure but trained on a larger scale (Ex: Revisiting architectures like Parallel PixelCNN that can provide a good tradeoff between autoregressive structure and sampling time with more independence assumptions.
- New architecture design choices such as self-attention which introduce inductive biases that leverage a lot of computation per parameter introduced.

Autoregressive Models: Future

Still only scratching the surface of what's possible.

- Advances with model-parallelism to come
- Trillion parameter language models trained on all the Internet's text (ex Google Books / Kindle / HackerNews / Reddit / Podcast transcripts, so on). Could compress the Internet's text.
- Same model for both text and pixels (image / video). Share self-attention blocks and compress both of Wikipedia and Youtube / Instagram. Only separate blocks of encoders and decoders.
- Fast sampling with better low-level core engineering - new kernels with sparsity and efficiency for the bottleneck ops. Ex: WaveRNN instead of Parallel Wavenet.
- Hybrid models with weaker autoregressive structure but trained on a larger scale (Ex: Revisiting architectures like Parallel PixelCNN that can provide a good tradeoff between autoregressive structure and sampling time with more independence assumptions.
- New architecture design choices such as self-attention which introduce inductive biases that leverage a lot of computation per parameter introduced.

Autoregressive Models: Future

Still only scratching the surface of what's possible.

- Advances with model-parallelism to come
- Trillion parameter language models trained on all the Internet's text (ex Google Books / Kindle / HackerNews / Reddit / Podcast transcripts, so on). Could compress the Internet's text.
- Same model for both text and pixels (image / video). Share self-attention blocks and compress both of Wikipedia and Youtube / Instagram. Only separate blocks of encoders and decoders.
- Fast sampling with better low-level core engineering - new kernels with sparsity and efficiency for the bottleneck ops. Ex: WaveRNN instead of Parallel Wavenet.
- Hybrid models with weaker autoregressive structure but trained on a larger scale (Ex: Revisiting architectures like Parallel PixelCNN that can provide a good tradeoff between autoregressive structure and sampling time with more independence assumptions.
- New architecture design choices such as self-attention which introduce inductive biases that leverage a lot of computation per parameter introduced.

Autoregressive Models: Future

Still only scratching the surface of what's possible.

- Advances with model-parallelism to come
- Trillion parameter language models trained on all the Internet's text (ex Google Books / Kindle / HackerNews / Reddit / Podcast transcripts, so on). Could compress the Internet's text.
- Same model for both text and pixels (image / video). Share self-attention blocks and compress both of Wikipedia and Youtube / Instagram. Only separate blocks of encoders and decoders.
- Fast sampling with better low-level core engineering - new kernels with sparsity and efficiency for the bottleneck ops. Ex: WaveRNN instead of Parallel Wavenet.
- Hybrid models with weaker autoregressive structure but trained on a larger scale (Ex: Revisiting architectures like Parallel PixelCNN that can provide a good tradeoff between autoregressive structure and sampling time with more independence assumptions.
- New architecture design choices such as self-attention which introduce inductive biases that leverage a lot of computation per parameter introduced.

Autoregressive Models: Future

- Active topic with cutting edge results
- Lot of scope for more engineering and creative architecture design
- Larger models and datasets
- Successful in all of (un)conditional video, audio, text, images
- Sampling Time Engineering

Autoregressive Models: Future

- Active topic with cutting edge results
- Lot of scope for more engineering and creative architecture design
- Larger models and datasets
- Successful in all of (un)conditional video, audio, text, images
- Sampling Time Engineering

Autoregressive Models: Future

- Active topic with cutting edge results
- Lot of scope for more engineering and creative architecture design
- Larger models and datasets
- Successful in all of (un)conditional video, audio, text, images
- Sampling Time Engineering

Autoregressive Models: Future

- Active topic with cutting edge results
- Lot of scope for more engineering and creative architecture design
- Larger models and datasets
- Successful in all of (un)conditional video, audio, text, images
- Sampling Time Engineering

Autoregressive Models: Future

- Active topic with cutting edge results
- Lot of scope for more engineering and creative architecture design
- Larger models and datasets
- Successful in all of (un)conditional video, audio, text, images
- Sampling Time Engineering

Autoregressive Models: Future

- Active topic with cutting edge results
- Lot of scope for more engineering and creative architecture design
- Larger models and datasets
- Successful in all of (un)conditional video, audio, text, images
- Sampling Time Engineering

Autoregressive Models: Negatives

- No single layer of learned representation
- Currently, sampling time is slow for practical deployment.
- Not directly usable for downstream tasks.
- No interpolations.

Autoregressive Models: Negatives

- No single layer of learned representation
- Currently, sampling time is slow for practical deployment.
- Not directly usable for downstream tasks.
- No interpolations.

Autoregressive Models: Negatives

- No single layer of learned representation
- Currently, sampling time is slow for practical deployment.
- Not directly usable for downstream tasks.
- No interpolations.

Autoregressive Models: Negatives

- No single layer of learned representation
- Currently, sampling time is slow for practical deployment.
- Not directly usable for downstream tasks.
- No interpolations.

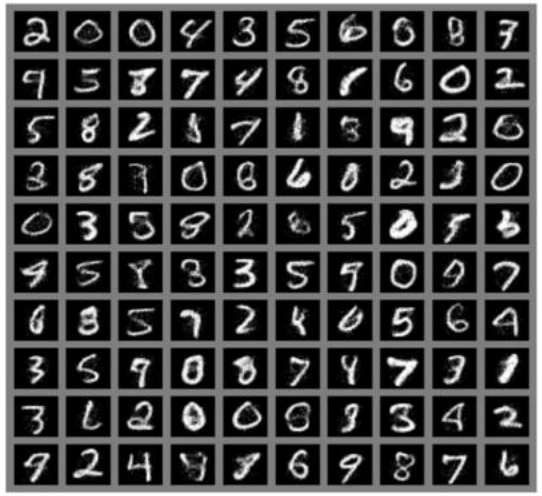
Autoregressive Models: Negatives

- No single layer of learned representation
- Currently, sampling time is slow for practical deployment.
- Not directly usable for downstream tasks.
- No interpolations.

Lecture overview

- Autoregressive models
- Flow models
  - NICE, RealNVP, Autoregressive Flows, Inverse Autoregressive Flows, Glow, Flow++
- Latent Variable models
- Implicit models

Flow Models



(a) Model trained on MNIST

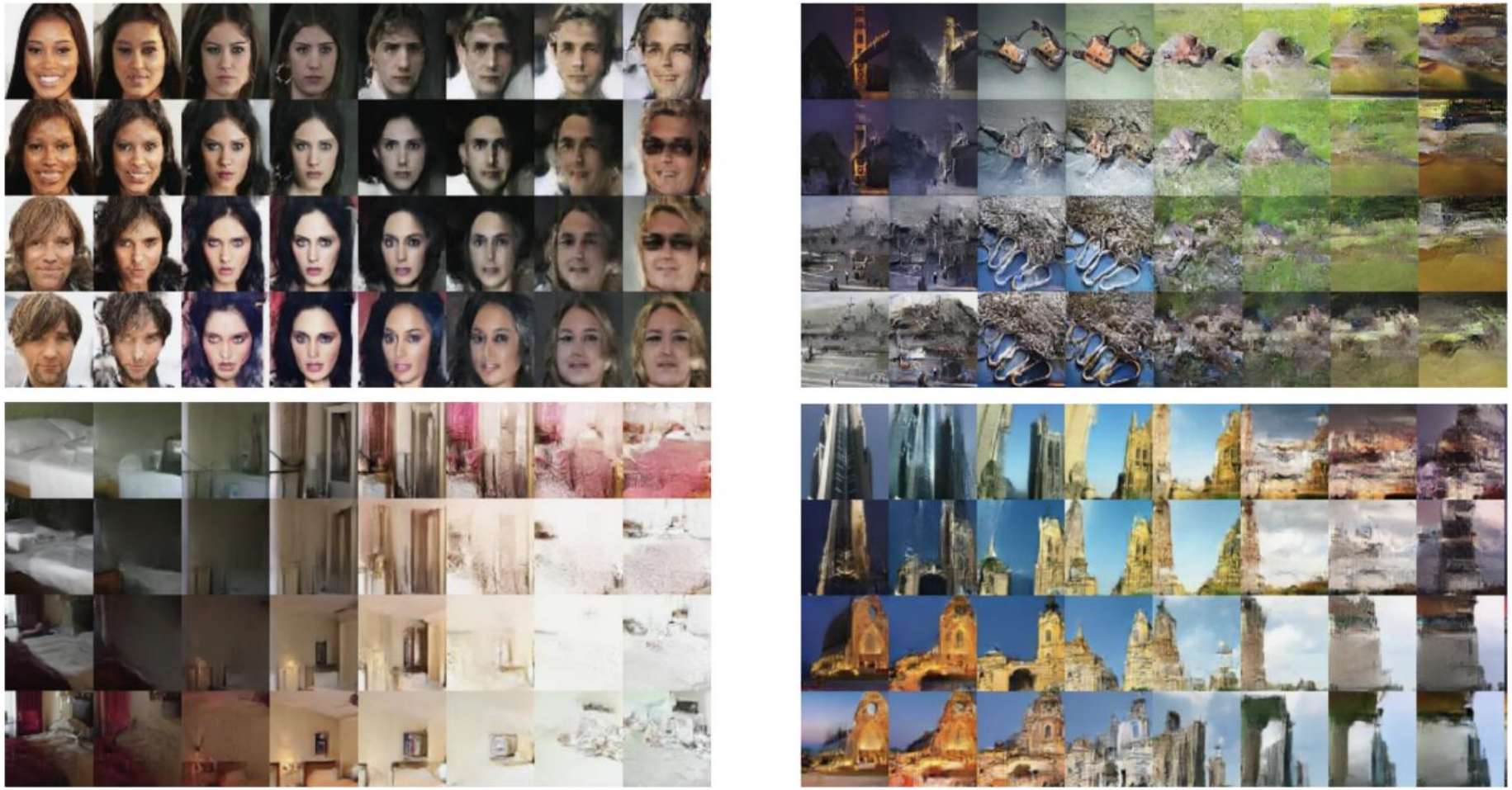
(b) Model trained on TFD

(c) Model trained on SVHN

(d) Model trained on CIFAR-10

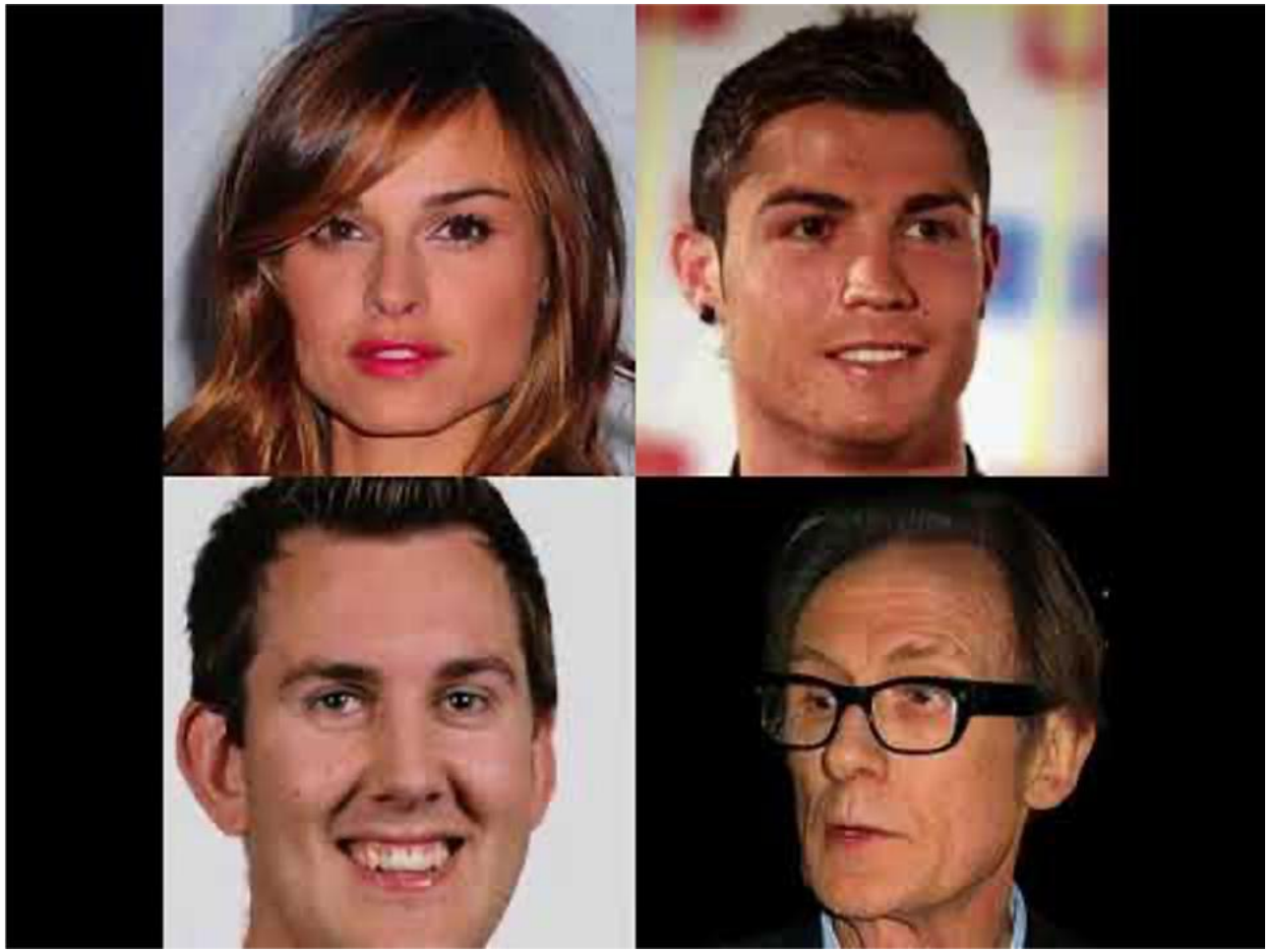
NICE (Dinh et al 2014)

Flow Models



RealNVP (Dinh et al 2016)

Glow: Big progress on sample quality



OpenAI Glow

Flow++: Progress on bits/dim on high entropy datasets

Model family	Model	CIFAR10 bits/dim	ImageNet 32x32 bits/dim	ImageNet 64x64 bits/dim
Non-autoregressive	RealNVP (Dinh et al., 2016)	3.49	4.28	–
	Glow (Kingma & Dhariwal, 2018)	3.35	4.09	3.81
	IAF-VAE (Kingma et al., 2016)	3.11	–	–
	Flow++ (ours)	**3.09**	**3.86**	**3.69**
Autoregressive	Multiscale PixelCNN (Reed et al., 2017)	–	3.95	3.70
	PixelCNN (van den Oord et al., 2016b)	3.14	–	–
	PixelRNN (van den Oord et al., 2016b)	3.00	3.86	3.63
	Gated PixelCNN (van den Oord et al., 2016c)	3.03	3.83	3.57
	PixelCNN++ (Salimans et al., 2017)	2.92	–	–
	Image Transformer (Parmar et al., 2018)	2.90	3.77	–
	PixelSNAIL (Chen et al., 2017)	2.85	3.80	3.52

Flow++

Flow Models: Future

- Learning the mask for coupling
- Close the gap with autoregressive models even further - use hybrid flows?
- Fewer expressive flows vs Several shallow flows
- Usage of Multiscale Loss - bits/dim vs sample quality tradeoffs
- Representation Learning with Flows
- Initialization

Flow Models: Future

- Learning the mask for coupling
- Close the gap with autoregressive models even further - use hybrid flows?
- Fewer expressive flows vs Several shallow flows
- Usage of Multiscale Loss - bits/dim vs sample quality tradeoffs
- Representation Learning with Flows
- Initialization

Flow Models: Future

- Learning the mask for coupling
- Close the gap with autoregressive models even further - use hybrid flows?
- Fewer expressive flows vs Several shallow flows
- Usage of Multiscale Loss - bits/dim vs sample quality tradeoffs
- Representation Learning with Flows
- Initialization

Flow Models: Future

- Learning the mask for coupling
- Close the gap with autoregressive models even further - use hybrid flows?
- Fewer expressive flows vs Several shallow flows
- Usage of Multiscale Loss - bits/dim vs sample quality tradeoffs
- Representation Learning with Flows
- Initialization

Flow Models: Future

- Learning the mask for coupling
- Close the gap with autoregressive models even further - use hybrid flows?
- Fewer expressive flows vs Several shallow flows
- Usage of Multiscale Loss - bits/dim vs sample quality tradeoffs
- Representation Learning with Flows
- Initialization

Flow Models: Future

- Learning the mask for coupling
- Close the gap with autoregressive models even further - use hybrid flows?
- Fewer expressive flows vs Several shallow flows
- Usage of Multiscale Loss - bits/dim vs sample quality tradeoffs
- Representation Learning with Flows
- Initialization

Flow Models: Future

- Learning the mask for coupling
- Close the gap with autoregressive models even further - use hybrid flows?
- Fewer expressive flows vs Several shallow flows
- Usage of Multiscale Loss - bits/dim vs sample quality tradeoffs
- Representation Learning with Flows
- Initialization

Flow Models: Future

- Glow-level samples with fewer parameters
- Glow-level samples on 1 MP (1024x1024) images.
- Dimension reduction
- Conditional Flow Models: Architecture and Execution
- Summary: Long way to go before GAN level samples and autoregressive model-level likelihood scores (and samples) combined with stable training and a fixed set of engineering practices.

Flow Models: Future

- Glow-level samples with fewer parameters
- Glow-level samples on 1 MP (1024x1024) images.
- Dimension reduction
- Conditional Flow Models: Architecture and Execution
- Summary: Long way to go before GAN level samples and autoregressive model-level likelihood scores (and samples) combined with stable training and a fixed set of engineering practices.

Flow Models: Future

- Glow-level samples with fewer parameters
- Glow-level samples on 1 MP (1024x1024) images.
- Dimension reduction
- Conditional Flow Models: Architecture and Execution
- Summary: Long way to go before GAN level samples and autoregressive model-level likelihood scores (and samples) combined with stable training and a fixed set of engineering practices.

Flow Models: Future

- Glow-level samples with fewer parameters
- Glow-level samples on 1 MP (1024x1024) images.
- Dimension reduction
- Conditional Flow Models: Architecture and Execution
- Summary: Long way to go before GAN level samples and autoregressive model-level likelihood scores (and samples) combined with stable training and a fixed set of engineering practices.

Flow Models: Future

- Glow-level samples with fewer parameters
- Glow-level samples on 1 MP (1024x1024) images.
- Dimension reduction
- Conditional Flow Models: Architecture and Execution
- Summary: Long way to go before GAN level samples and autoregressive model-level likelihood scores (and samples) combined with stable training and a fixed set of engineering practices.

Flow Models: Future

- Glow-level samples with fewer parameters
- Glow-level samples on 1 MP (1024x1024) images.
- Dimension reduction
- Conditional Flow Models: Architecture and Execution
- Summary: Long way to go before GAN level samples and autoregressive model-level likelihood scores (and samples) combined with stable training and a fixed set of engineering practices.

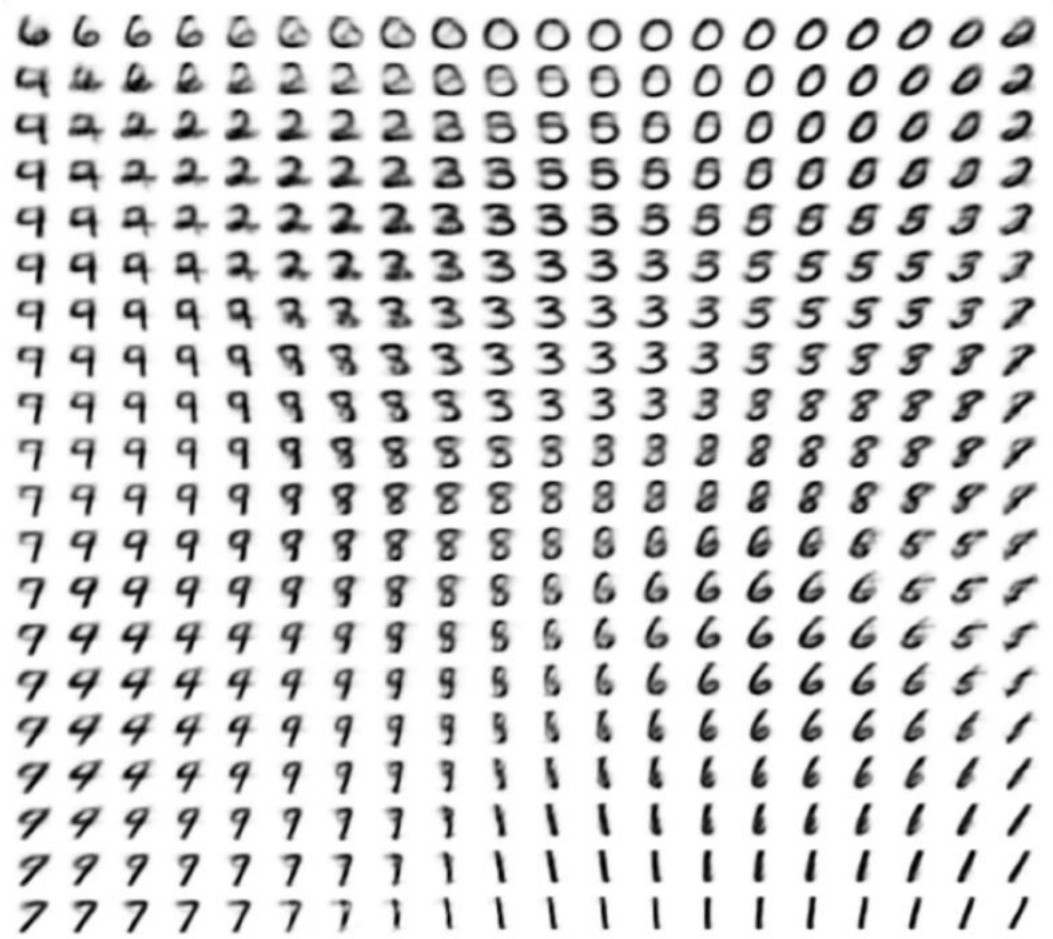
Flow Models: Negatives

- z is as big as x. Models end up becoming big.
- As of now, no notion of lower dimensional embedding.
- Careful initialization (not really a negative)

Lecture overview

- Autoregressive models
- Flow models
- Latent Variable models
  - Approximate likelihood with Variational Lower Bound
  - Variational Auto-Encoder, IWAE, IAF-VAE, PixelVAE (VLAE), VQ-VAE
- Implicit models

Latent Variable Models

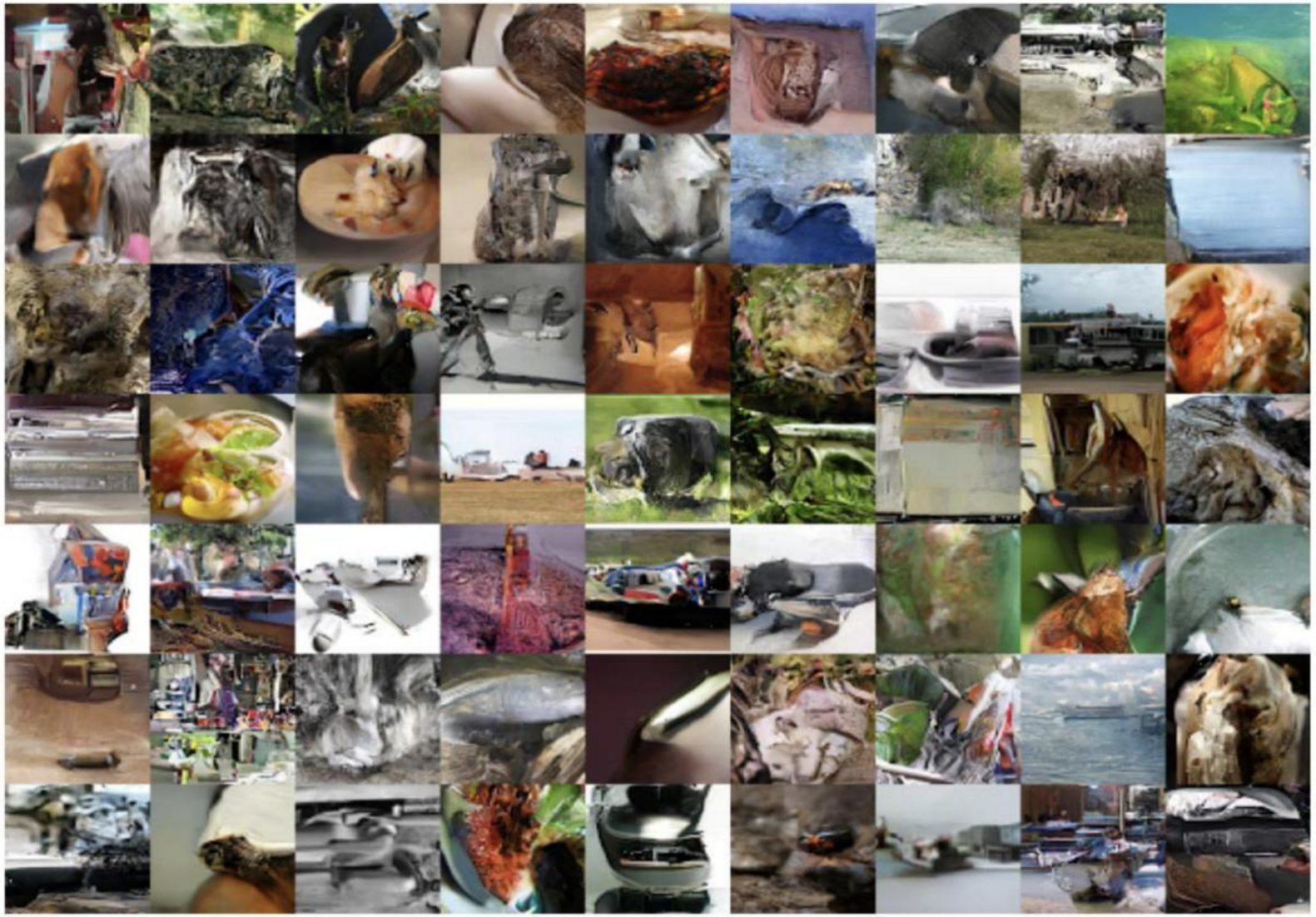


Auto-Encoding Variational Bayes
(Kingma 2013)

Latent Variable Models: PixelVAE

PixelVAE
Gularajani et al
2016

Latent Variable Models: PixelVAE



PixelVAE
Gularajani et al
2016

Latent Variable Models - BIVA

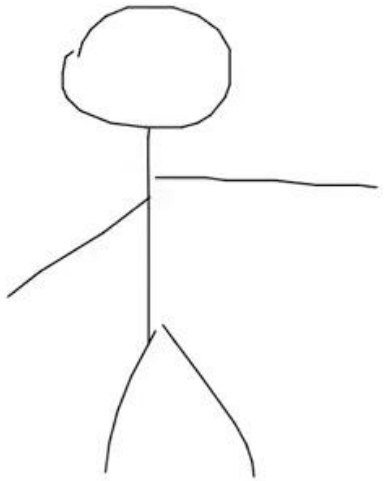
BIVA
(Maaloe et al 2019)

Well known VAE Applications

- Sketch-RNN
- World Models
- Visual concepts for RL (beta-VAE)
- Generative Query Networks

Well known VAE Applications: Sketch-RNN

yoga poses generated by moving through the learned representation (latent space) of the model trained on yoga drawings



https://magenta.tensorflow.org/assets/sketch_rnn_demo/index.html

Well known VAE Applications: World Models

At each time step, our agent receives an **observation** from the environment.

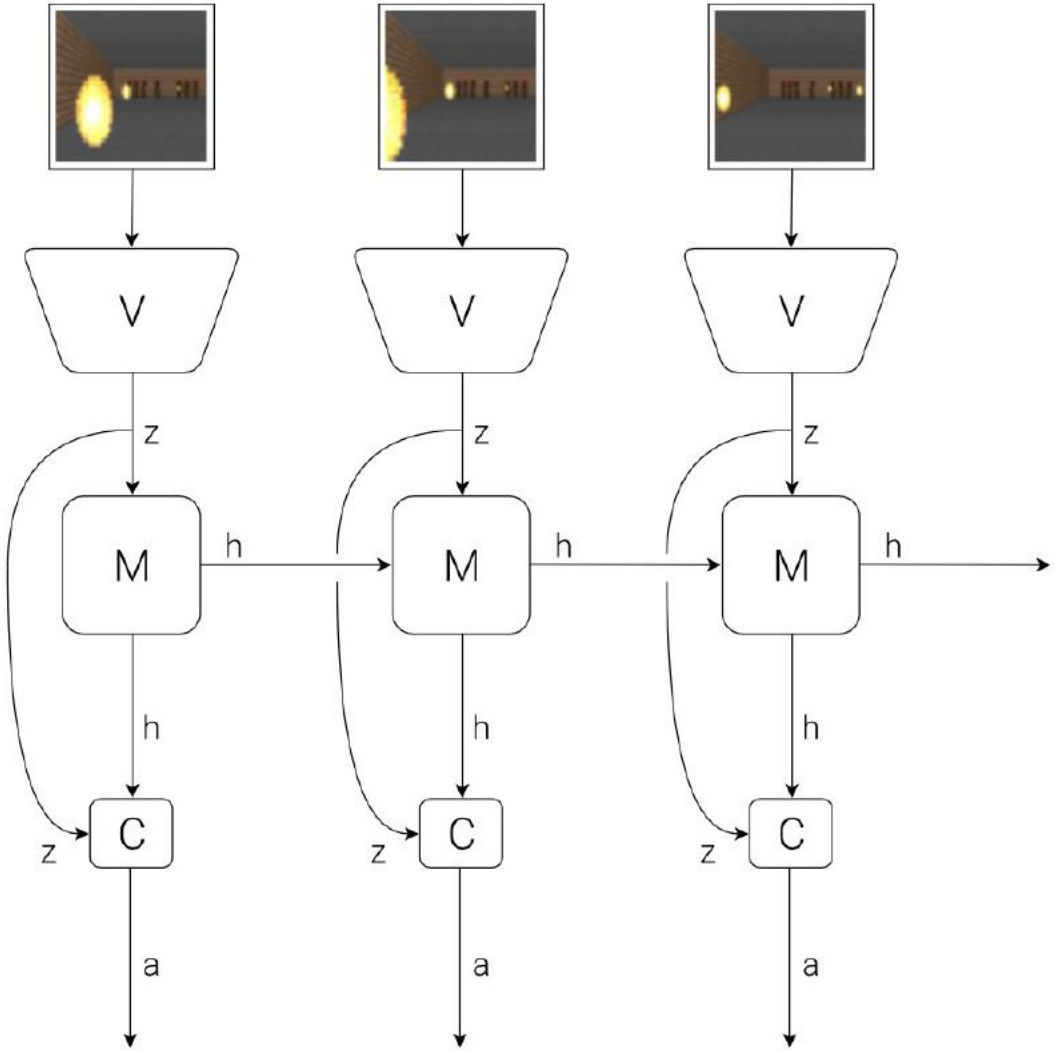
World Model

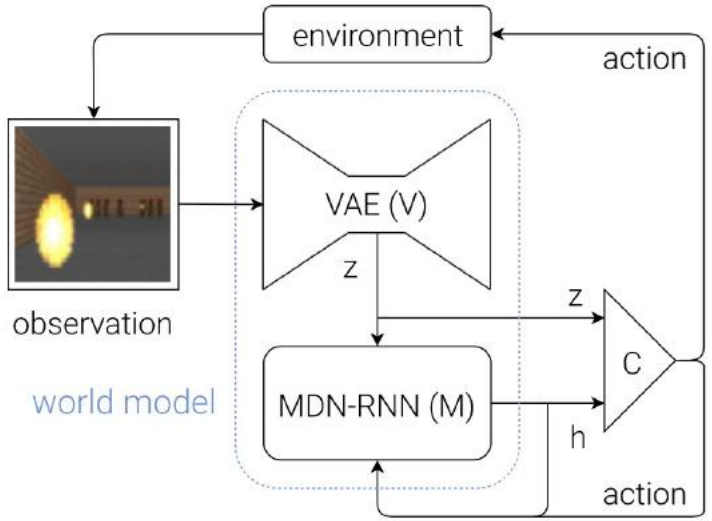
The **Vision Model (V)** encodes the high-dimensional observation into a low-dimensional latent vector.

The **Memory RNN (M)** integrates the historical codes to create a representation that can predict future states.

A small **Controller (C)** uses the representations from both **V** and **M** to select good actions.

The agent performs **actions** that go back and affect the environment.





https://worldmodels.github.io

Well known VAE Applications: beta-VAE

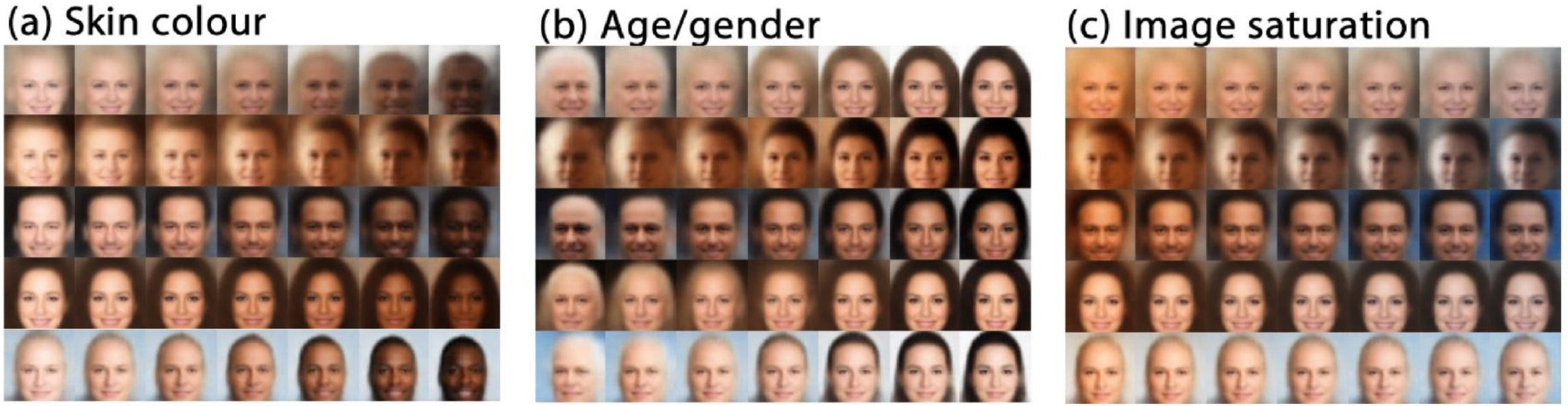
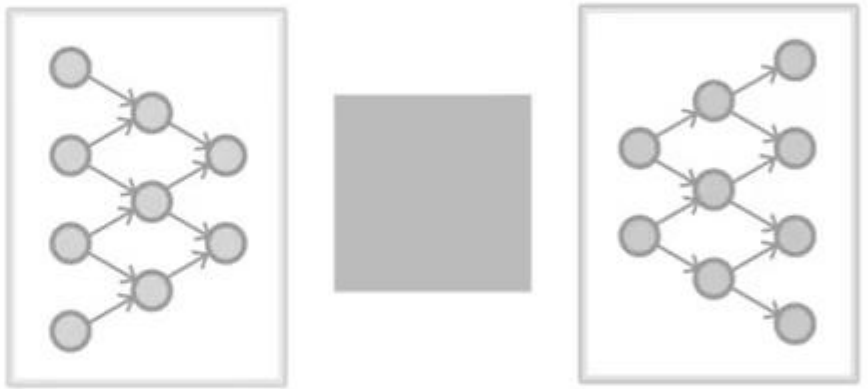


Figure 4: **Latent factors learnt by β-VAE on celebA:** traversal of individual latents demonstrates that β-VAE discovered in an unsupervised manner factors that encode skin colour, transition from an elderly male to younger female, and image saturation.

Well known VAE Applications: Generative Query Networks



VAE: Advantages

- Notion of “compressed” representation learning
- Also gives you approximate log-likelihood
- Interpolations, retrospective analysis of what the model learns
- Disentangled representations
- Generative Model + Density Model + Latent Variables + Dimensionality Reduction

VAE: Advantages

- Notion of “compressed” representation learning
- Also gives you approximate log-likelihood
- Interpolations, retrospective analysis of what the model learns
- Disentangled representations
- Generative Model + Density Model + Latent Variables + Dimensionality Reduction

VAE: Advantages

- Notion of “compressed” representation learning
- Also gives you approximate log-likelihood
- Interpolations, retrospective analysis of what the model learns
- Disentangled representations
- Generative Model + Density Model + Latent Variables + Dimensionality Reduction

VAE: Advantages

- Notion of “compressed” representation learning
- Also gives you approximate log-likelihood
- Interpolations, retrospective analysis of what the model learns
- Disentangled representations
- Generative Model + Density Model + Latent Variables + Dimensionality Reduction

VAE: Advantages

- Notion of “compressed” representation learning
- Also gives you approximate log-likelihood
- Interpolations, retrospective analysis of what the model learns
- Disentangled representations
- Generative Model + Density Model + Latent Variables + Dimensionality Reduction

VAE: Advantages

- Notion of “compressed” representation learning
- Also gives you approximate log-likelihood
- Interpolations, retrospective analysis of what the model learns
- Disentangled representations
- Generative Model + Density Model + Latent Variables + Dimensionality Reduction

VAE: Disadvantages

- Blurry samples
- Factorized gaussian posterior or decoder assumptions maybe too limiting
- Success on large scale is still on-going work
- Encouraging disentanglement with the KL term still only shown on relatively toy domains
- There maybe other ways to learn better representations or to get better samples or get better density estimates (basically, not the best at any one thing but gives you all together)

VAE: Disadvantages

- Blurry samples
- Factorized gaussian posterior or decoder assumptions maybe too limiting
- Success on large scale is still on-going work
- Encouraging disentanglement with the KL term still only shown on relatively toy domains
- There maybe other ways to learn better representations or to get better samples or get better density estimates (basically, not the best at any one thing but gives you all together)

VAE: Disadvantages

- Blurry samples
- Factorized gaussian posterior or decoder assumptions maybe too limiting
- Success on large scale is still on-going work
- Encouraging disentanglement with the KL term still only shown on relatively toy domains
- There maybe other ways to learn better representations or to get better samples or get better density estimates (basically, not the best at any one thing but gives you all together)

VAE: Disadvantages

- Blurry samples
- Factorized gaussian posterior or decoder assumptions maybe too limiting
- Success on large scale is still on-going work
- Encouraging disentanglement with the KL term still only shown on relatively toy domains
- There maybe other ways to learn better representations or to get better samples or get better density estimates (basically, not the best at any one thing but gives you all together)

VAE: Disadvantages

- Blurry samples
- Factorized gaussian posterior or decoder assumptions maybe too limiting
- Success on large scale is still on-going work
- Encouraging disentanglement with the KL term still only shown on relatively toy domains
- There maybe other ways to learn better representations or to get better samples or get better density estimates (basically, not the best at any one thing but gives you all together)

VAE: Disadvantages

- Blurry samples
- Factorized gaussian posterior or decoder assumptions maybe too limiting
- Success on large scale is still on-going work
- Encouraging disentanglement with the KL term still only shown on relatively toy domains
- There maybe other ways to learn better representations or to get better samples or get better density estimates (basically, not the best at any one thing but gives you all together)

VAE: Future

- Modern decoders [cross-entropy based, weakly autoregressive]
- More powerful posteriors
- Hierarchical latent variable models to learn coarse and fine features and interpolations
- Discrete latent variable models to prevent posterior collapse and still be able to use PixelCNN-like decoders
- Scale at the level of Flow Models training

VAE: Future

- Modern decoders [cross-entropy based, weakly autoregressive]
- More powerful posteriors
- Hierarchical latent variable models to learn coarse and fine features and interpolations
- Discrete latent variable models to prevent posterior collapse and still be able to use PixelCNN-like decoders
- Scale at the level of Flow Models training

VAE: Future

- Modern decoders [cross-entropy based, weakly autoregressive]
- More powerful posteriors
- Hierarchical latent variable models to learn coarse and fine features and interpolations
- Discrete latent variable models to prevent posterior collapse and still be able to use PixelCNN-like decoders
- Scale at the level of Flow Models training

VAE: Future

- Modern decoders [cross-entropy based, weakly autoregressive]
- More powerful posteriors
- Hierarchical latent variable models to learn coarse and fine features and interpolations
- Discrete latent variable models to prevent posterior collapse and still be able to use PixelCNN-like decoders
- Scale at the level of Flow Models training

VAE: Future

- Modern decoders [cross-entropy based, weakly autoregressive]
- More powerful posteriors
- Hierarchical latent variable models to learn coarse and fine features and interpolations
- Discrete latent variable models to prevent posterior collapse and still be able to use PixelCNN-like decoders
- Scale at the level of Flow Models training

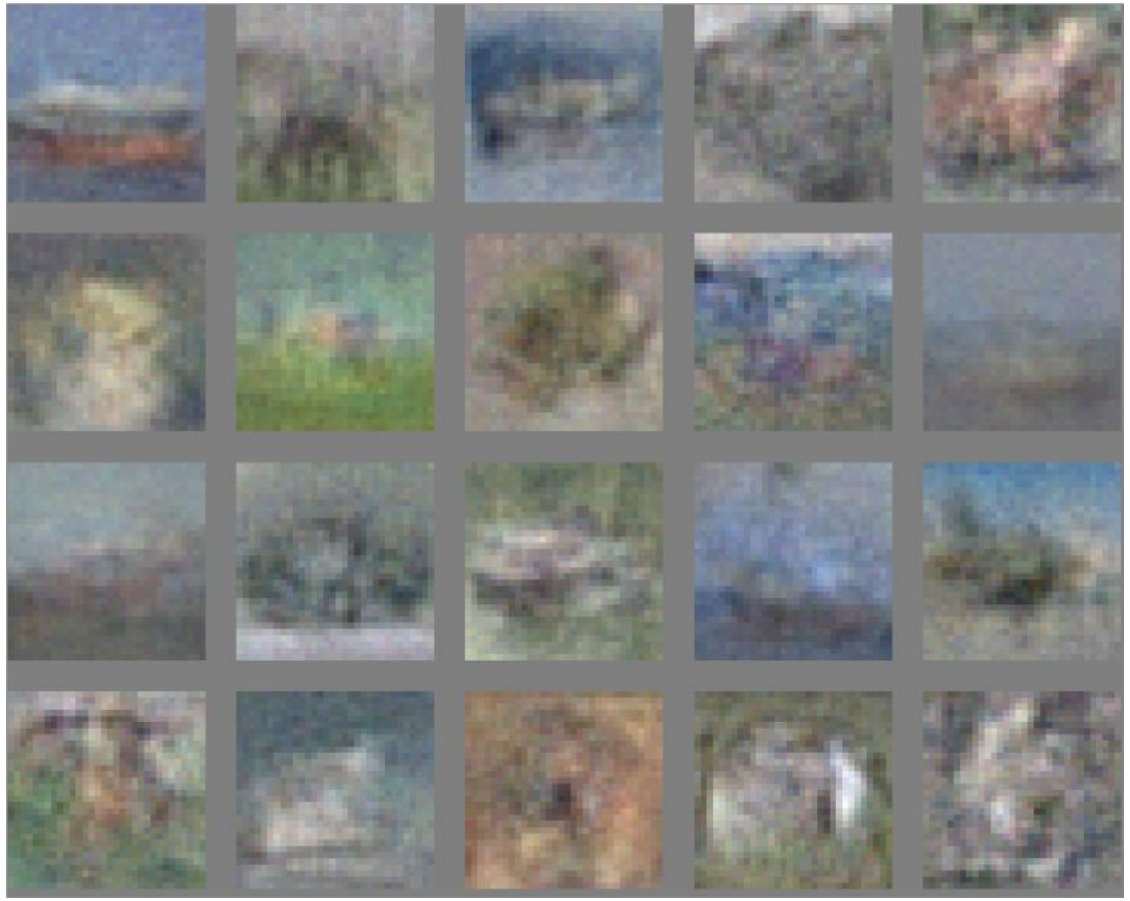
VAE: Future

- Modern decoders [cross-entropy based, weakly autoregressive]
- More powerful posteriors
- Hierarchical latent variable models to learn coarse and fine features and interpolations
- Discrete latent variable models to prevent posterior collapse and still be able to use PixelCNN-like decoders
- Scale at the level of Flow Models training

Lecture overview

- Autoregressive models
- Flow models
- Latent Variable models
- **Implicit models**
  - **Generative Adversarial Networks (GAN)**

Generative Adversarial Networks

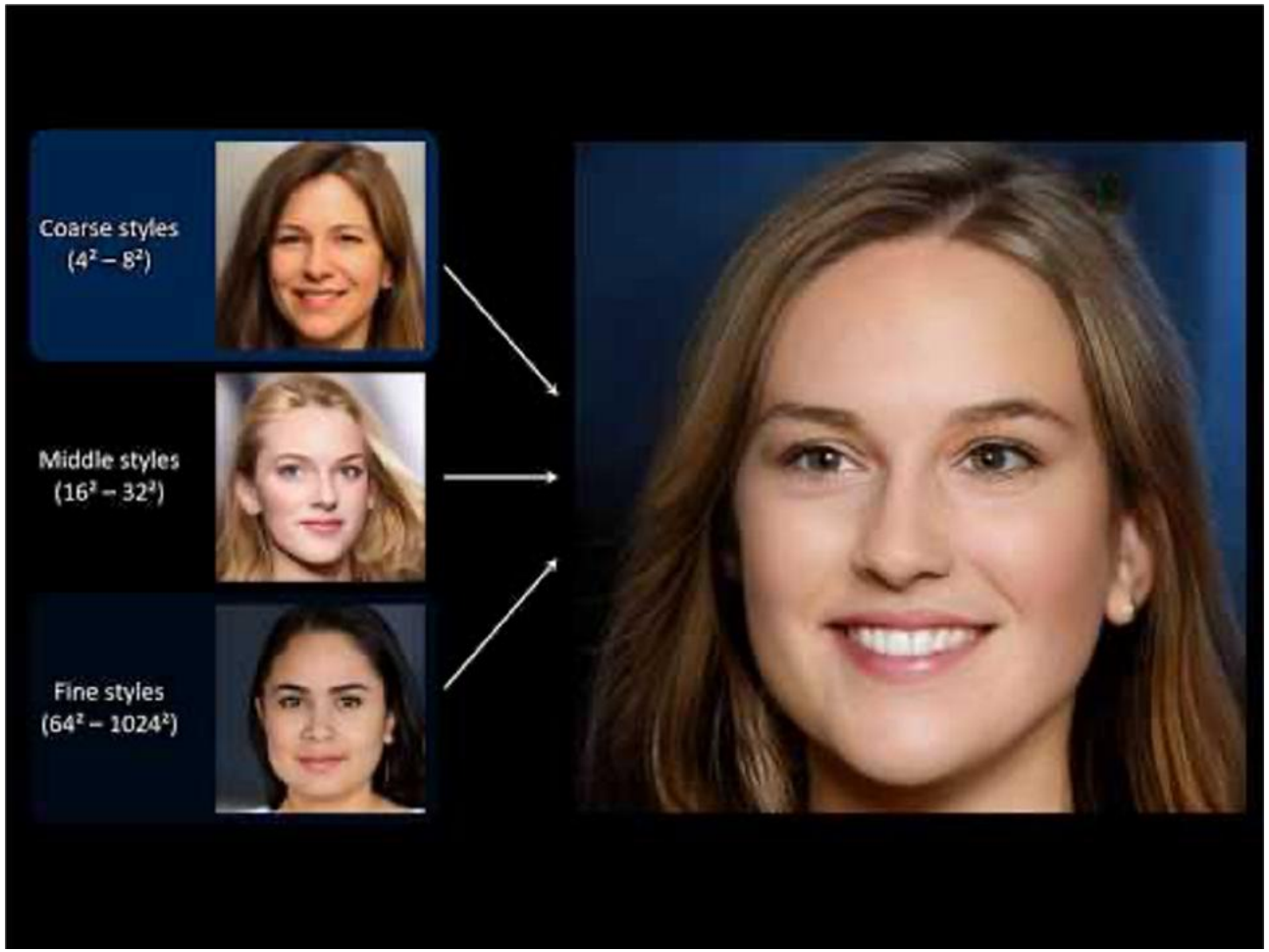


Original GAN (2014) - Goodfellow et al

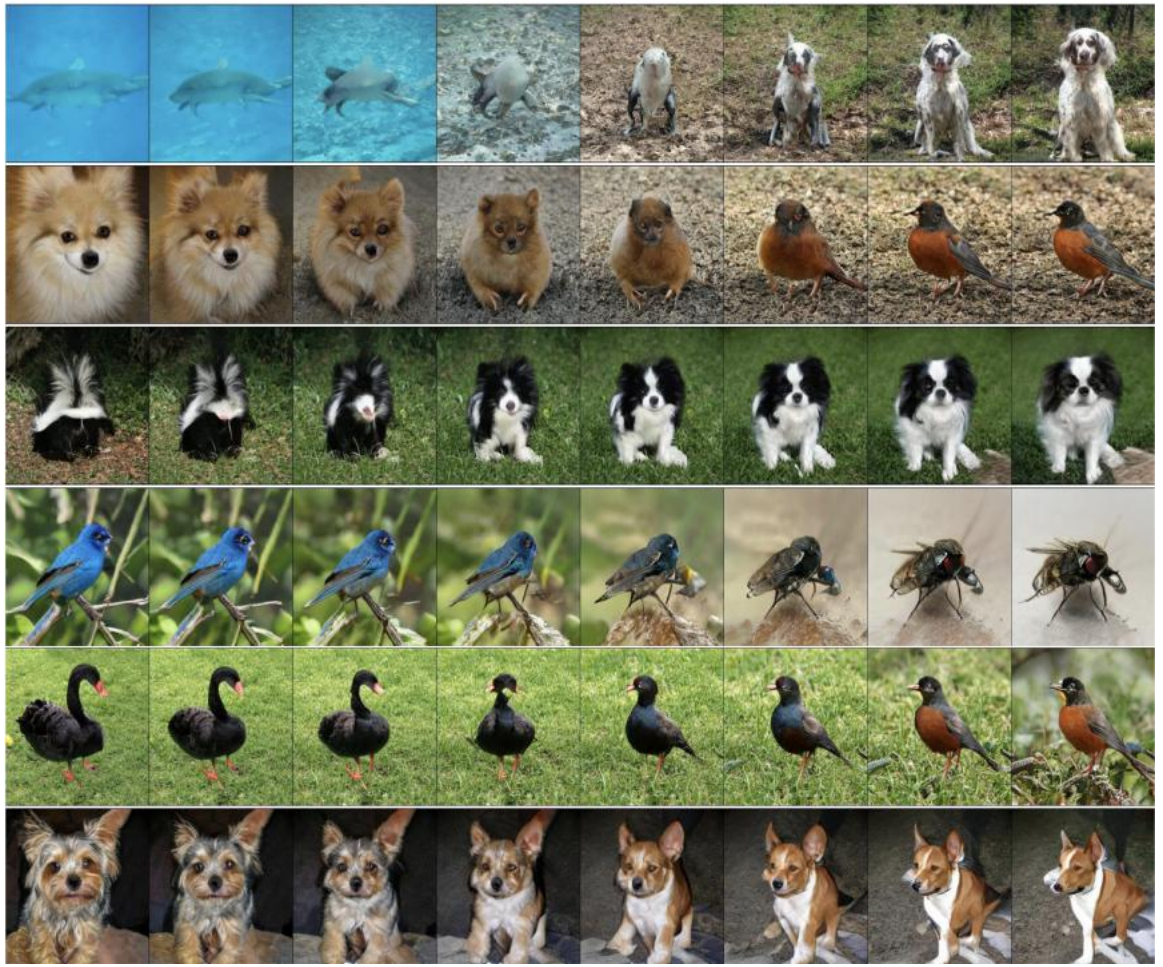
Generative Adversarial Networks

DCGAN - Radford, Metz, Chintala 2015

Generative Adversarial Networks



StyleGAN



BigGAN

Generative Adversarial Networks: Future

- Hard to predict against them given an array of the most powerful generation results for images.
- Progress in unconditional GANs.
- Handling more fine-grained details
- More complex scenes (multiple people with objects)
- Video generation

Generative Adversarial Networks: Future

- Lipschitzness constraints (New approaches)
- Conditioning tricks (Feeding in noise at different levels, batch / instance normalization)
- Architecture design (Upsampling, downsampling, deep vs wide tradeoff)
- Objective functions (Hinge Loss …)
- Stability and scalability (Deeper models with fewer parameters + larger batch sizes)
- Perturbations at different levels (StyleGAN) + Coarse / Fine interpolations

Generative Adversarial Networks: Future

- Lipschitzness constraints (New approaches)
- Conditioning tricks (Feeding in noise at different levels, batch / instance normalization)
- Architecture design (Upsampling, downsampling, deep vs wide tradeoff)
- Objective functions (Hinge Loss …)
- Stability and scalability (Deeper models with fewer parameters + larger batch sizes)
- Perturbations at different levels (StyleGAN) + Coarse / Fine interpolations

Generative Adversarial Networks: Future

- Lipschitzness constraints (New approaches)
- Conditioning tricks (Feeding in noise at different levels, batch / instance normalization)
- Architecture design (Upsampling, downsampling, deep vs wide tradeoff)
- Objective functions (Hinge Loss …)
- Stability and scalability (Deeper models with fewer parameters + larger batch sizes)
- Perturbations at different levels (StyleGAN) + Coarse / Fine interpolations

Generative Adversarial Networks: Future

- Lipschitzness constraints (New approaches)
- Conditioning tricks (Feeding in noise at different levels, batch / instance normalization)
- Architecture design (Upsampling, downsampling, deep vs wide tradeoff)
- Objective functions (Hinge Loss …)
- Stability and scalability (Deeper models with fewer parameters + larger batch sizes)
- Perturbations at different levels (StyleGAN) + Coarse / Fine interpolations

Generative Adversarial Networks: Future

- Lipschitzness constraints (New approaches)
- Conditioning tricks (Feeding in noise at different levels, batch / instance normalization)
- Architecture design (Upsampling, downsampling, deep vs wide tradeoff)
- Objective functions (Hinge Loss …)
- Stability and scalability (Deeper models with fewer parameters + larger batch sizes)
- Perturbations at different levels (StyleGAN) + Coarse / Fine interpolations

Generative Adversarial Networks: Future

- Lipschitzness constraints (New approaches)
- Conditioning tricks (Feeding in noise at different levels, batch / instance normalization)
- Architecture design (Upsampling, downsampling, deep vs wide tradeoff)
- Objective functions (Hinge Loss …)
- Stability and scalability (Deeper models with fewer parameters + larger batch sizes)
- Perturbations at different levels (StyleGAN) + Coarse / Fine interpolations

Generative Adversarial Networks: Future

- Lipschitzness constraints (New approaches)
- Conditioning tricks (Feeding in noise at different levels, batch / instance normalization)
- Architecture design (Upsampling, downsampling, deep vs wide tradeoff)
- Objective functions (Hinge Loss …)
- Stability and scalability (Deeper models with fewer parameters + larger batch sizes)
- Perturbations at different levels (StyleGAN) + Coarse / Fine interpolations

Generative Adversarial Networks: Negatives

- Plenty of varying engineering tricks and details
- Hard to know which piece is significantly helping push the cutting edge results
- Ablations for large scale datasets are time-consuming
- Unconditional GANs - sample diversity (or mode dropping behavior)
- Evaluation metrics to account for generalization
- Ablations / Key pieces / engineering details isn't a negative specific to GANs

Generative Adversarial Networks: Negatives

- Plenty of varying engineering tricks and details
- Hard to know which piece is significantly helping push the cutting edge results
- Ablations for large scale datasets are time-consuming
- Unconditional GANs - sample diversity (or mode dropping behavior)
- Evaluation metrics to account for generalization
- Ablations / Key pieces / engineering details isn't a negative specific to GANs

Generative Adversarial Networks: Negatives

- Plenty of varying engineering tricks and details
- Hard to know which piece is significantly helping push the cutting edge results
- Ablations for large scale datasets are time-consuming
- Unconditional GANs - sample diversity (or mode dropping behavior)
- Evaluation metrics to account for generalization
- Ablations / Key pieces / engineering details isn't a negative specific to GANs

Generative Adversarial Networks: Negatives

- Plenty of varying engineering tricks and details
- Hard to know which piece is significantly helping push the cutting edge results
- Ablations for large scale datasets are time-consuming
- Unconditional GANs - sample diversity (or mode dropping behavior)
- Evaluation metrics to account for generalization
- Ablations / Key pieces / engineering details isn't a negative specific to GANs

Generative Adversarial Networks: Negatives

- Plenty of varying engineering tricks and details
- Hard to know which piece is significantly helping push the cutting edge results
- Ablations for large scale datasets are time-consuming
- Unconditional GANs - sample diversity (or mode dropping behavior)
- Evaluation metrics to account for generalization
- Ablations / Key pieces / engineering details isn't a negative specific to GANs

Generative Adversarial Networks: Negatives

- Plenty of varying engineering tricks and details
- Hard to know which piece is significantly helping push the cutting edge results
- Ablations for large scale datasets are time-consuming
- Unconditional GANs - sample diversity (or mode dropping behavior)
- Evaluation metrics to account for generalization
- Ablations / Key pieces / engineering details isn't a negative specific to GANs

Generative Adversarial Networks: Negatives

- Plenty of varying engineering tricks and details
- Hard to know which piece is significantly helping push the cutting edge results
- Ablations for large scale datasets are time-consuming
- Unconditional GANs - sample diversity (or mode dropping behavior)
- Evaluation metrics to account for generalization
- Ablations / Key pieces / engineering details isn’t a negative specific to GANs

GANs or Density Models?

- Not true that density models do not need comparable level of engineering details and hacks to work as GANs (e.g.: FiLM conditioning, gating, LayerNorm, ActNorm).
- Not true that there is absolutely no theoretical understanding of GANs: Lag behind empirical practice and difficulty doesn't mean non-existence
- Blurry / improbable samples (vs) Mode collapse :: Compression at the cost of sample quality : Sample quality at the cost of missing modes
- Apart from amazing samples, GANs are more popular because:
  - Works well with less compute (Ex: Good 1024 x 1024 (megapixel) Celeb A samples with few couple hours of training on a single V100 GPU)
  - Density models are huge, require distributed training - not doable by too many people. Takes a lot more time to output reasonably sharp samples.
  - Interpolations and conditional generation [some success on Glow but not possible with autoregressive models] - adoption by artists.

GANs or Density Models?

- Not true that density models do not need comparable level of engineering details and hacks to work as GANs (e.g.: FiLM conditioning, gating, LayerNorm, ActNorm).
- Not true that there is absolutely no theoretical understanding of GANs: Lag behind empirical practice and difficulty doesn't mean non-existence
- Blurry / improbable samples (vs) Mode collapse :: Compression at the cost of sample quality : Sample quality at the cost of missing modes
- Apart from amazing samples, GANs are more popular because:
  - Works well with less compute (Ex: Good 1024 x 1024 (megapixel) Celeb A samples with few couple hours of training on a single V100 GPU)
  - Density models are huge, require distributed training - not doable by too many people. Takes a lot more time to output reasonably sharp samples.
  - Interpolations and conditional generation [some success on Glow but not possible with autoregressive models] - adoption by artists.

GANs or Density Models?

- Not true that density models do not need comparable level of engineering details and hacks to work as GANs (e.g.: FiLM conditioning, gating, LayerNorm, ActNorm).
- Not true that there is absolutely no theoretical understanding of GANs: Lag behind empirical practice and difficulty doesn't mean non-existence
- Blurry / improbable samples (vs) Mode collapse :: Compression at the cost of sample quality : Sample quality at the cost of missing modes
- Apart from amazing samples, GANs are more popular because:
  - Works well with less compute (Ex: Good 1024 x 1024 (megapixel) Celeb A samples with few couple hours of training on a single V100 GPU)
  - Density models are huge, require distributed training - not doable by too many people. Takes a lot more time to output reasonably sharp samples.
  - Interpolations and conditional generation [some success on Glow but not possible with autoregressive models] - adoption by artists.

GANs or Density Models?

- Not true that density models do not need comparable level of engineering details and hacks to work as GANs (e.g.: FiLM conditioning, gating, LayerNorm, ActNorm).
- Not true that there is absolutely no theoretical understanding of GANs: Lag behind empirical practice and difficulty doesn't mean non-existence
- Blurry / improbable samples (vs) Mode collapse :: Compression at the cost of sample quality : Sample quality at the cost of missing modes
- Apart from amazing samples, GANs are more popular because:
  - Works well with less compute (Ex: Good 1024 x 1024 (megapixel) Celeb A samples with few couple hours of training on a single V100 GPU)
  - Density models are huge, require distributed training - not doable by too many people. Takes a lot more time to output reasonably sharp samples.
  - Interpolations and conditional generation [some success on Glow but not possible with autoregressive models] - adoption by artists.

GANs or Density Models?

- Not true that density models do not need comparable level of engineering details and hacks to work as GANs (e.g.: FiLM conditioning, gating, LayerNorm, ActNorm).
- Not true that there is absolutely no theoretical understanding of GANs: Lag behind empirical practice and difficulty doesn't mean non-existence
- Blurry / improbable samples (vs) Mode collapse :: Compression at the cost of sample quality : Sample quality at the cost of missing modes
- Apart from amazing samples, GANs are more popular because:
  - Works well with less compute (Ex: Good 1024 x 1024 (megapixel) Celeb A samples with few couple hours of training on a single V100 GPU)
  - Density models are huge, require distributed training - not doable by too many people. Takes a lot more time to output reasonably sharp samples.
  - Interpolations and conditional generation [some success on Glow but not possible with autoregressive models] - adoption by artists.

GANs or Density Models?

- Not true that density models do not need comparable level of engineering details and hacks to work as GANs (e.g.: FiLM conditioning, gating, LayerNorm, ActNorm).
- Not true that there is absolutely no theoretical understanding of GANs: Lag behind empirical practice and difficulty doesn't mean non-existence
- Blurry / improbable samples (vs) Mode collapse :: Compression at the cost of sample quality : Sample quality at the cost of missing modes
- Apart from amazing samples, GANs are more popular because:
  - Works well with less compute (Ex: Good 1024 x 1024 (megapixel) Celeb A samples with few couple hours of training on a single V100 GPU)
  - Density models are huge, require distributed training - not doable by too many people. Takes a lot more time to output reasonably sharp samples.
  - Interpolations and conditional generation [some success on Glow but not possible with autoregressive models] - adoption by artists.

GANs or Density Models?

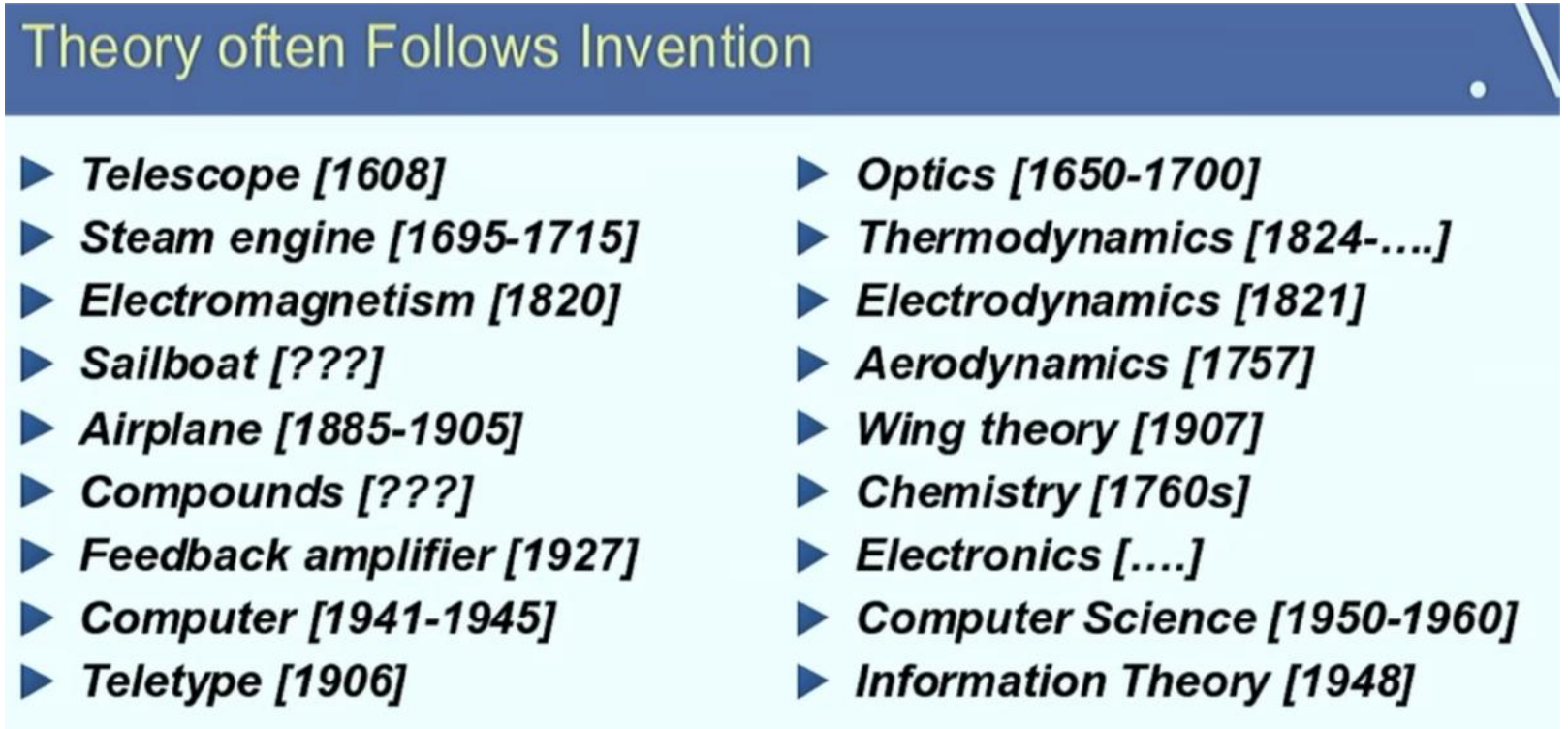
- Not true that density models do not need comparable level of engineering details and hacks to work as GANs (e.g.: FiLM conditioning, gating, LayerNorm, ActNorm).
- Not true that there is absolutely no theoretical understanding of GANs: Lag behind empirical practice and difficulty doesn't mean non-existence
- Blurry / improbable samples (vs) Mode collapse :: Compression at the cost of sample quality : Sample quality at the cost of missing modes
- Apart from amazing samples, GANs are more popular because:
  - Works well with less compute (Ex: Good 1024 x 1024 (megapixel) Celeb A samples with few couple hours of training on a single V100 GPU)
  - Density models are huge, require distributed training - not doable by too many people. Takes a lot more time to output reasonably sharp samples.
  - Interpolations and conditional generation [some success on Glow but not possible with autoregressive models] - adoption by artists.

GANs or Density Models?

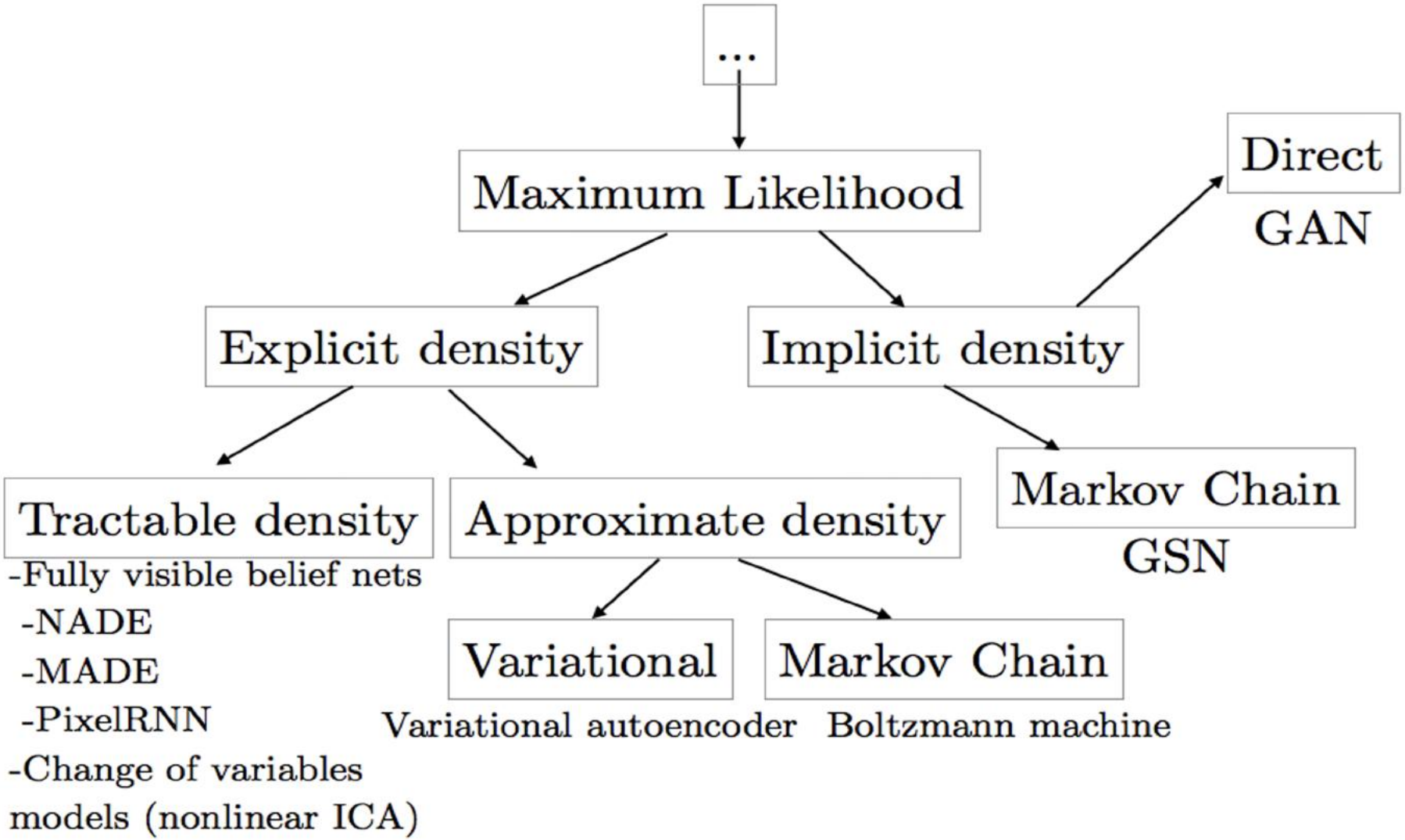
- Not true that density models do not need comparable level of engineering details and hacks to work as GANs (e.g.: FiLM conditioning, gating, LayerNorm, ActNorm).
- Not true that there is absolutely no theoretical understanding of GANs: Lag behind empirical practice and difficulty doesn't mean non-existence
- Blurry / improbable samples (vs) Mode collapse :: Compression at the cost of sample quality : Sample quality at the cost of missing modes
- Apart from amazing samples, GANs are more popular because:
  - Works well with less compute (Ex: Good 1024 x 1024 (megapixel) Celeb A samples with few couple hours of training on a single V100 GPU)
  - Density models are huge, require distributed training - not doable by too many people. Takes a lot more time to output reasonably sharp samples.
  - Interpolations and conditional generation [some success on Glow but not possible with autoregressive models] - adoption by artists.

GANs or Density Models?

- **Bright side**: Upto to aesthetics / taste in terms of betting on one (density models vs GAN)
- Many technological advances in the past have been possible without rigorous science built for them (science followed later) [Le Cun: Epistemology of DL]



Taxonomy


...
Maximum Likelihood
Direct
GAN
Explicit density
Implicit density
Markov Chain
GSN
Tractable density
Approximate density
-Fully visible belief nets
-NADE
-MADE
-PixelRNN
-Change of variables
models (nonlinear ICA)
Variational
Markov Chain
Variational autoencoder
Boltzmann machine

If training density models...

- If you only care about density and not sampling, go for autoregressive models
- If you care about sampling time but not too much, autoregressive is still fine. Just use smaller models and preferably RNNs [or write efficient convolution / self-attention]

- If you really can't afford linear (in number of dimensions) sampling, weaker autoregressive models (log time) such as Parallel PixelCNN are worth considering.
- Notion that autoregressive models are meant for "discrete" is unfounded.
  Ex: Alex Graves' Handwriting Recognition, Sketch-RNN, World Models, CPC.

- Flow Models are good for modeling densities of continuous valued data and are getting better for discrete (pixels). Larger models needed for complex datasets.
- If you want both representations and sampling or just want to try the simplest thing first, variational auto-encoders with factorized decoders are a natural first choice.

If training density models...

- If you only care about density and not sampling, go for autoregressive models
- If you care about sampling time but not too much, autoregressive is still fine. Just use smaller models and preferably RNNs [or write efficient convolution / self-attention]

- If you really can't afford linear (in number of dimensions) sampling, weaker autoregressive models (log time) such as Parallel PixelCNN are worth considering.
- Notion that autoregressive models are meant for "discrete" is unfounded. Ex: Alex Graves' Handwriting Recognition, Sketch-RNN, World Models, CPC.

- Flow Models are good for modeling densities of continuous valued data and are getting better for discrete (pixels). Larger models needed for complex datasets.
- If you want both representations and sampling or just want to try the simplest thing first, variational auto-encoders with factorized decoders are a natural first choice.

If training density models...

- If you only care about density and not sampling, go for autoregressive models
- If you care about sampling time but not too much, autoregressive is still fine. Just use smaller models and preferably RNNs [or write efficient convolution / self-attention]

- If you really can't afford linear (in number of dimensions) sampling, weaker autoregressive models (log time) such as Parallel PixelCNN are worth considering.
- Notion that autoregressive models are meant for "discrete" is unfounded. Ex: Alex Graves' Handwriting Recognition, Sketch-RNN, World Models, CPC.

- Flow Models are good for modeling densities of continuous valued data and are getting better for discrete (pixels). Larger models needed for complex datasets.
- If you want both representations and sampling or just want to try the simplest thing first, variational auto-encoders with factorized decoders are a natural first choice.

If training density models...

- If you only care about density and not sampling, go for autoregressive models
- If you care about sampling time but not too much, autoregressive is still fine. Just use smaller models and preferably RNNs [or write efficient convolution / self-attention]

- If you really can't afford linear (in number of dimensions) sampling, weaker autoregressive models (log time) such as Parallel PixelCNN are worth considering.
- Notion that autoregressive models are meant for "discrete" is unfounded. Ex: Alex Graves' Handwriting Recognition, Sketch-RNN, World Models, CPC.

- Flow Models are good for modeling densities of continuous valued data and are getting better for discrete (pixels). Larger models needed for complex datasets.
- If you want both representations and sampling or just want to try the simplest thing first, variational auto-encoders with factorized decoders are a natural first choice.

If training density models...

- If you only care about density and not sampling, go for autoregressive models
- If you care about sampling time but not too much, autoregressive is still fine. Just use smaller models and preferably RNNs [or write efficient convolution / self-attention]

- If you really can't afford linear (in number of dimensions) sampling, weaker autoregressive models (log time) such as Parallel PixelCNN are worth considering.
- Notion that autoregressive models are meant for "discrete" is unfounded.
  Ex: Alex Graves' Handwriting Recognition, Sketch-RNN, World Models, CPC.

- Flow Models are good for modeling densities of continuous valued data and are getting better for discrete (pixels). Larger models needed for complex datasets.
- If you want both representations and sampling or just want to try the simplest thing first, variational auto-encoders with factorized decoders are a natural first choice.

If training density models...

- If you only care about density and not sampling, go for autoregressive models
- If you care about sampling time but not too much, autoregressive is still fine. Just use smaller models and preferably RNNs [or write efficient convolution / self-attention]

- If you really can't afford linear (in number of dimensions) sampling, weaker autoregressive models (log time) such as Parallel PixelCNN are worth considering.
- Notion that autoregressive models are meant for "discrete" is unfounded.
  Ex: Alex Graves' Handwriting Recognition, Sketch-RNN, World Models, CPC.

- Flow Models are good for modeling densities of continuous valued data and are getting better for discrete (pixels). Larger models needed for complex datasets.
- If you want both representations and sampling or just want to try the simplest thing first, variational auto-encoders with factorized decoders are a natural first choice.

If training density models...

- If you only care about density and not sampling, go for autoregressive models
- If you care about sampling time but not too much, autoregressive is still fine. Just use smaller models and preferably RNNs [or write efficient convolution / self-attention]

- If you really can't afford linear (in number of dimensions) sampling, weaker autoregressive models (log time) such as Parallel PixelCNN are worth considering.
- Notion that autoregressive models are meant for "discrete" is unfounded.
  Ex: Alex Graves' Handwriting Recognition, Sketch-RNN, World Models, CPC.

- Flow Models are good for modeling densities of continuous valued data and are getting better for discrete (pixels). Larger models needed for complex datasets.
- If you want both representations and sampling or just want to try the simplest thing first, variational auto-encoders with factorized decoders are a natural first choice.

When GANs?

- Cool samples
- Really large images and HQ datasets like faces, buildings, etc.
- Class-conditional models
- Image to Image translation problems (edges / seg-map to real image, adding texture, adding color, etc.)
- If you care only about perceptual quality and want controllable generation and don’t have lot of compute, GAN is the best choice.

When GANs?

- Cool samples
- Really large images and HQ datasets like faces, buildings, etc.
- Class-conditional models
- Image to Image translation problems (edges / seg-map to real image, adding texture, adding color, etc.)
- If you care only about perceptual quality and want controllable generation and don’t have lot of compute, GAN is the best choice.

When GANs?

- Cool samples
- Really large images and HQ datasets like faces, buildings, etc.
- Class-conditional models
- Image to Image translation problems (edges / seg-map to real image, adding texture, adding color, etc.)
- If you care only about perceptual quality and want controllable generation and don’t have lot of compute, GAN is the best choice.

When GANs?

- Cool samples
- Really large images and HQ datasets like faces, buildings, etc.
- Class-conditional models
- Image to Image translation problems (edges / seg-map to real image, adding texture, adding color, etc.)
- If you care only about perceptual quality and want controllable generation and don't have lot of compute, GAN is the best choice.

When GANs?

- Cool samples
- Really large images and HQ datasets like faces, buildings, etc.
- Class-conditional models
- Image to Image translation problems (edges / seg-map to real image, adding texture, adding color, etc.)
- If you care only about perceptual quality and want controllable generation and don't have lot of compute, GAN is the best choice.

When GANs?

- Cool samples
- Really large images and HQ datasets like faces, buildings, etc.
- Class-conditional models
- Image to Image translation problems (edges / seg-map to real image, adding texture, adding color, etc.)
- If you care only about perceptual quality and want controllable generation and don’t have lot of compute, GAN is the best choice.

TABLE 1: Comparison between deep generative models in terms of training and test speed, parameter efficiency, sample quality, sample diversity, and ability to scale to high resolution data. Quantitative evaluation is reported on the CIFAR-10 dataset [114] in terms of Fréchet Inception Distance (FID) and negative log-likelihood (NLL) in bits-per-dimension (BPD).

Method	Train Speed	Sample Speed	Param. Effic.	Sample Quality	Relative Divers.	Resolution Scaling	FID	NLL (in BPD)
Generative Adversarial Networks								
DCGAN [169]	★★★★★	★★★★★	★★★★☆	★★★☆☆	★☆☆☆☆	★★☆☆☆	17.70	-
ProGAN [102]	★★☆☆☆	★★★★★	★★★★☆	★★★☆☆	★★☆☆☆	★★★★★	15.52	-
BigGAN [17]	★★★★☆	★★★★★	★★★★☆	★★★★☆	★☆☆☆☆	★★★★☆	14.73	-
StyleGAN2 + ADA [103]	★★★☆☆	★★★★★	★★★★☆	★★★★☆	★★☆☆☆	★★★★★	2.42	-
Energy Based Models								
IGEBM [42]	★★★☆☆	★★☆☆☆	★★★★★	★★★☆☆	★★★☆☆	★★★☆☆	37.9	-
Denoising Diffusion [80]	★★☆☆☆	★★☆☆☆	★★★☆☆	★★★★☆	★★★★☆	★★★★☆	3.17	$\leq$ 3.75
DDPM++ Continuous [191]	★★☆☆☆	★★★☆☆	★★★☆☆	★★★★☆	★★★★☆	★★★★☆	2.92	2.99
Flow Contrastive [51]	★☆☆☆☆	★★★☆☆	★☆☆☆☆	★★☆☆☆	★★★☆☆	★★☆☆☆	37.30	$\approx$ 3.27
VAEBM [226]	★★☆☆☆	★★★☆☆	★★★☆☆	★★★☆☆	★★★☆☆	★★★★☆	12.19	-
Variational Autoencoders								
Convolutional VAE [110]	★★★★★	★★★★★	★★★★☆	★★☆☆☆	★★★★☆	★★☆☆☆	106.37	$\leq$ 4.54
Variational Lossy AE [27]	★☆☆☆☆	★☆☆☆☆	★☆☆☆☆	★★☆☆☆	★★★★☆	★☆☆☆☆	-	$\leq$ 2.95
VQ-VAE [171], [215]	★★☆☆☆	★★★☆☆	★★☆☆☆	★★★☆☆	★★★★☆	★★★★★	-	$\leq$ 4.67
VD-VAE [29]	★★★☆☆	★★★★☆	★★★☆☆	★★★☆☆	★★★★☆	★★★★★	-	$\leq$ 2.87
Autoregressive Models								
PixelRNN [214]	★☆☆☆☆	★☆☆☆☆	★☆☆☆☆	★☆☆☆☆	★★★★★	★★★☆☆	-	3.00
Gated PixelCNN [213]	★☆☆☆☆	★☆☆☆☆	★☆☆☆☆	★★☆☆☆	★★★★★	★★★☆☆	65.93	3.03
PixelIQN [161]	★☆☆☆☆	★☆☆☆☆	★☆☆☆☆	★★☆☆☆	★★★★★	★★★☆☆	49.46	-
Sparse Trans. + DistAug [30], [99]	★★☆☆☆	★☆☆☆☆	★★☆☆☆	★★★★☆	★★★★★	★★★☆☆	14.74	2.66
Normalizing Flows								
RealNVP [39]	★☆☆☆☆	★★★★☆	★☆☆☆☆	★☆☆☆☆	★★★★★	★★★☆☆	-	3.49
Masked Autoregressive Flow [165]	★★☆☆☆	★☆☆☆☆	★★☆☆☆	★☆☆☆☆	★★★☆☆	★★☆☆☆	-	4.30
GLOW [111]	★☆☆☆☆	★★★★☆	★☆☆☆☆	★★☆☆☆	★★★★★	★★★★☆	45.99	3.35
FFJORD [56]	★☆☆☆☆	★★★☆☆	★★★★★	★★☆☆☆	★★★★★	★★☆☆☆	-	3.40
Residual Flow [24]	★★☆☆☆	★★★★☆	★★★★☆	★★☆☆☆	★★★★★	★★★★☆	46.37	3.28

[Bond-Taylor et al. 2021]

Next lecture:
Self-Supervised Learning