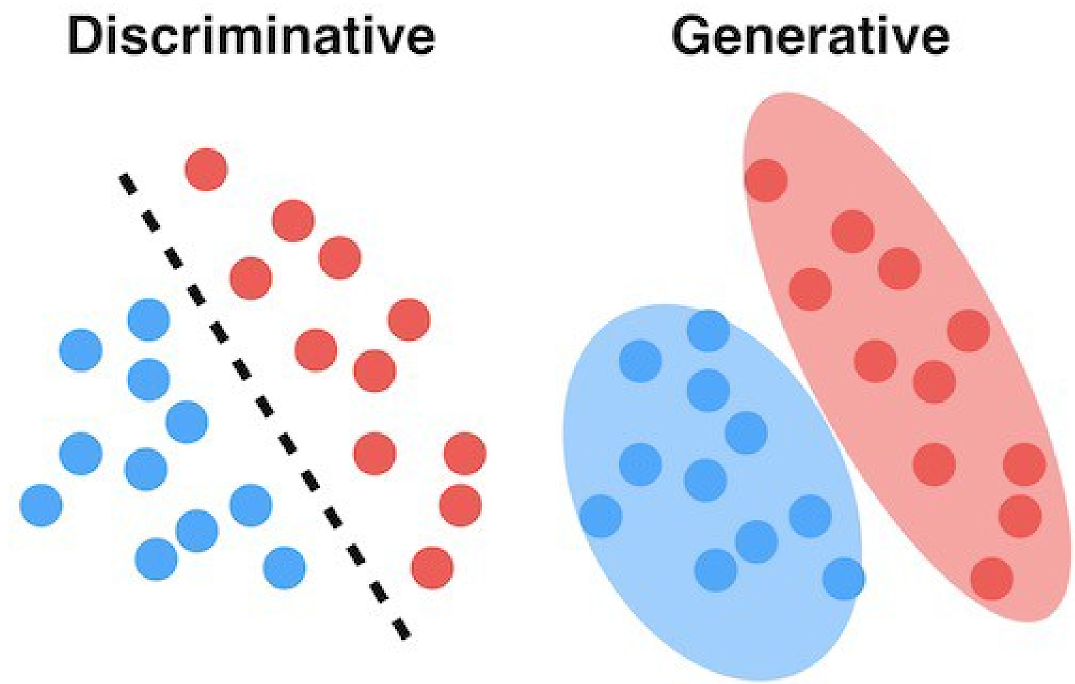
What is a “generative model”?

- Discriminative
  - Logistic Regression
  - Support Vector Machines
  - Random Forests

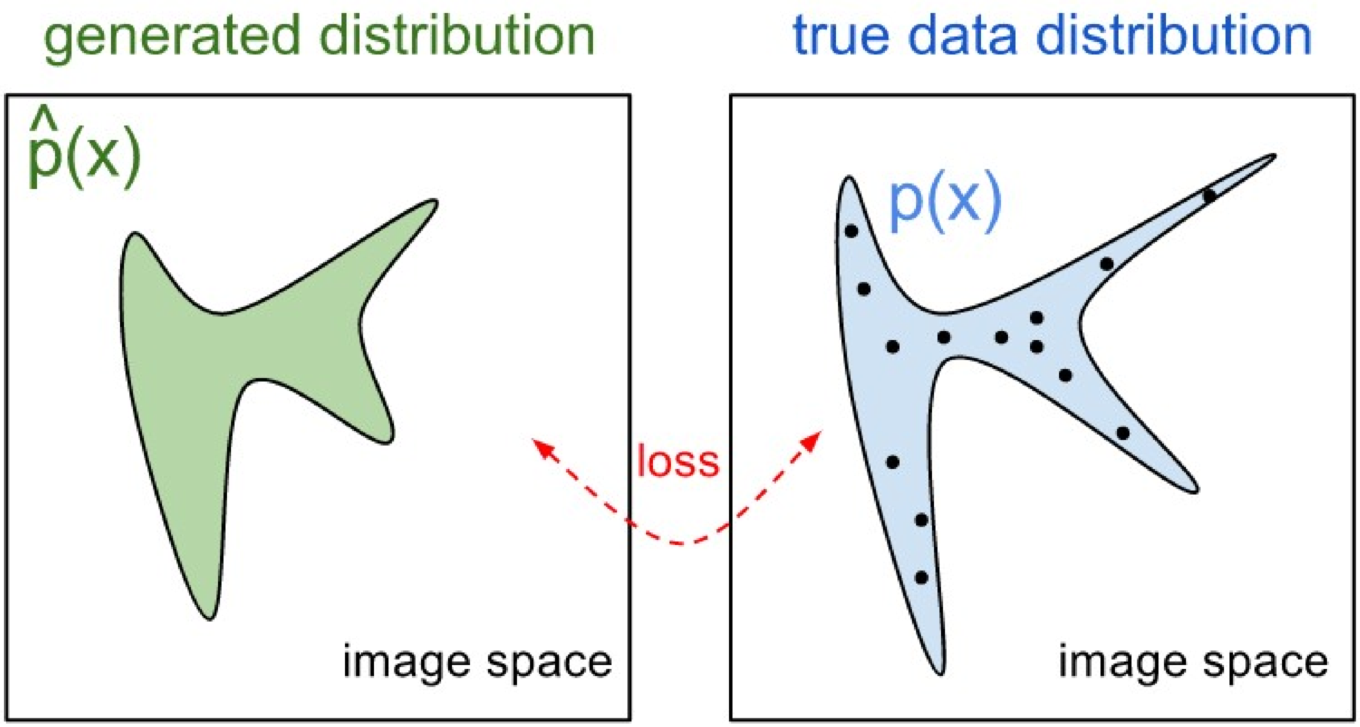
P(Y | X)

- Generative
  - Gaussian Naïve Bayes
  - Variational Autoencoders
  - Adversarial Networks

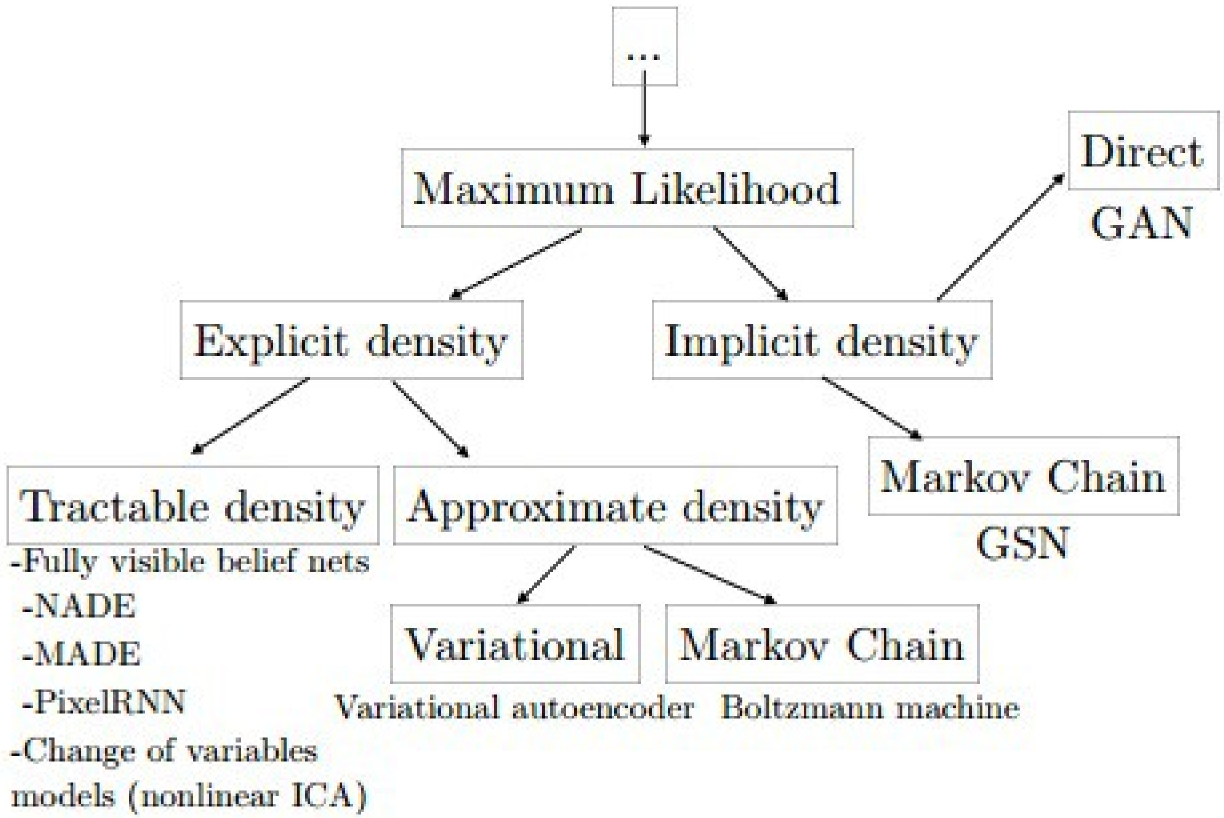
P(X, Y) and P(Y)



Generative Models

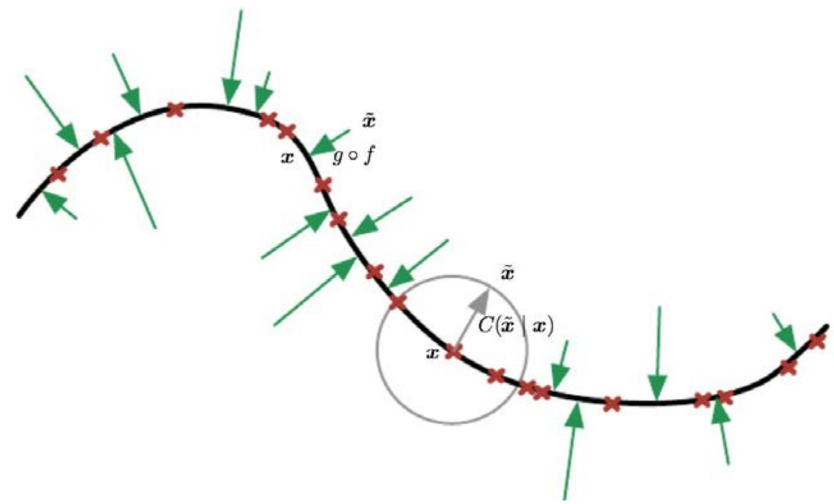


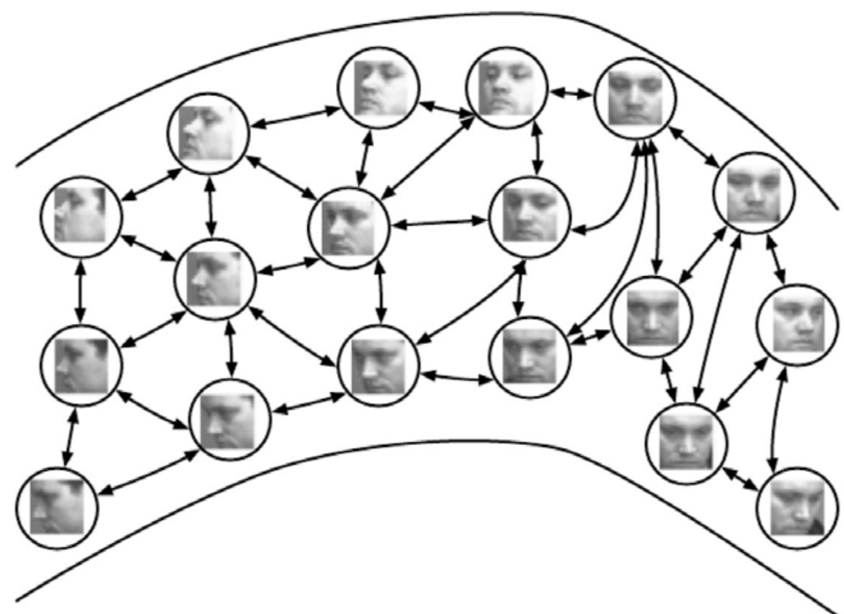
Generative Models



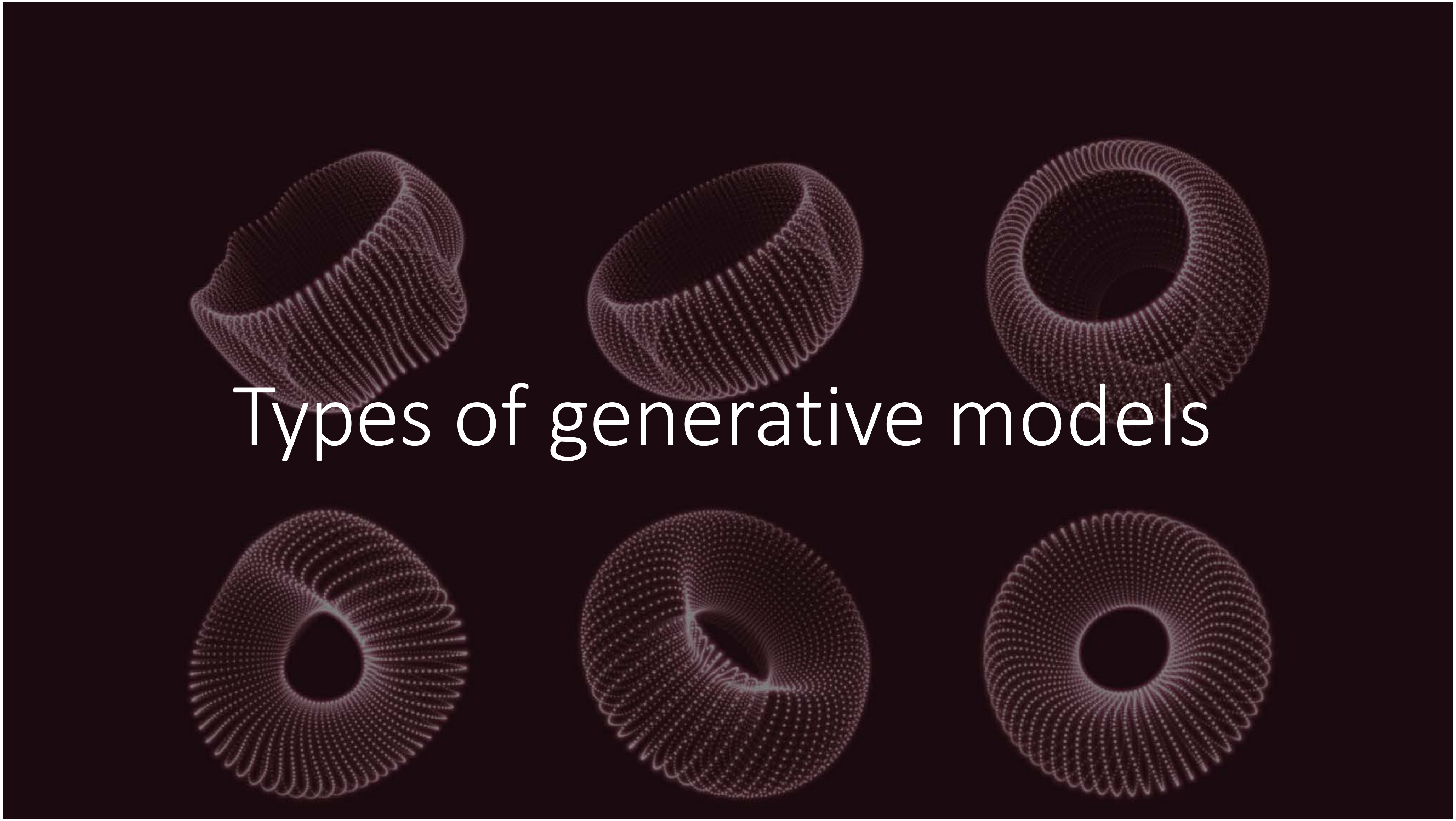
Generative Models

- Learning a *distribution* or *manifold*
  - Statistical notion of *how the data were generated*
- P(X) asks: how *likely* is the data point X?
  - If likely -> X was **generated** by this process
- Compare to P(Y), which asks: how likely is the *label*?
  - If likely -> X has label Y





Types of generative models



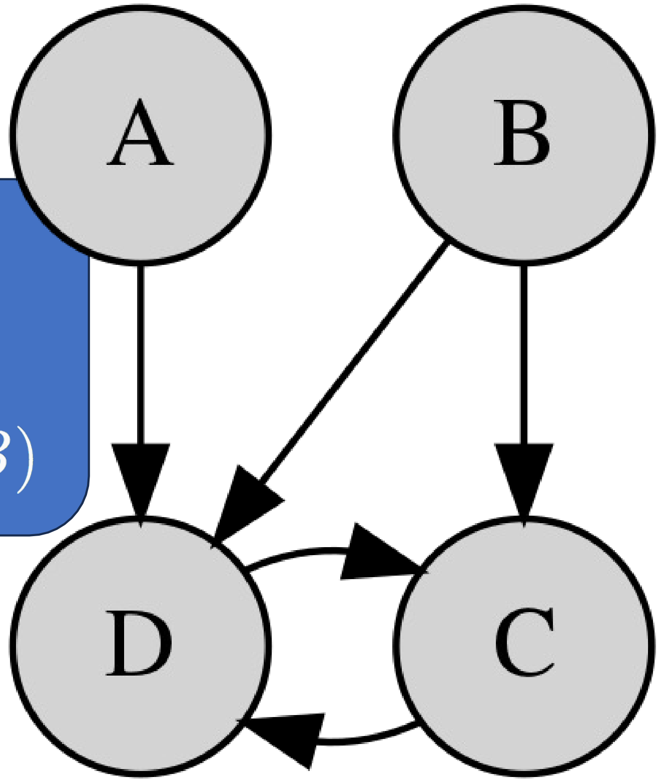
Probabilistic Graphical Models

- Arrows represent conditional dependencies between random variables

$$P(X_1, ..., X_n) = \prod_{i=1}^{n} P(X_i | \text{parents}_i)$$

$$P(A, B, C, D) = P(A)P(B)P(C, D | A, B)$$

- Structure is used in generative models
  - Latent generating distribution (hidden)
  - Observed variables (influenced by latent vars)



Variational Inference

- What is variational inference?
- Good for learning latent variable models (i.e., generating distributions of data)
- For each observation *x* we assign a hidden variable *z*; our model *p* describes the joint distribution between *x* and *z*

$p_\theta(z)$ is very easy 🐣,
$p_\theta(x|z)$ is easy 🐹,
$p_\theta(x,z)$ is easy 🐨,
$p_\theta(x)$ is super-hard 🐍,
$p_\theta(z|x)$ is mega-hard 🐲

Of course these are the things we want to calculate

- Inference is p(z|x)
- Learning involves p(x)

Variational Inference

- Rather than learning *p(z|x)* directly, variational inference approximates with *q(z|x)*
- Maximize the evidence lower bound (ELBO)

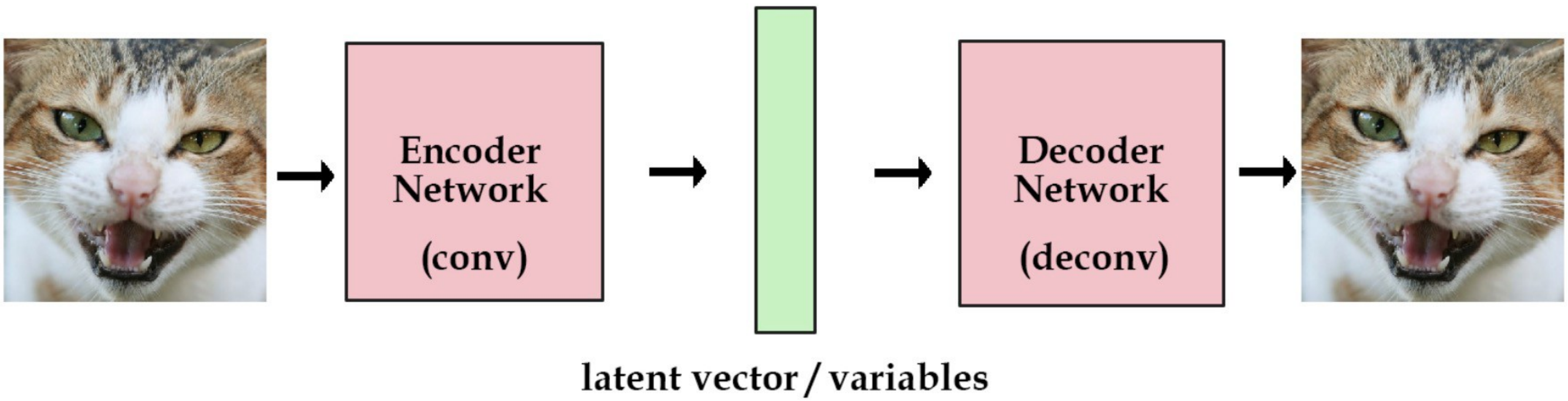
$$\mathrm{ELBO}(\theta, \psi) = \sum_n \log p(x_n) - \mathrm{KL}\left[q_\psi(z|x_n) || p_\theta(z|x_n)\right]$$

- This can be written in terms of the "friendly" emojis

$p_\theta(z)$ is very easy 🐣,
$p_\theta(x|z)$ is easy 🐹,
$p_\theta(x, z)$ is easy 🐨,
$p_\theta(x)$ is super-hard 🐍,
$p_\theta(z|x)$ is mega-hard 🐲

$$\text{💪} = -\sum_n \mathbb{E}_{\text{🐰}} \log \frac{\text{🐰}}{\text{🐨}} + \text{constant}$$
$$= \sum_n \mathbb{E}_{\text{🐰}} \log \text{🐹} - \sum_n \mathbb{E}_{\text{🐰}} \mathrm{KL}[\text{🐰} || \text{🐣}]$$

Recall: Autoencoders



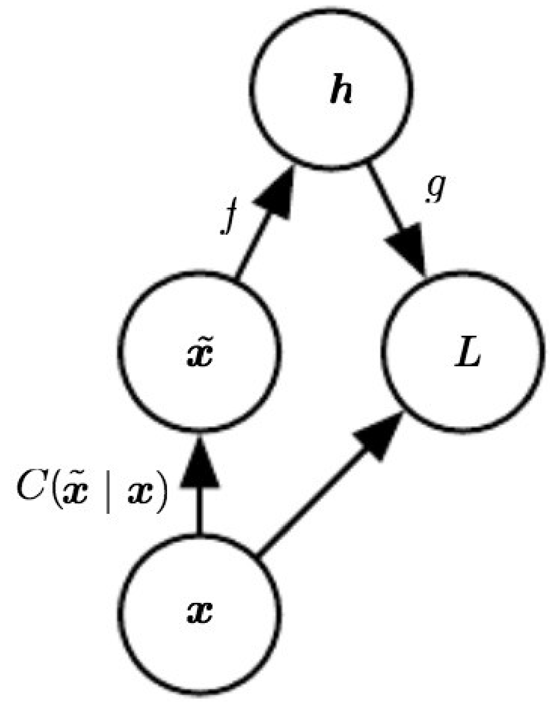
Denoising Autoencoders

- Define a corruption process, *C*

$$C(\tilde{x} | x)$$

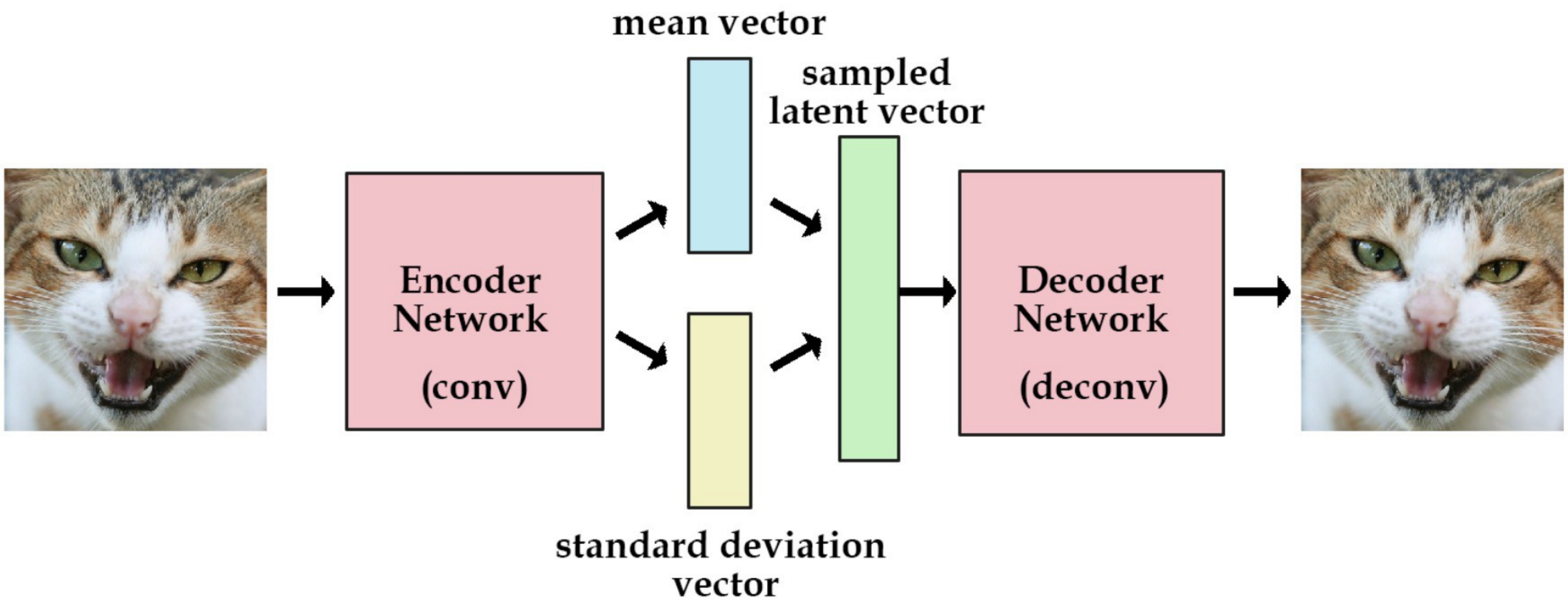
- Autoencoder learns a *reconstruction distribution* $p_{reconstruct}(x \mid \tilde{x})$

1. Sample a training example *x*
2. Sample a corrupted version $\tilde{x}$ from *C*
3. Use $(x, \tilde{x})$ as a training pair



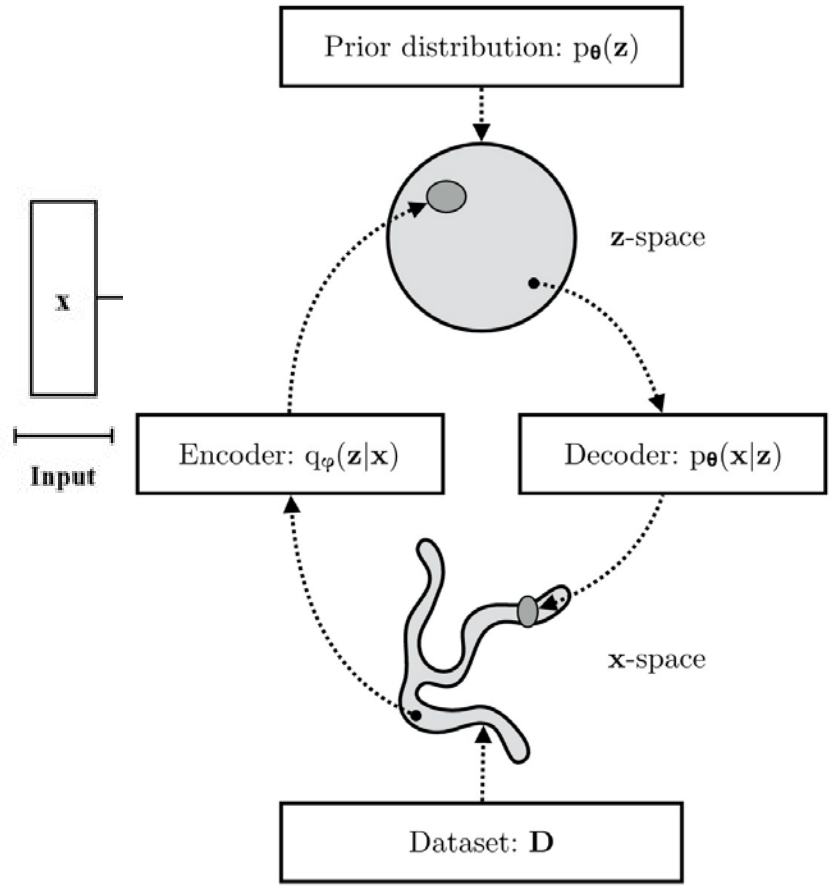
Denoising Autoencoders

- De-corruption process results in learning a *distribution*



Variational Autoencoders (VAEs)

- Associated with autoencoders by virtue of architecture
  - Goal is to map inputs to latent space
- Encoder: Learn parameters of variational distribution, *q(z | x)*
- Decoder: Sample (generate!) from learned distribution to reconstruct input

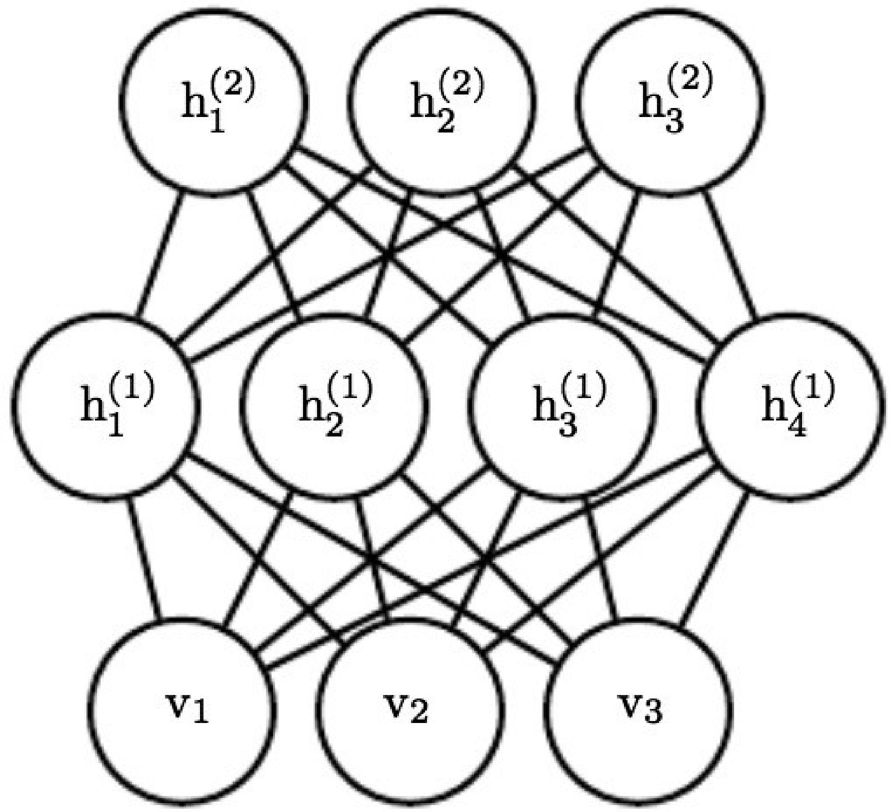


Restricted Boltzmann Machines (RBMs)

- Wholly undirected deep network
  - Implementation of a probabilistic graphical model
  - Each variable conditionally independent given neighboring nodes
- Parameterized by energy function

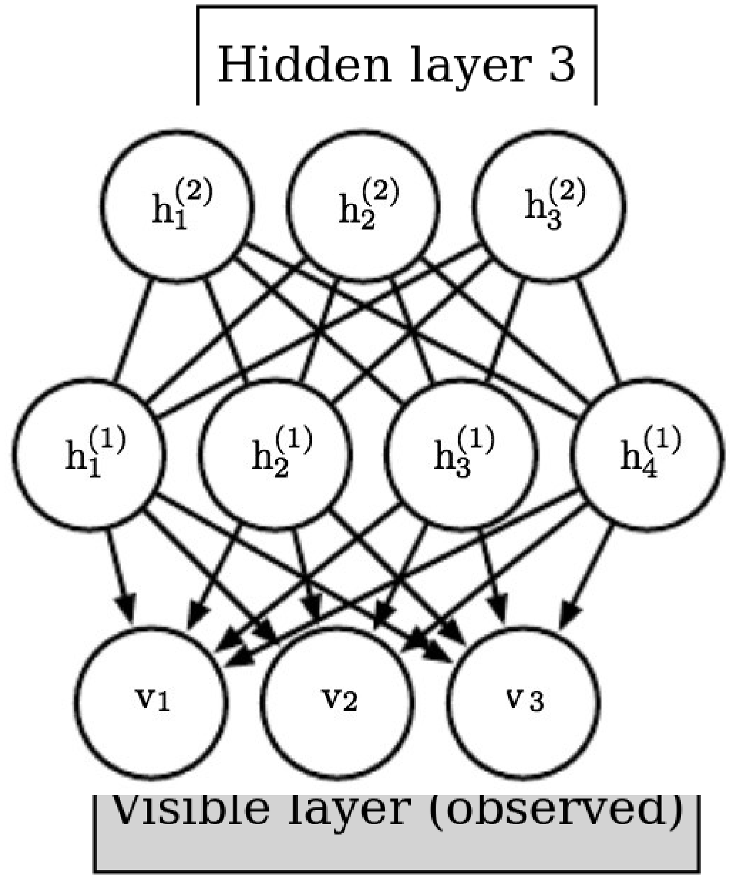
$$P\left(\boldsymbol{v}, \boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}, \boldsymbol{h}^{(3)}\right) = \frac{1}{Z(\boldsymbol{\theta})} \exp\left(-E(\boldsymbol{v}, \boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}, \boldsymbol{h}^{(3)}; \boldsymbol{\theta})\right)$$

hard, but training is paradoxically easy

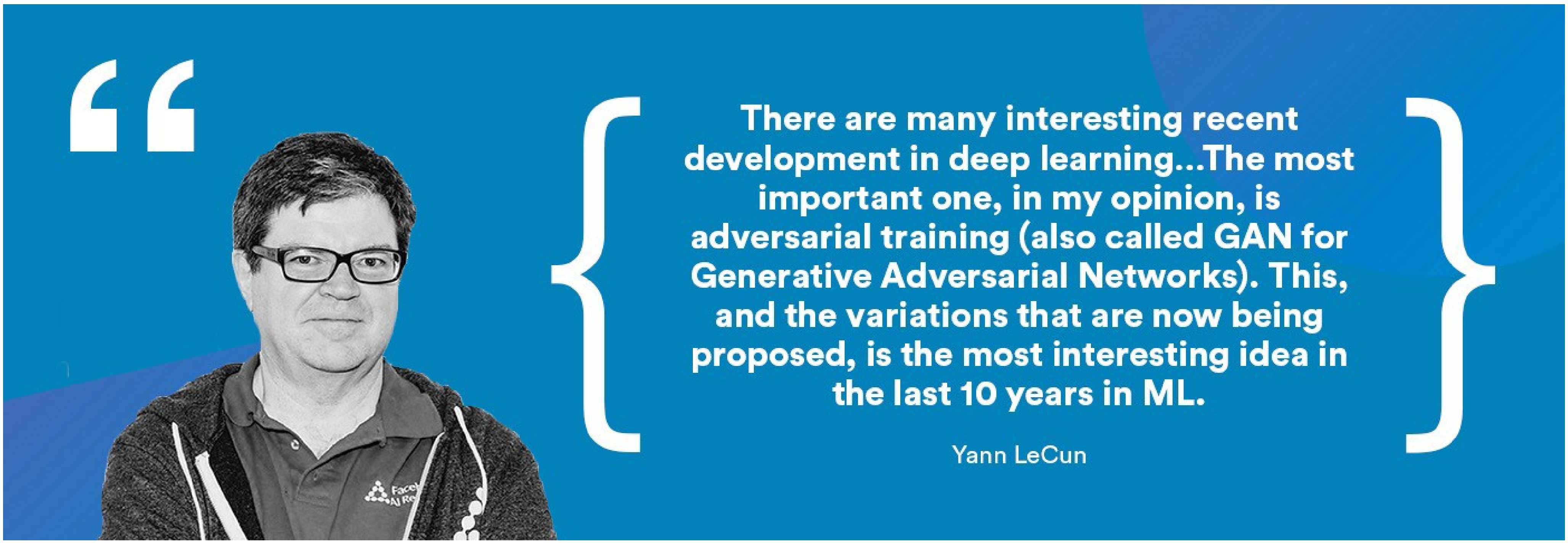


Deep Belief Nets (DBNs)

- Connections *between* layers, but not units *within* a layer
- Arguably one of the first successful applications of modern deep learning
  - Hinton 2006 and 2007
- Often built from an RBM template
- Training is nearly intractable
  - Posterior has to be approximated through annealed importance sampling (AIS)

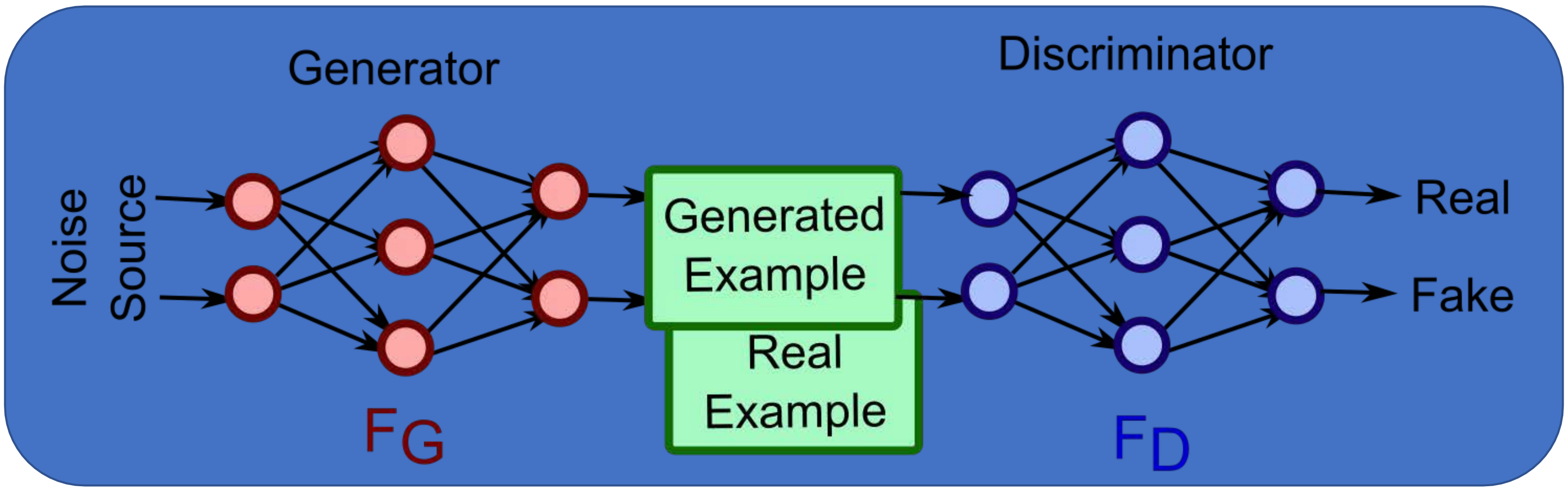


Generative Adversarial Networks (GANs)



GANs

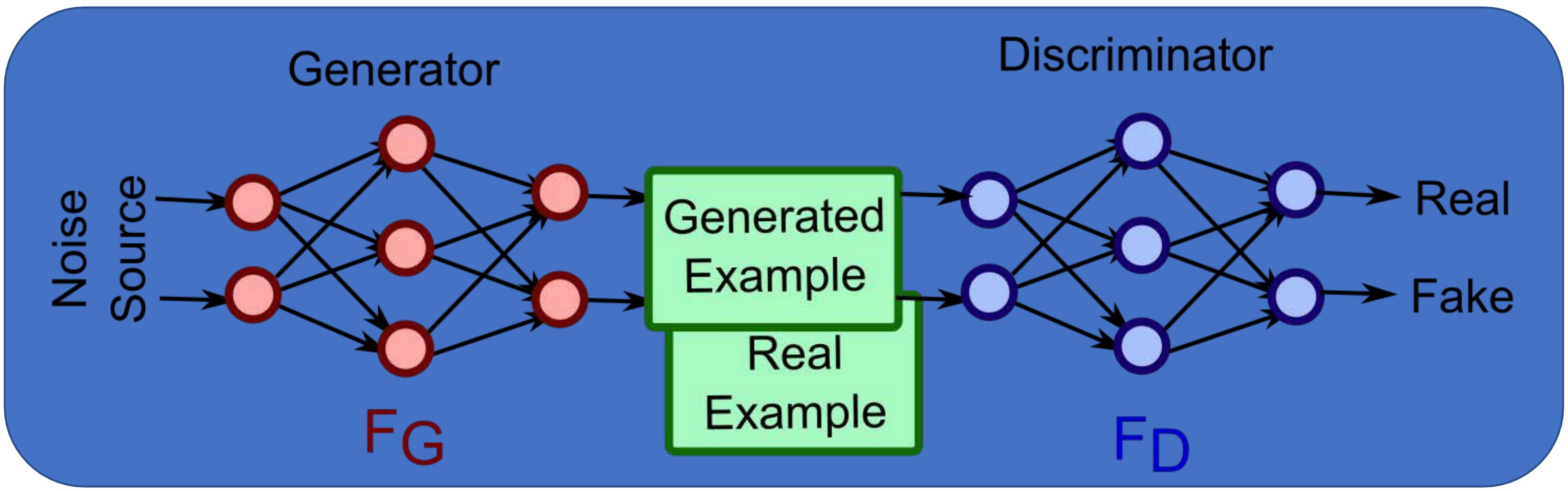
- Game-theoretic approach to generative modeling
- Two deep networks: a **generator** (G) and **discriminator** (D)



GANs

- **Generator**
- Input: a random vector *z*
- Output: something as close to a "real" data point as possible

- **Discriminator**
- Input: a "real" data point OR a synthetic example from *G*
- Output: 1 or 0 (real or fake)



GANs

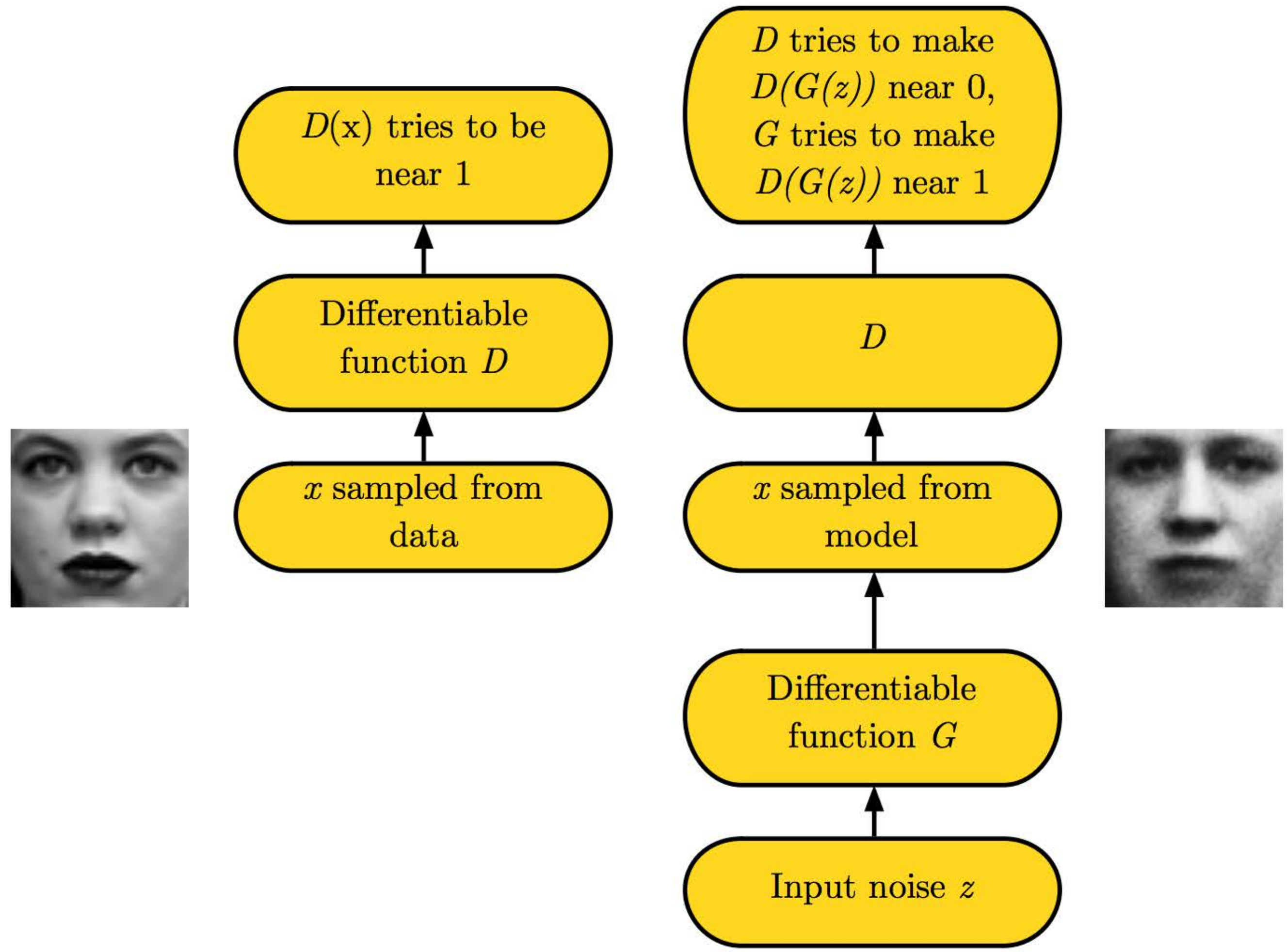
- Minimax "game"
  - Generator and Discriminator have competing objectives
  - Goal is to find an equilibrium point

$$\min_G \max_D \mathbb{E}_{x \sim P_{real}} \log D(x) - \mathbb{E}_z \log(1 - D(G(z)))$$

Maximize the Discriminator's likelihood of identifying a real data example

Minimize the Discriminator's ability to differentiate real data from Generator examplars

GANs

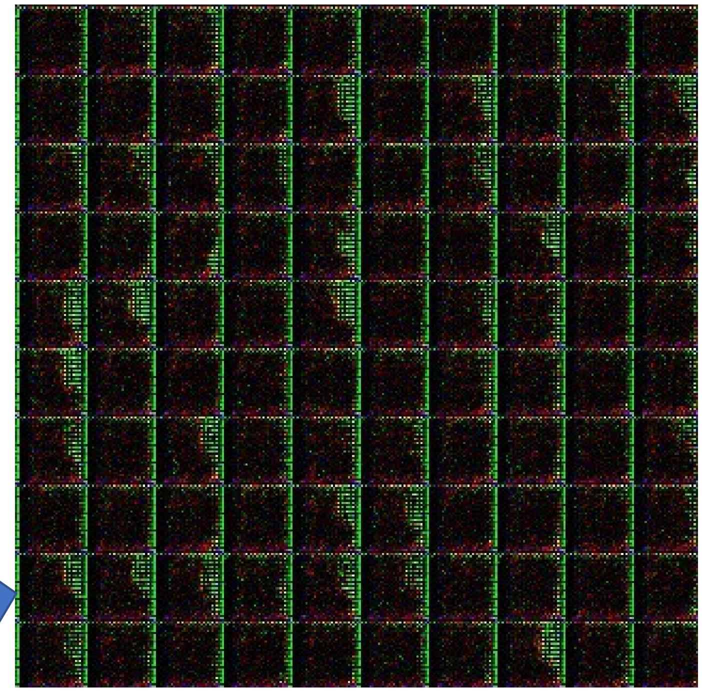


D(x) tries to be near 1
Differentiable function D
x sampled from data
D tries to make $D(G(z))$ near 0, G tries to make $D(G(z))$ near 1
D
x sampled from model
Differentiable function G
Input noise z

VAEs versus GANs

VAEs
Expectation over learned distribution results in blurring

GANs
Samples from learned distribution, resulting in sharper images

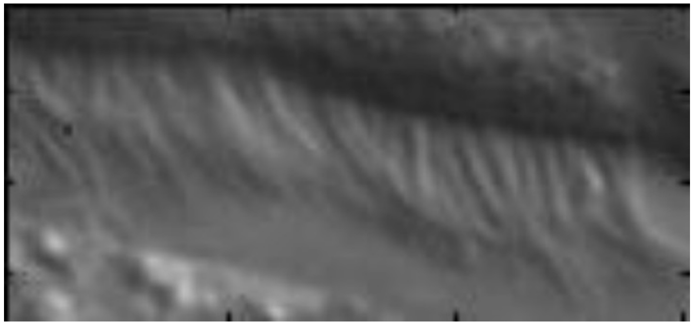


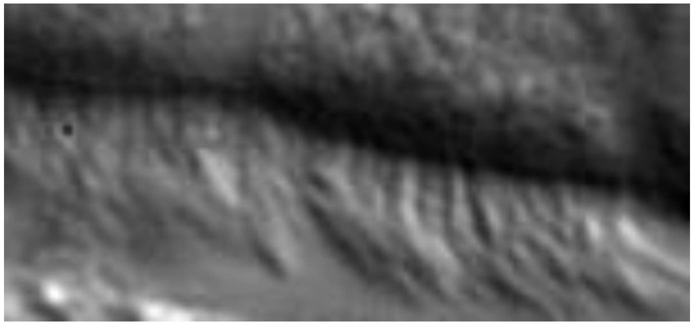
Autoregressive (AR) Models

- DALLE-1, in January 2021, was an autoregressive Transformer
- Our good friends, ~~Thing 1 and Thing 2~~ Appearance and State

$$y_t = Cx_t + u_t$$
$$x_t = Ax_{t-1} + Wv_t$$

- Once you've learned A_i, you can generate new x_t!

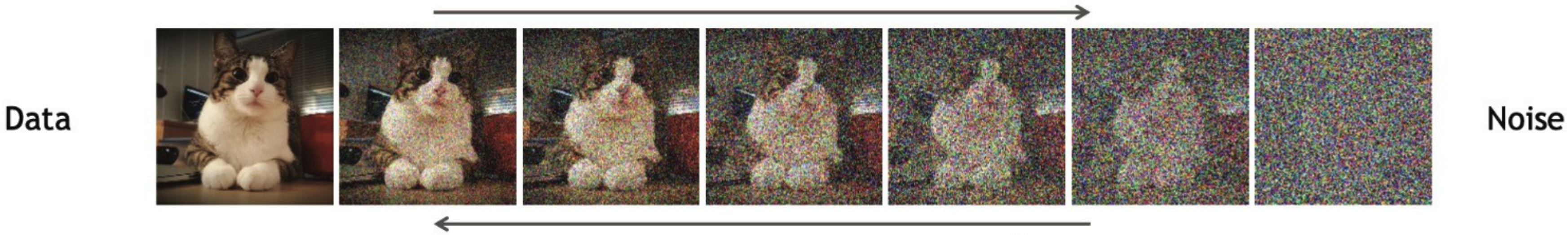




Latent Diffusion

- Closely related to VAEs, normalizing flows, and energy-based models
- **Hard** to convert noise into structured data
- **Easy** to convert structured data into noise

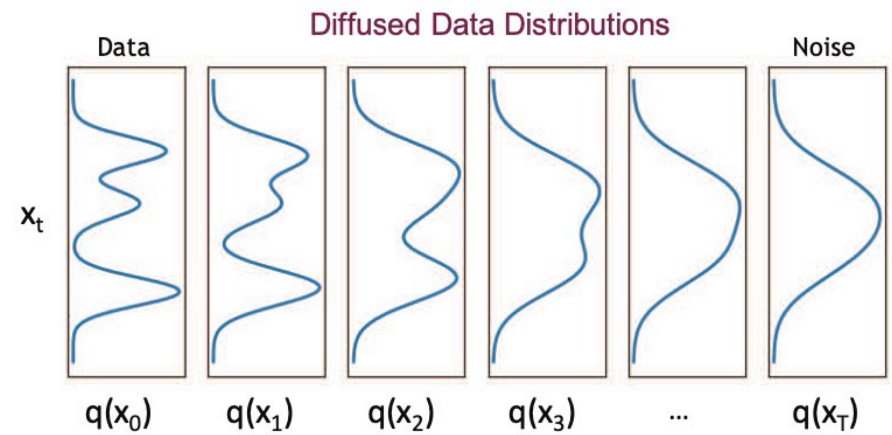
$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x_t}; \sqrt{1-\beta_t}\mathbf{x_{t-1}}, \beta_t\mathbf{I})$$



$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I})$$

Latent Diffusion

- Similar to hierarchical VAE
  - BUT all latent states have same dimensionality as input
  - BUT encoder is a linear Gaussian model, rather than being learned
- Results in a very simple objective
  - No risk of posterior collapse (unlike GANs)
- Numerous variations of LD
  - Denoising Diffusion Probabilistic Models (DDPM)
  - Noise-conditioned Score Networks (NCSN)
  - Stochastic Differential Equations (SDE)



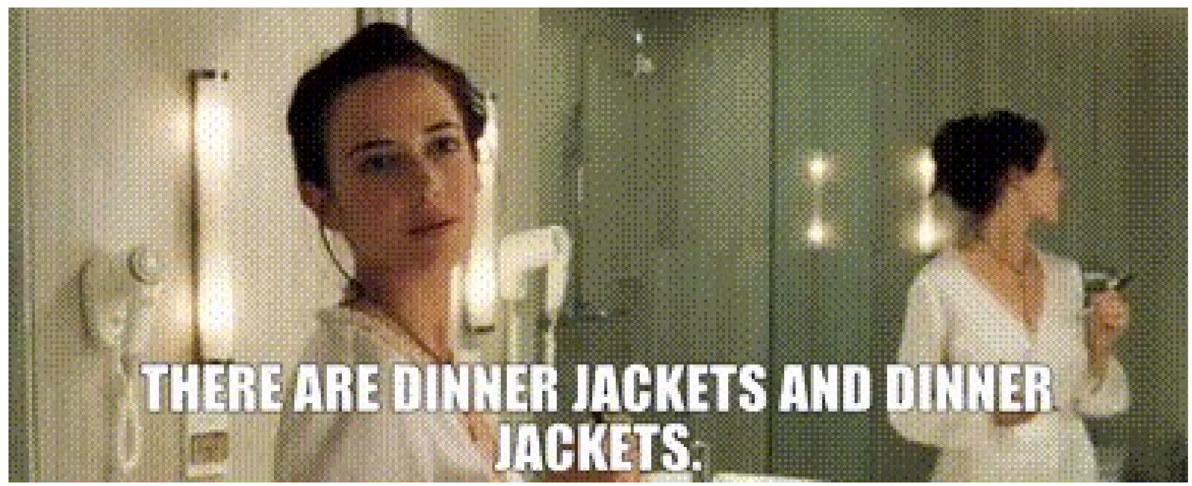
Large Language Models (LLMs)

- Not unique generative models *per se*
  - LLMs = very, very large Transformers
  - Usually with autoregressive blocks at inference / decoding
  - Trained on city blocks' worth of GPUs
- Sometimes called "Foundation Models"
  - "Foundation Models" are only found on Terminus; elsewhere in the galaxy, they're just "Sparkling Language Models"

Commercial generative models

Probably screaming into the void here, but…

- There's AI, and there's AI
- AI
  - Large language or image models
  - Trained on massive amounts of data with large numbers of parameters
  - Does a frighteningly good job of mimicking humans at very specific tasks
  - **Not intelligent**
- AI
  - Intelligence that isn't human but made by humans, aka artificial
  - Mimics humans very well at *all possible tasks, even those it wasn't trained on*
  - Nowhere in the 5-10 year roadmap

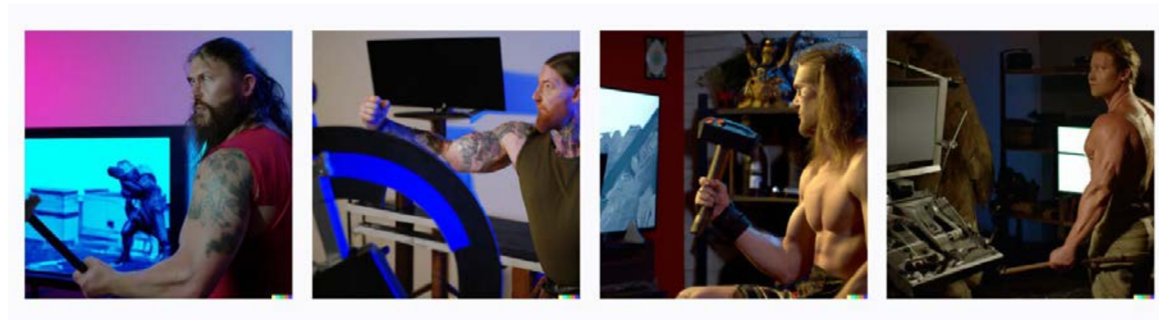


DALL-E, Midjourney, Stable Diffusion, Firefly

- Corporate backed text-to-image generators
- Subscription fees
- Open source options
- Training data

Evolution of Image generators

- DALL-E
- “a muscular barbarian with weapons beside a CRT television set, cinematic, 8K, studio lighting.”
- April 2022
- October 2023



https://arstechnica.com/information-technology/2023/11/from-toy-to-tool-dall-e-3-is-a-wake-up-call-for-visual-artists-and-the-rest-of-us/

GPT-4

- Powers ChatGPT

"Attention is All You Need", 2017

- A "Transformer-style model pre-trained to predict the next token in a document, using both publicly available data (such as internet data) and data licensed from third-party providers."
- "The model was then fine-tuned using Reinforcement Learning from Human Feedback (RLHF)."

"Deep reinforcement learning from human preferences", 2017

- Several thousand GPUs + petabytes of data = ChatGPT

PaLM, Cerebras, LLaMA, Falcon, OpenHermes

- Similar underlying architecture to ChatGPT
- Billions (to trillions?) of parameters
  - GPT-4 rumored to have ~2T parameters (source: SoC Day keynote speaker)
- Billions to trillions of training tokens
  - PaLM 2 and LLaMA 2: 3.6T and 2T, respectively
- Varying levels of openness
  - Some pre-trained models on Huggingface
  - An open LLM + RLHF (reinforcement learning from human feedback) + RLAIF (reinforcement learning from AI feedback) + DPO (direct preference optimization) = best bang for buck, outside of ChatGPT or similar

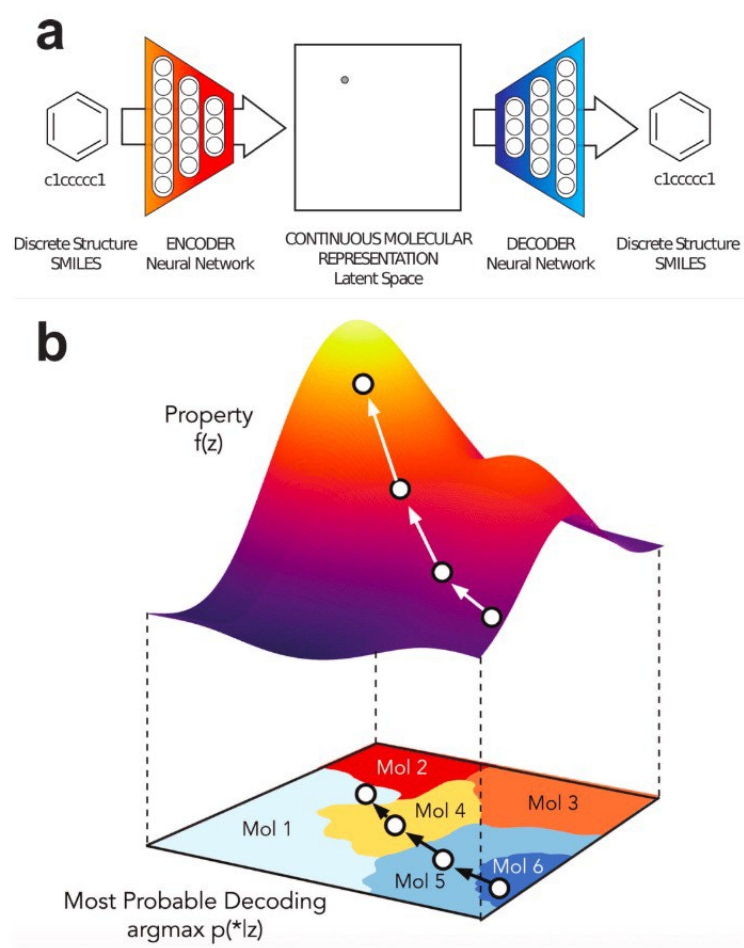
Advantages of Generative AI

- Already legion!

- Democratize access to art and figure generation
- Interactive, natural-language interfaces
  - As opposed to arcane tricks and query optimization hacks with traditional search engines
- Revealed clear weaknesses in our assessment protocols
  - Educational assessment (i.e., grading) should not be contingent on whether or not you had access to a chatbot

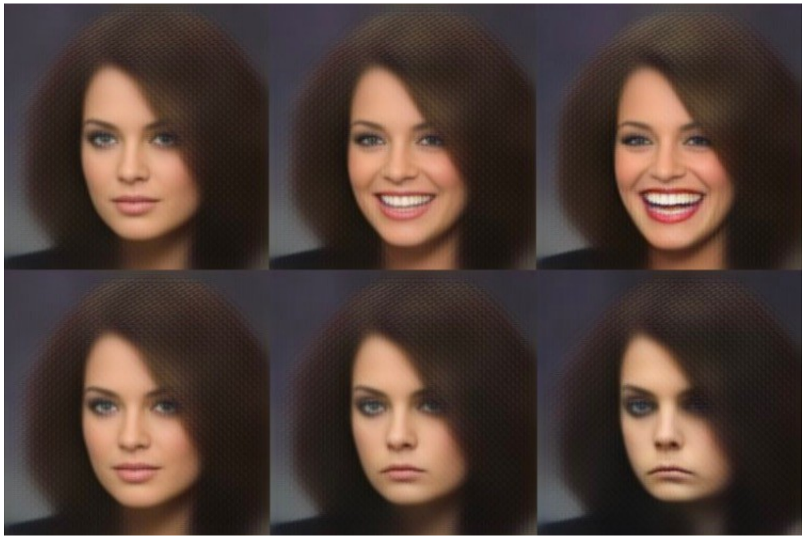
Advantages of Generative AI

- New scientific discoveries around medicine, biology, chemistry, and biochemistry
- Design new compounds (drugs, antibiotics, treatments) by teaching generative models about known ones
- Keynote speaker at IOB Symposium in October spoke about using LLMs to discover new proteins!



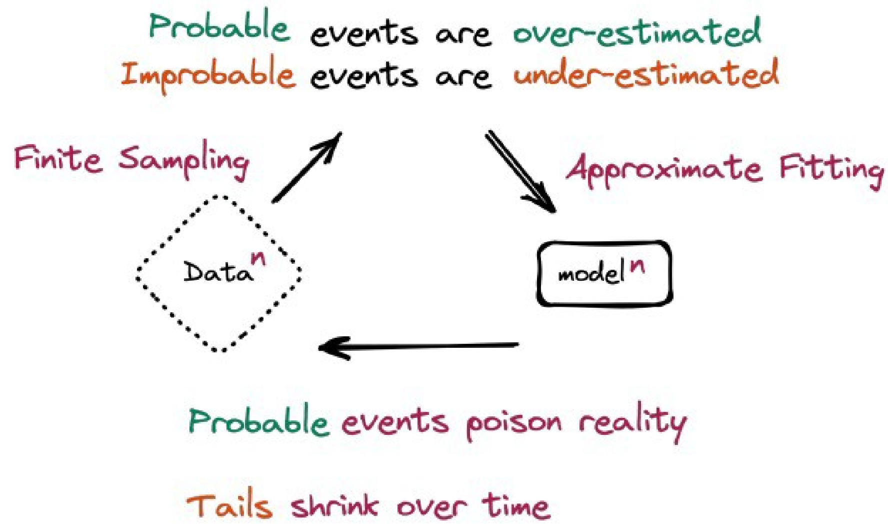
Advantages of Generative AI

- Accessibility and interactivity
- Original image (top left) interpolated along VAE latent distribution, producing different facial expressions
- Virtual avatars, assistants, video gaming



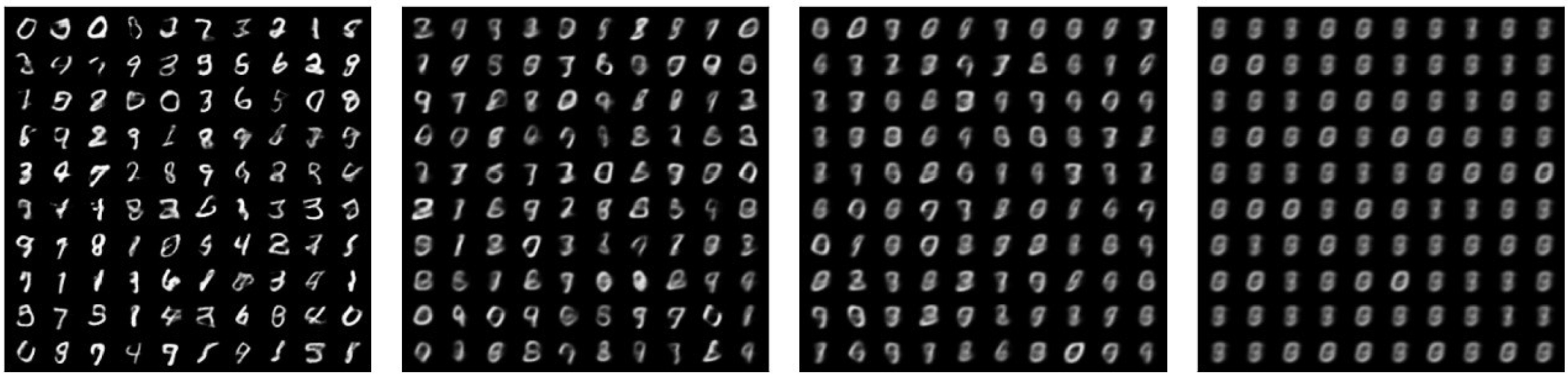
Technical issues

- Recursive model training
  - As more information on the internet (images, text) is AI-generated, LLMs will ingest this data as part of their training, creating a recursive training loop
  - “The Curse of Recursion: Training on Generated Data Makes Models Forget”
  - https://arxiv.org/abs/2305.17493



Technical issues

- Examples of recursive model training



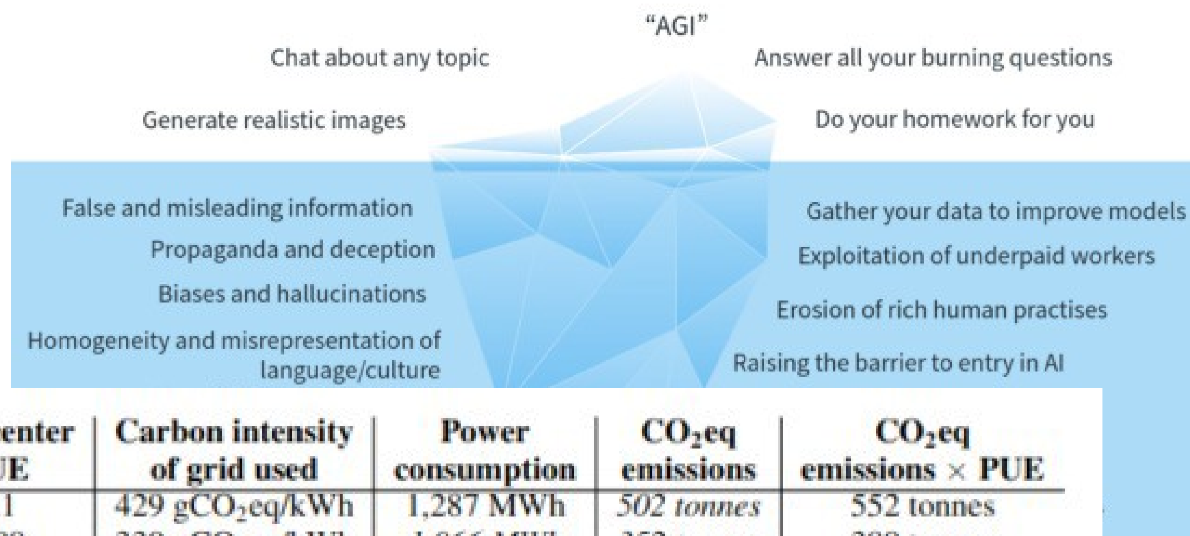
(a) Original model (b) Generation 5 (c) Generation 10 (d) Generation 20

Legal issues

- Copyright
  - OpenAI, Midjourney most likely training on image datasets **without** permission from authors
  - Currently in the US, AI-generated art cannot be copyrighted à **potential boon for public domain!**
- Plagiarism
  - Simply: if you didn’t write/code/create it **yourself**, and you didn’t otherwise specify where it came from (and sometimes, even if you did), it’s **plagiarism**
  - Is getting the answer from ChatGPT and presenting it as your own any different from getting the answer from your classmate and presenting it as your own?
  - Huge implications in professional fields, given current chatbot accuracy levels

Ethical and moral issues

- Disinformation
- Enabling/scaling abuse
- Environmental concerns
- Worker exploitation
- Hidden costs of AI



Model name	Number of parameters	Datacenter PUE	Carbon intensity of grid used	Power consumption	CO_2eq emissions	CO_2eq emissions × PUE
GPT-3	175B	1.1	429 gCO_2eq/kWh	1,287 MWh	*502 tonnes*	552 tonnes
Gopher	280B	1.08	330 gCO_2eq/kWh	*1,066 MWh*	*352 tonnes*	380 tonnes
OPT	175B	*1.09*[2]	*231 gCO_2eq/kWh*	*324 MWh*	70 tonnes	*76.3 tonnes*[3]
BLOOM	176B	1.2	57 gCO_2eq/kWh	433 MWh	25 tonnes	30 tonnes

Table 4: Comparison of carbon emissions between BLOOM and similar LLMs. Numbers in *italics* have been inferred based on data provided in the papers describing the models.

Philosophical issues

- Novelty
  - Is the content generated from ChatGPT / Midjourney **new**?
- The “tool” analogy
  - Generative AI is inherently neither good nor bad, but dependent on its application

- In 1999, French cultural theorist Paul Virilio wrote, "*When you invent the ship, you also invent the shipwreck; when you invent the plane you also invent the plane crash; and when you invent electricity, you invent electrocution... Every technology carries its own negativity, which is invented at the same time as technical progress*."

Conclusions

- Generative modeling
  - Learn a *distribution* instead of a decision boundary
  - Can still be used for classification
  - Usually requires more data than discriminative models
- Deep generative modeling
  - DBNs, RBMs, Denoising & Variational Autoencoders, GANs, AR models, LD
  - All ways of learning a generating distribution from data in deep neural architectures
- Deployments of generative AI
  - Commercial products (ChatGPT, Stable Diffusion, Midjourney, DALL-E)
  - Possibilities, advantages, moral/ethical/legal/philosophical considerations
  - Consider the possible use-cases

References

- “Generative Learning algorithms”, CS 229 notes by Andrew Ng http://cs229.stanford.edu/notes/cs229-notes2.pdf
- Andrew Ng and Michael I. Jordan (!!!), “On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes” (NIPS 2002) http://papers.nips.cc/paper/2020-on-discriminative-vs-generative-classifiers-a-comparison-of-logistic-regression-and-naive-bayes.pdf
- “Generative versus Discriminative”, Cross-Validated https://stats.stackexchange.com/questions/12421/generative-vs-discriminative/223850
- “Generative Adversarial Networks”, NIPS 2016 Tutorial https://arxiv.org/pdf/1701.00160.pdf
- Variational Inference http://www.inference.vc/choice-of-recognition-models-in-vaes-a-regularisation-view/
- Tutorial on Variational Autoencoders https://arxiv.org/abs/1606.05908
- “Progressive Growing of GANs for Improved Quality, Stability, and Variation”, https://arxiv.org/abs/1710.10196
- “Generative Adversarial Networks (GANs), Some Open Questions”, http://www.offconvex.org/2017/03/15/GANs/
- “GANs are Broken in More than One Way: The Numerics of GANs”, http://www.inference.vc/my-notes-on-the-numerics-of-gans/
- “Diffusion Models in Vision: A Survey”, https://ieeexplore.ieee.org/iel7/34/4359286/10081412.pdf
- “GPT-4 Technical Report”, https://arxiv.org/abs/2303.08774
- “Deep reinforcement learning from human preferences”, https://proceedings.neurips.cc/paper_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf
- "An introduction to variational autoencoders”, https://www.nowpublishers.com/article/DownloadSummary/MAL-056
- “Probabilistic Machine Learning, Advanced Topics”, Kevin Murphy. Part IV: Generation. https://probml.github.io/pml-book/book2.html