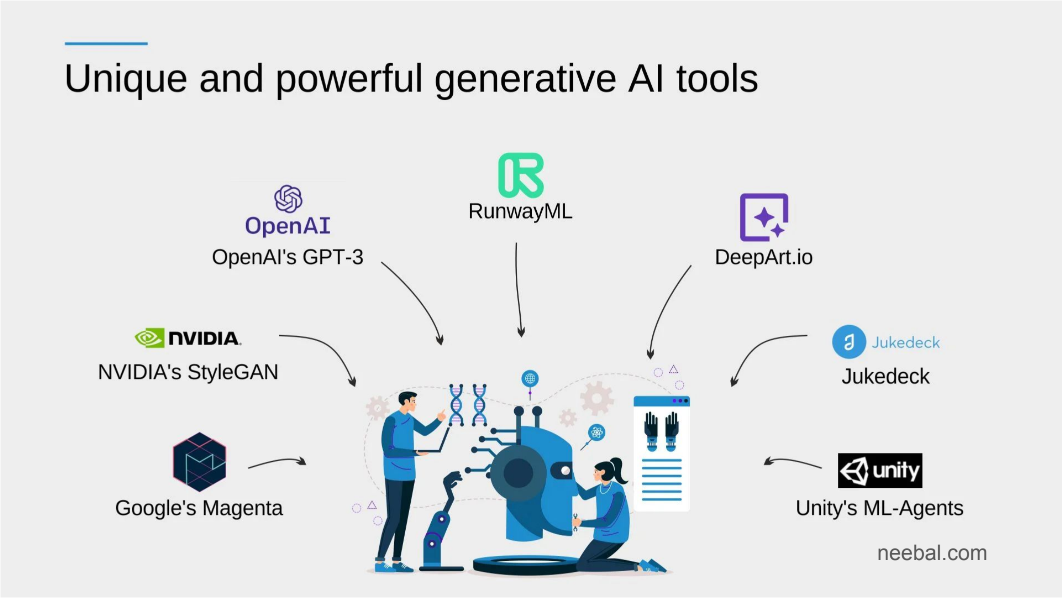
Generative AI and Astronomy



Ashish Mahabal
Deputy Director, Center for Data Driven Discovery, Caltech
Astronomy and AI, 6 Jan 2025

Overview

- **Historical Overview of AI**
  - Key milestones
  - From symbolic AI to ML and neural networks
- **What are Generative Models?**
  - Definition and Comparison with discriminiative models
- **Cutting-edge Developments in Generative AI**
  - Transformers and their role in generative tasks
  - Diffusion models
- **Some astronomy**
  - … interspersed

Historical Overview of Artificial Intelligence

Evolution from Symbolic AI to Modern Neural Networks

What is AI?

- Understanding natural language
- Recognizing patterns and images
- Making decisions based on data

What is it that Humans can do but AI can not?

Write code?
Write essays?
Write poetry?

What is it that Humans can do but AI can not?

Write code?
Write essays?
Write poetry?

Now?

Tomorrow?

In fifty years?

Joanna Maciejewska—Snakebitten is on preorder now!
@AuthorJMac · **Follow**

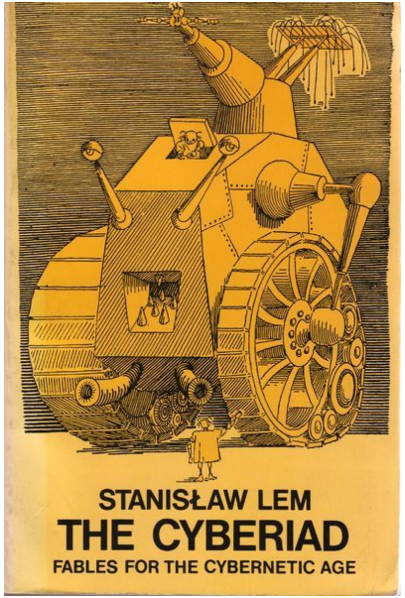
You know what the biggest problem with pushing all-things-AI is? Wrong direction.
I want AI to do my laundry and dishes so that I can do art and writing, not for AI to do my art and writing so that I can do my laundry and dishes.

4:50 AM · Mar 29, 2024

Early Concepts and Philosophical Foundations

Philosophical roots: Discussions of artificial beings in myths and folklore (and sci-fi).

The “**First Sally - Trurl’s Electronic Bard**” tells the story of Trurl the Constructor, who created a machine to write poems. He quickly realizes that he needs to start wayback to imbibe culture; the machine can do style transfer; leads to huge electricity bills and so on. (Written in 1960!)



https://english.lem.pl/blog/a-kingdom-in-the-box/

Symbolic AI

AI systems that directly use human-understandable symbols to process knowledge.

Major achievements: ELIZA (1966), SHRDLU (1970).

ELIZA natural language processing computer program (1964 -- 1966) by Joseph Weizenbaum at MIT. Named after Eliza Doolittle, a character in George Bernard Shaw's play *Pygmalion*.

Superficiality of communication between humans and machines.
Elicited emotional responses from users who interacted with it, leading to significant discussions about the possibilities of computer programs in mimicking human conversation.

Symbolic AI

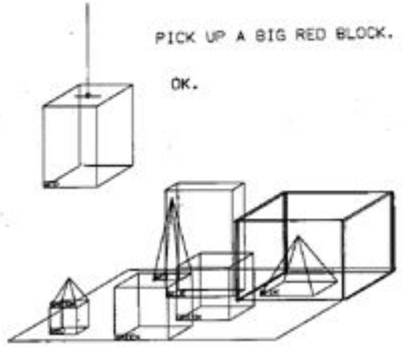
AI systems that directly use human-understandable symbols to process knowledge.

Major achievements: ELIZA (1966), SHRDLU (1970).

ELIZA's DOCTOR Script

- **Interaction Example**:
  - User: "I am feeling sad."
  - ELIZA: "I am sorry to hear you are sad. Can you tell me what is making you feel sad?"
- **Technique**: Reflect the user's statements back at them in the form of questions. Based on Carl Rogers' client-centered therapy -- a non-directive method of psychotherapy that seeks to facilitate the client's growth by allowing the client to lead the discussion.

Symbolic AI



AI systems that directly use human-understandable symbols to process knowledge.

Major achievements: ELIZA (1966), SHRDLU (1970).

- **SHRDLU** natural language processing by Terry Winograd at MIT in the early 1970s. Ability to manipulate the blocks (cubes, pyramids) on the screen.

Capabilities:

- **Understanding Commands**: "Pick up the red block," "Find a block which is taller than the one you are holding and put it into the box."
- **Answering Questions**: "Is there anything which is bigger than the red cube?" or "What is supporting the blue block?"
- **Handling Ambiguities**: if there were two red blocks, it might ask, "Do you mean the red cube or the red pyramid?"

Internal Representation: SHRDLU used a model of the world that included various data structures to represent the position, size, and relationships between the blocks. This model was dynamically updated as the state of the world changed.

Hard-Coded Knowledge Systems

Systems were primarily rule-based, their behavior dictated by a set of explicitly defined rules and decision trees crafted by programmers. These systems:

- **Operated within Narrow Boundaries**: lacked flexibility.
- **Required Extensive Manual Effort**: Updating or adapting these systems to new tasks or data required significant manual reprogramming.
- **Lacked Scalability**: handling more general or complex tasks was often impractical due to the exhaustive need for detailed rules.

Learning from Data: Machine Learning Systems

Generalize Across Tasks: ML models, once trained, can perform a variety of tasks based on their learning, even those not explicitly programmed.

Automate Feature Extraction: Unlike rule-based systems where features must be manually crafted, ML algorithms can automatically discover useful patterns or features in the data.

Adapt and Improve Over Time: Machine learning models can improve their performance as they encounter more data or as data changes over time.

Significance of the Transition

Efficiency and Scalability: Learning from data allows AI systems to scale more efficiently across different domains and problems without human intervention for every new scenario.

Enhanced Capabilities: Machine learning, especially deep learning, has enabled breakthroughs in complex tasks like image and speech recognition, natural language processing, and autonomous driving, which were challenging with rule-based systems.

Dynamic Adaptation: Systems can now adapt to new, unforeseen scenarios, learning from new data in a way that was not possible with hard-coded knowledge.

Decision trees, Support Vector Machines, Neural Networks

Summary so far

Roots of current generative AI are a few decades old

There is a variety of techniques to explore

From data we can find correlations, but not necessary causality

Backpropagation brought about a revolution

Generative AI

Definition

- Generative AI involves algorithms that create new data similar to existing data.

Examples and applications

- Image generation (e.g., DALL-E)
- Text synthesis (e.g., GPT-3)
- Music composition (e.g., Jukedeck)

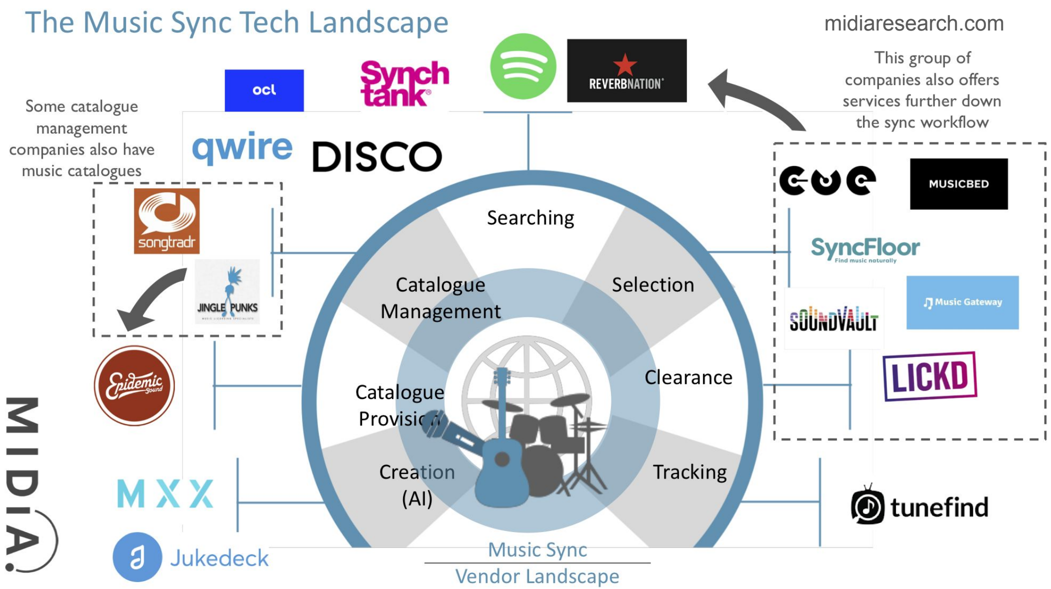


Image generation

Prompts are needed

Understanding connection between images and prompts

Understanding invariants (translation, rotation, flips etc.)

Clip algorithm

Text generation

All it is doing is getting next word

But what about math? Contacting Wolfram Alpha

Are some companies taking over? (Question of ethics)

Translations, Styles, Parallels, …

Hallucinations

Music generation

Tune, pitch, instruments, mixing, …

Plagiarism

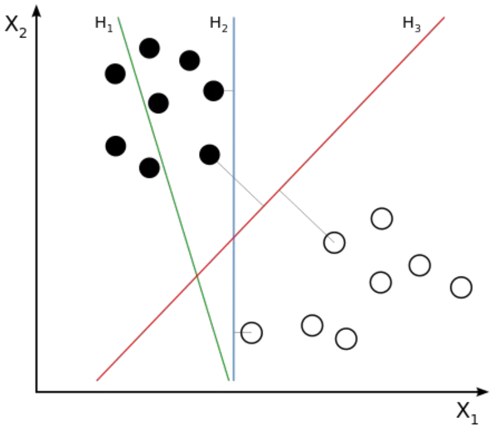
Clearly it gets more complex

Program I had written using perl

Generative vs. Discriminative Models

Discriminative Models

- Classify data instances
- Examples: CNNs, SVMs



H_1 does not separate the classes. H_2 does, but only with a small margin. H_3 separates them with the maximal margin.

wikipedia

Differences

Discriminative models focus on decision boundaries

Generative models focus on understanding the distribution of data

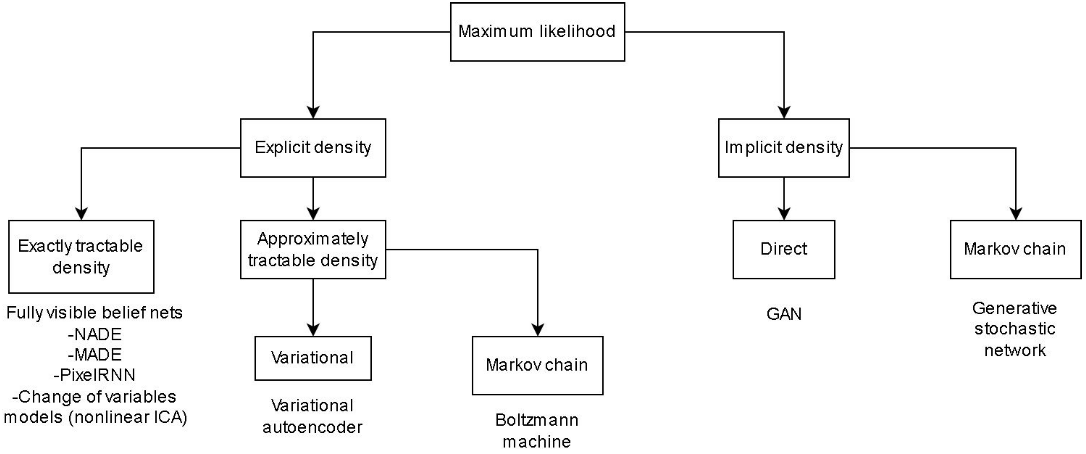
Generative vs. Discriminative Models

Generative Models

GANs: Ian Goodfellow 2014; Consist of a Generator and a Discriminator

VAEs: Kingma and Welling 2013; Probabilistic graphical models

Transformers: Vaswani et al. 2017; Self-attention mechanisms

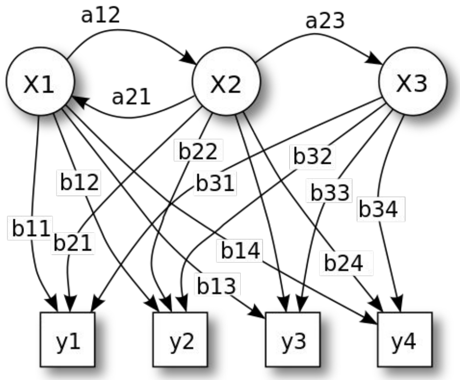


Historical Context

Key milestones in Generative AI development

- Early statistical models
  - Markov models, Hidden Markov Models (HMMs)
- Rise of neural networks
  - Advancements in deep learning and unsupervised learning
- Breakthroughs in GANs, VAEs, and Transformers

HMM



What is a Markov Model?

A Markov model is a mathematical system that undergoes **transitions from one state to another within a finite set of states**. It is based on the Markov property, which states that the **future state depends only on the current state** and not on the sequence of events that preceded it. This is often referred to as "**memorylessness**."

Key Components of a Markov Model

States: A finite set of possible states in which the system can be.

Transitions: The process of moving from one state to another.

Transition Probabilities: The probabilities associated with moving from one state to another. These probabilities are usually represented in a transition matrix.

Example of a Markov Model

	Sunny	**Rainy**
Sunny	0.8	0.2
Rainy	0.4	0.6

Sunny today suggests tomorrow to be sunny with 80% chance.
Rainy today suggests tomorrow to be sunny with 40% chance.

Predict the chance on day 2, 3 etc. (by hand, then write a simple program for that)

Applications of Markov Models

Speech Recognition: HMMs are widely used in speech recognition to model sequences of spoken words.

Bioinformatics: Markov models are used to model biological sequences and evolutionary processes.

Finance: To model stock prices and economic indicators.

Queueing Theory: Used to model systems with queues, like customer service centers or network traffic.

Hidden Markov Models

Here we have to infer from observables what the hidden state of the model is.

Example: Weather and Activity

States (Hidden)

1. **Sunny**
2. **Rainy**

Observations (Visible)

1. **Walking**
2. **Shopping**
3. **Cleaning**

Gaussian Mixture Models

Representation of (Static) Data Distribution: Model data as a combination of multiple Gaussian distributions. Each is a cluster in the data with its own mean and covariance, capturing the underlying structure and variability providing flexible representation of complex, multimodal distributions.

Probabilistic Model: Probability density function (pdf) allows for the modeling of uncertainty and the generation of new samples from the learned distribution. The mixture of Gaussians can approximate any continuous probability distribution given enough components.

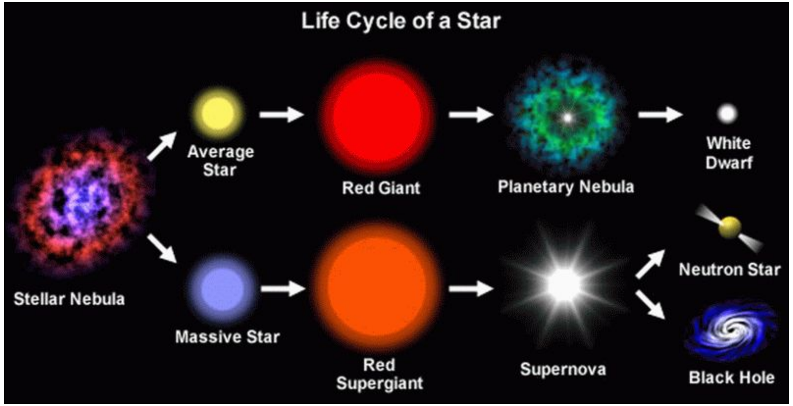
Gaussian Mixture Models

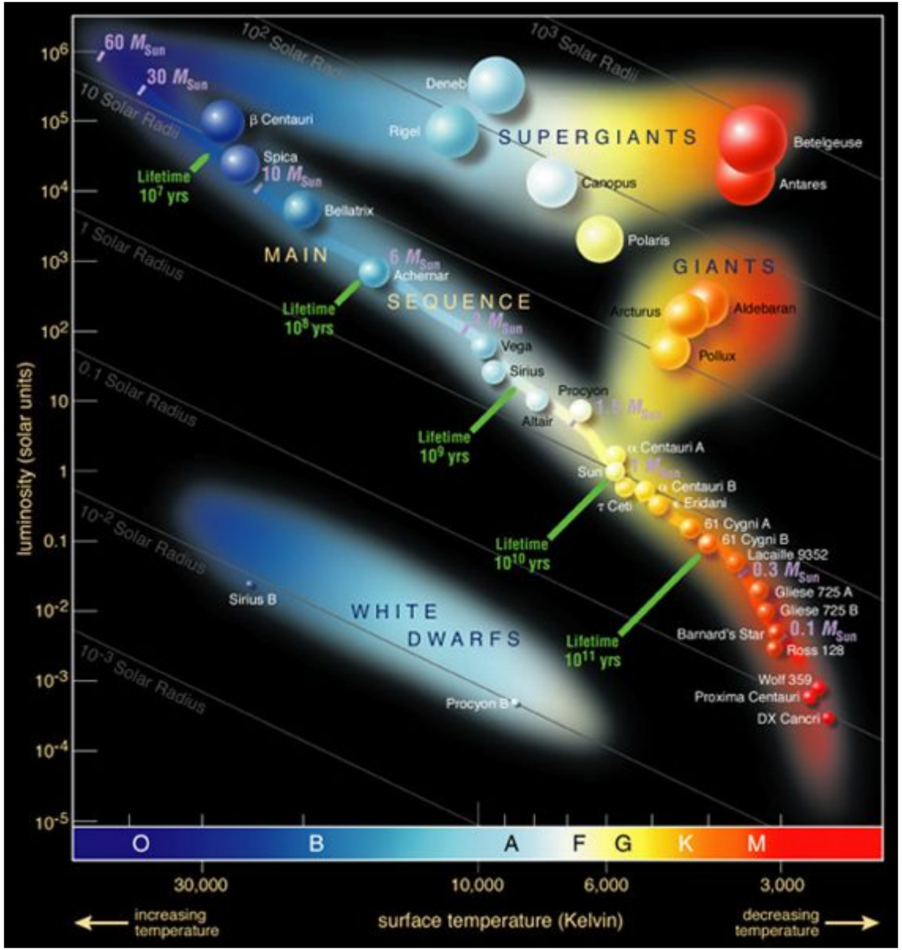
Expectation-Maximization Algorithm: The Expectation-Maximization (EM) algorithm is used to estimate the parameters of GMMs. EM iteratively improves the parameter estimates to maximize the likelihood of the observed data under the model. This iterative approach helps in finding the best-fitting model to the data, ensuring that the generative process is well-represented.

Data Generation: Once a GMM is trained, new data points can be generated by sampling from the learned distribution. This involves selecting a Gaussian component based on the component weights and then sampling from the chosen Gaussian distribution.

Latent Variable Models: Latent variables correspond to the cluster memberships. Incorporation of hidden structures in the data, facilitating sophisticated generation processes that account for underlying patterns and relationships.

HR diagram example





https://www.cosmos.esa.int/web/cesar/the-hertzsprung-russell-diagram

Go to the GMM/HR GMM notebook

https://colab.research.google.com/drive/1InJ1O44g0wW97NsCzlf35Z91iItSzbdj?usp=sharing

Exercise: Create a GMM that is closer to the HR diagram shown in the previous slide

Advanced Generative Models

- **Deep Generative Models**
  - Introduction to Variational Autoencoders (VAEs)
  - Introduction to Generative Adversarial Networks (GANs)

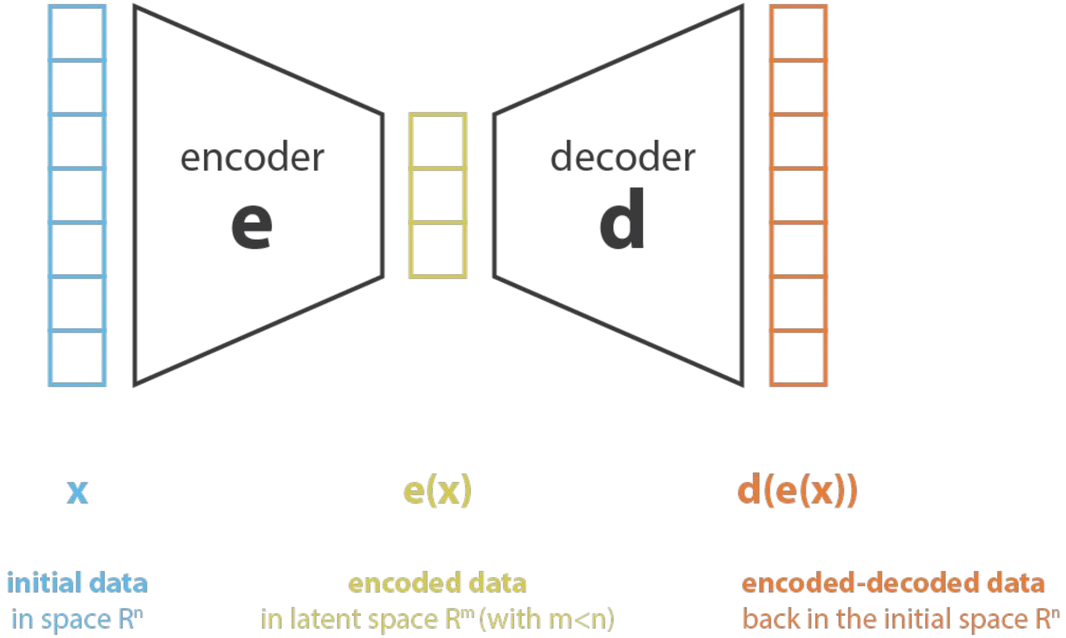
Overview of VAEs

Basic concept

- Encoder maps data to latent space
- Decoder reconstructs data from latent space

Encoder and Decoder

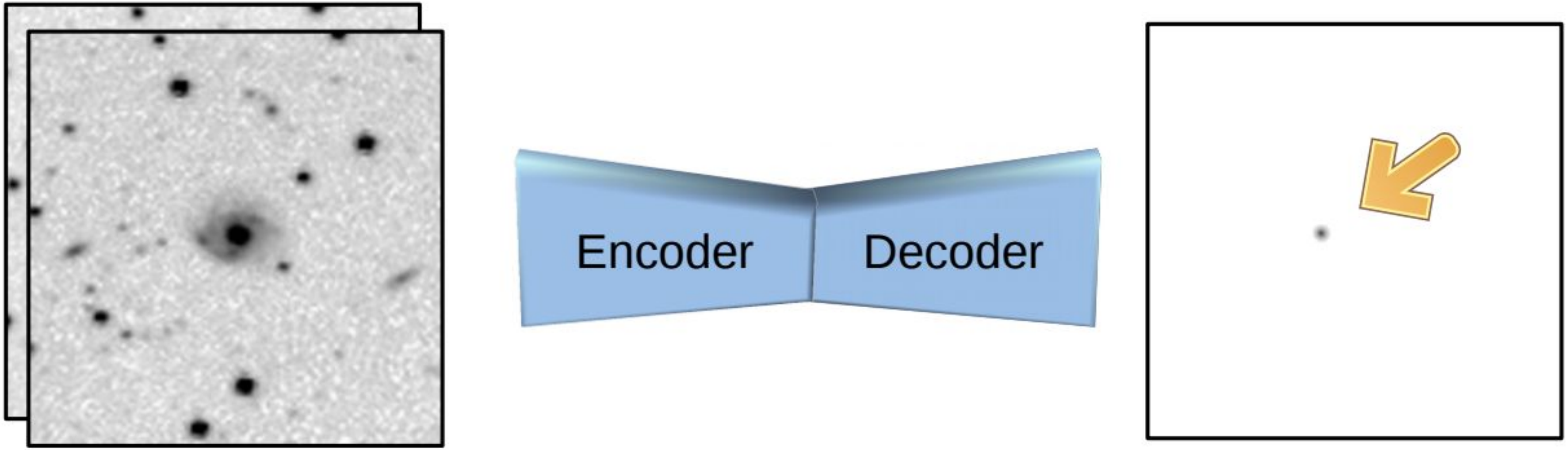
- Encoder compresses data into latent variables
- Decoder reconstructs data from latent variables
- Uses variational inference for training



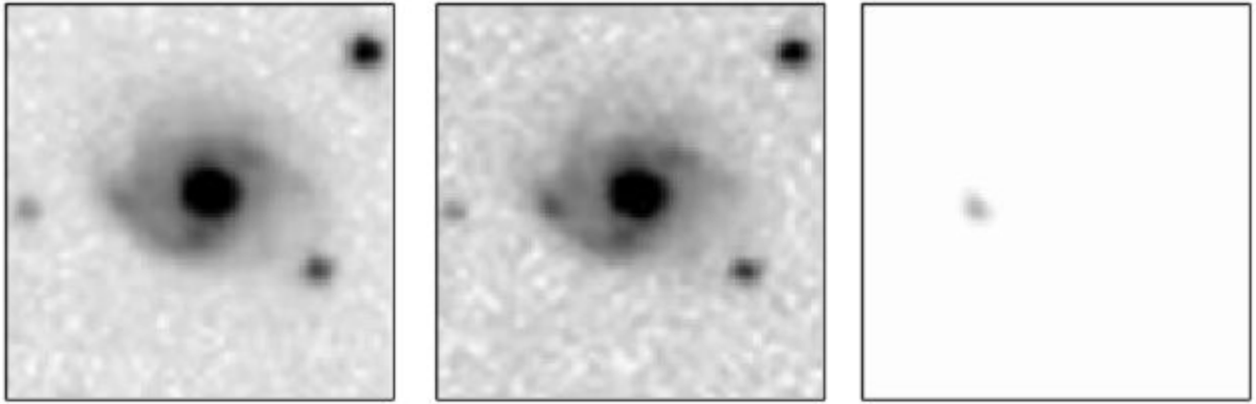
$x = d(e(x))$ ➔ **lossless encoding**
no information is lost when reducing the number of dimensions

$x \neq d(e(x))$ ➔ **lossy encoding**
some information is lost when reducing the number of dimensions and can't be recovered later

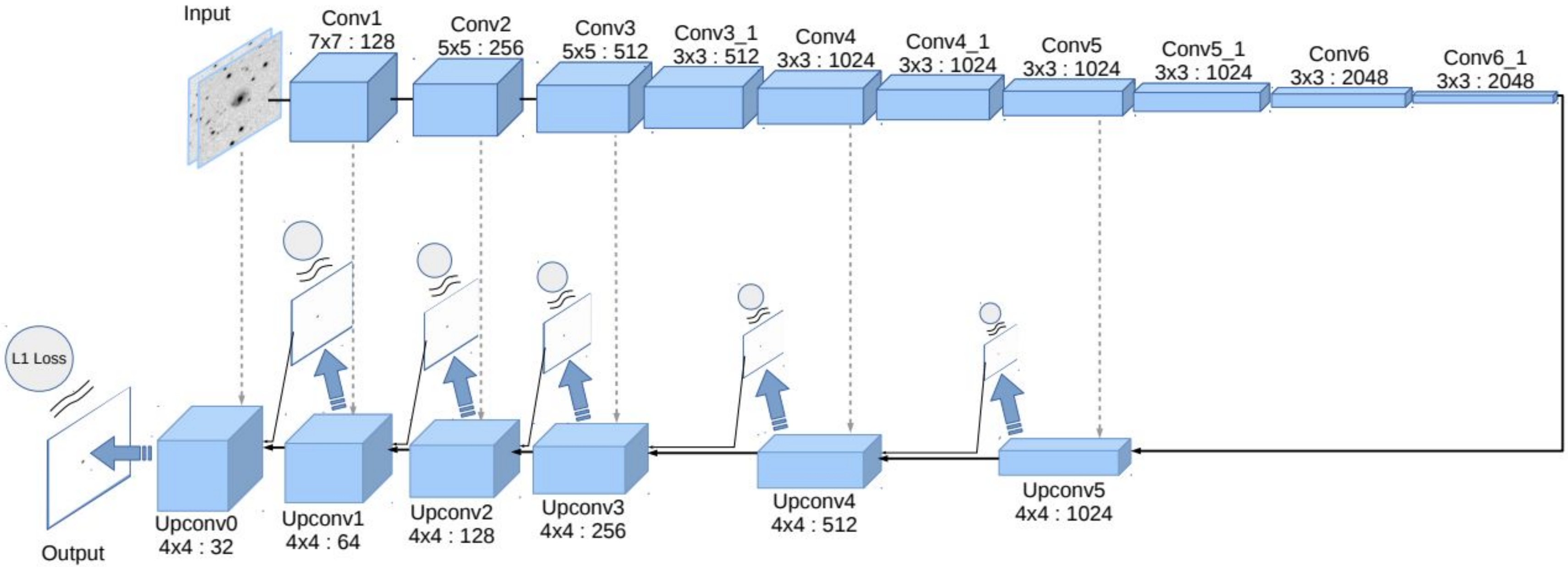
https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73



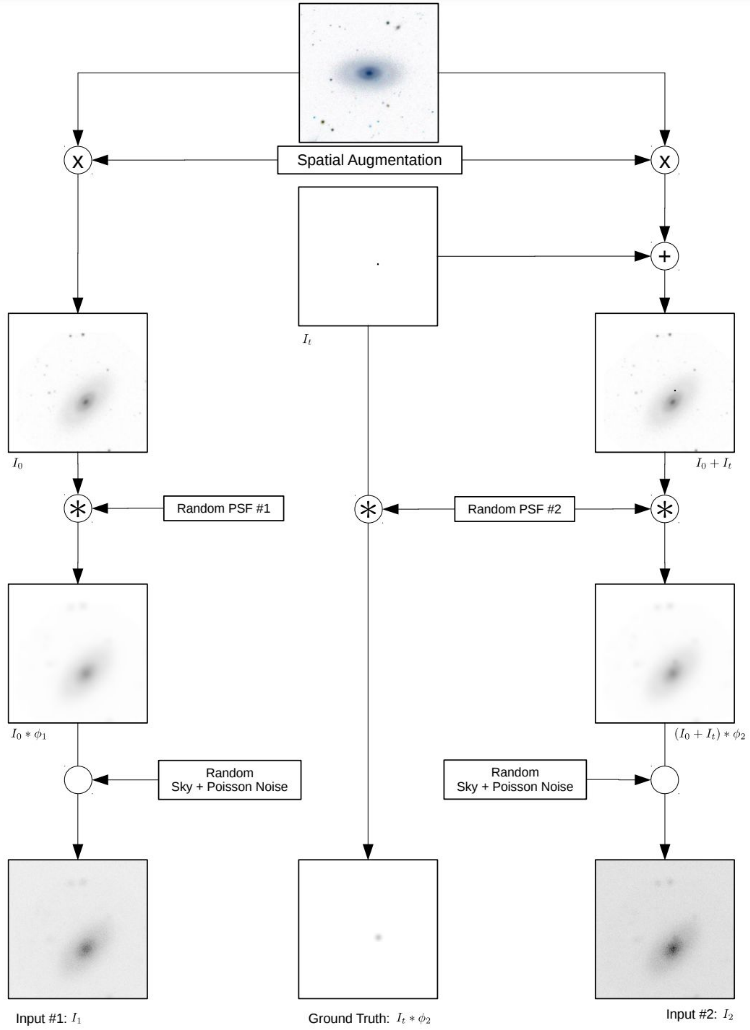
Sedaghat, Mahabal 1710.01422



$$I1 = I0 * \varphi1 + S1 + N1$$
$$I2 = (I0 + It) * \varphi2 + S2 + N2$$



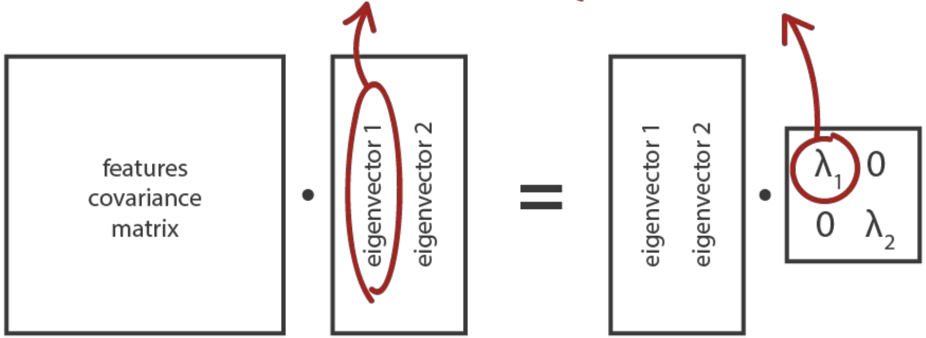
Input
Conv1
7x7 : 128
Conv2
5x5 : 256
Conv3
5x5 : 512
Conv3_1
3x3 : 512
Conv4
3x3 : 1024
Conv4_1
3x3 : 1024
Conv5
3x3 : 1024
Conv5_1
3x3 : 1024
Conv6
3x3 : 2048
Conv6_1
3x3 : 2048
L1 Loss
Output
Upconv0
4x4 : 32
Upconv1
4x4 : 64
Upconv2
4x4 : 128
Upconv3
4x4 : 256
Upconv4
4x4 : 512
Upconv5
4x4 : 1024

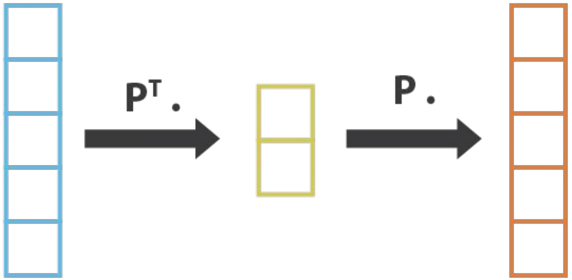


Has to incorporate physical conditions like the PSF eigenvector associated to the greatest eigenvalue λ_1 and orthogonal to other columns

greatest eigenvalue of the covariance matrix C (in absolute value)

notice that $d(e(x)) \neq x$ as soon as $C \neq P \lambda P^T$

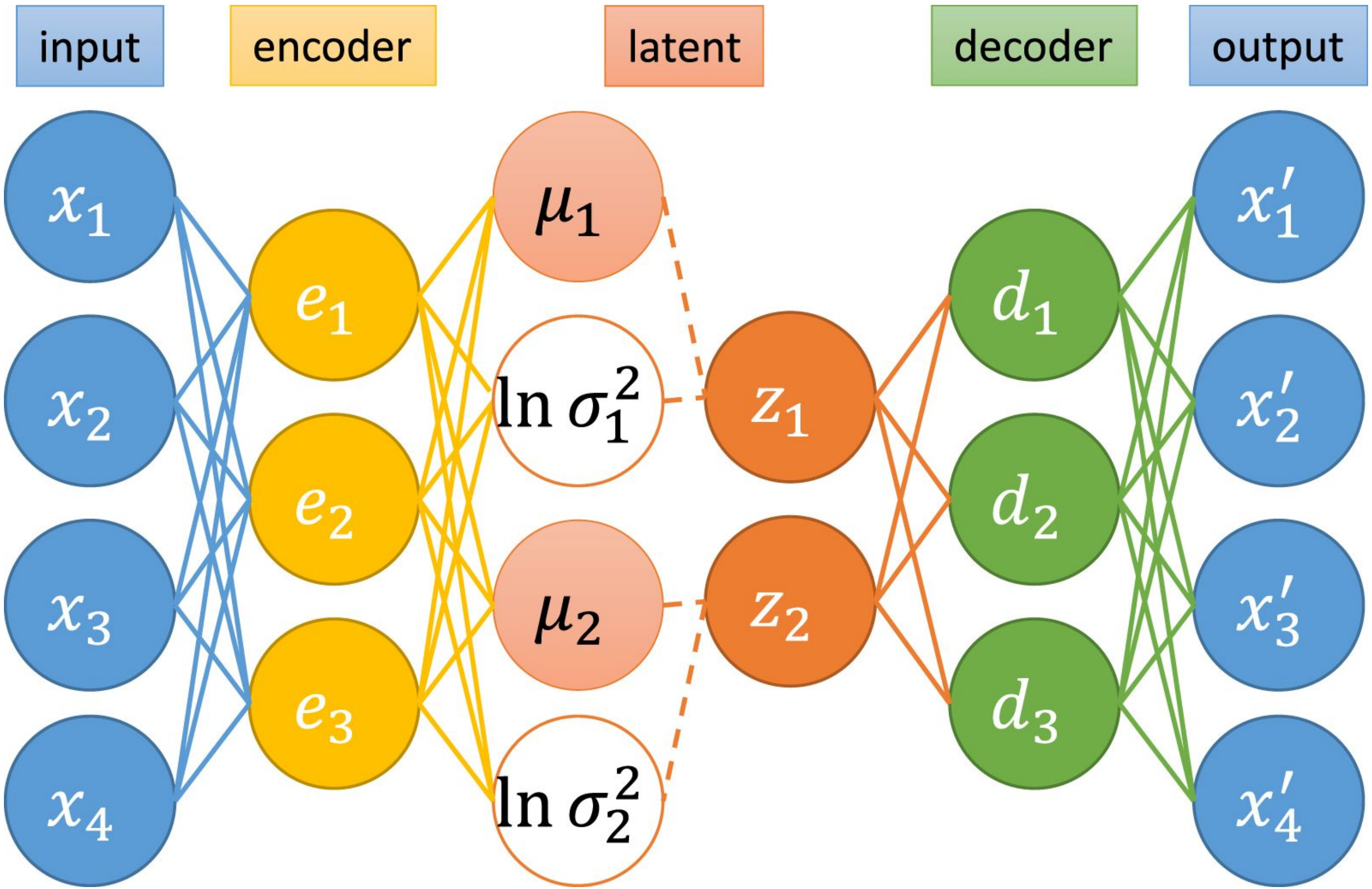




$$C \cdot P = P \cdot \lambda$$

x $\quad e(x) = P^T x \quad d(e(x)) = PP^T x$

In principle anything could be used for dimensionality reduction (encoding). But neural networks are superior.



Portillo et al. 2020

Application to SDSS spectra

Evidence lower bound (ELBO) is the objective function.
It is the sum of the reconstruction loss and the Kullback–Leibler (KL) divergence between the latent distribution for the input $q(z|x)$ and the prior $p(z)$

$$\mathrm{ELBO} = L(\boldsymbol{x}, \boldsymbol{x}') + D_{\mathrm{KL}}(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})).$$

$$D_{\mathrm{KL}}(q||p) = \int q(z)\log\left(\frac{q(z)}{p(z)}\right)dz.$$

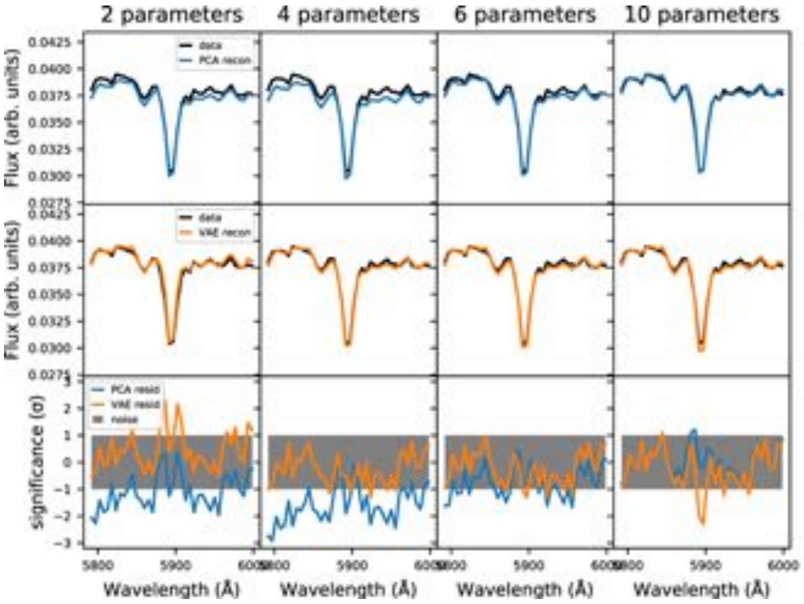
Table 1

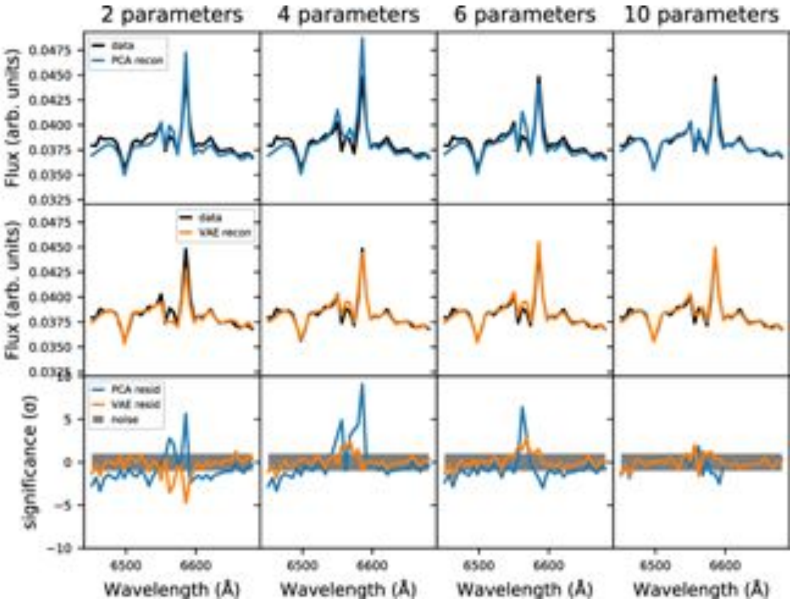
Best Architectures and MMD Coefficients λ Found by Random Search for VAEs with Two, Four, Six, and 10 Latent Parameters

Latent Parameters	Architecture	λ
2	1000-1663-42-2-42-1663-1000	11.2
4	1000-1134-64-4-64-1134-1000	21.2
6	1000-703-94-6-94-703-1000	3.02
10	1000-549-110-10-110-549-1000	7.72

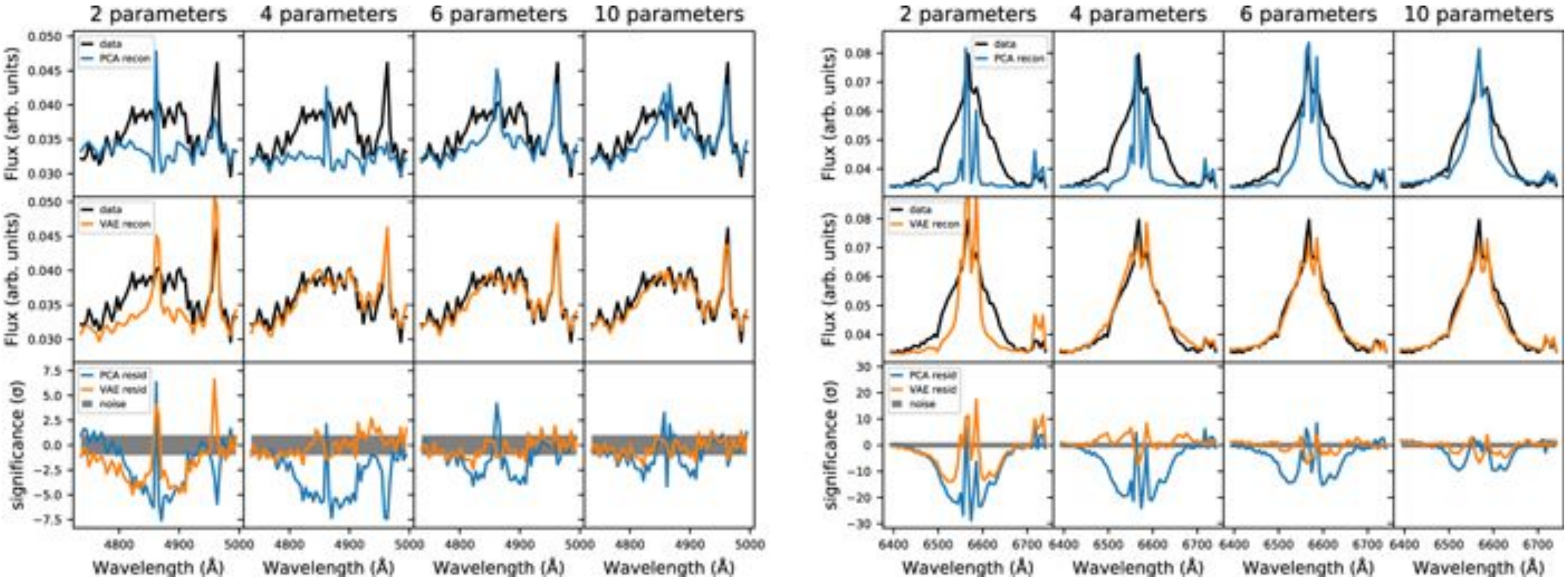
$$D_{\mathrm{MMD}} = \frac{1}{m^2}\sum_{i,j=1}^{m}\kappa(u_i,u_j) - \frac{2}{mn}\sum_{i,j=1}^{m,n}\kappa(u_i,v_j) + \frac{1}{n^2}\sum_{j=1}^{n}\kappa(v_i,v_j)$$

MMD is the maximum mean discrepancy

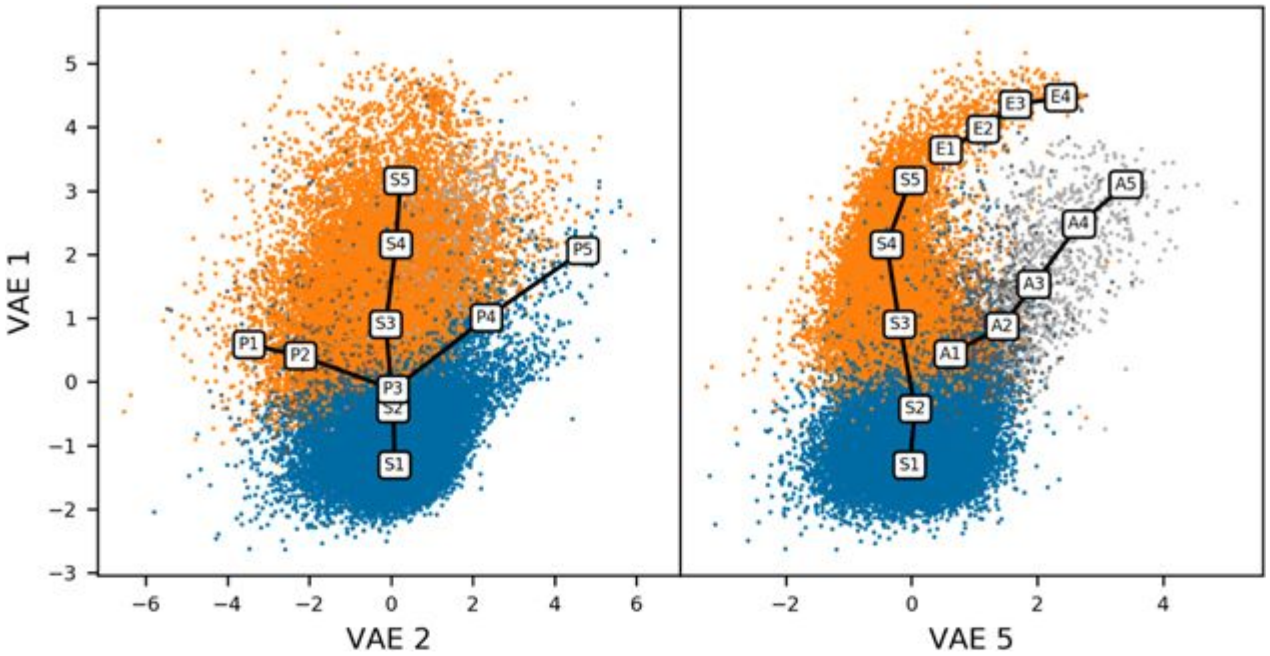




Reconstruction of two different lines with PCA and VAE with different number of parameters.

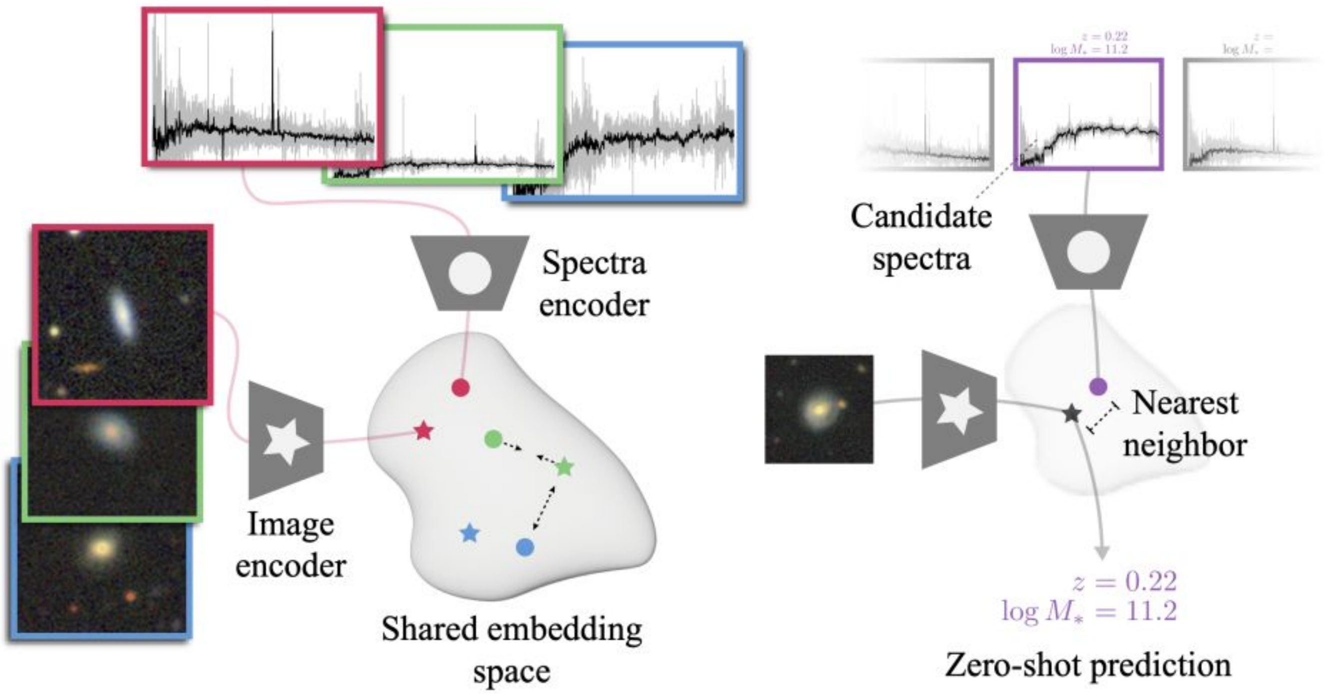


More complex lines and PCA is clearly falling behind



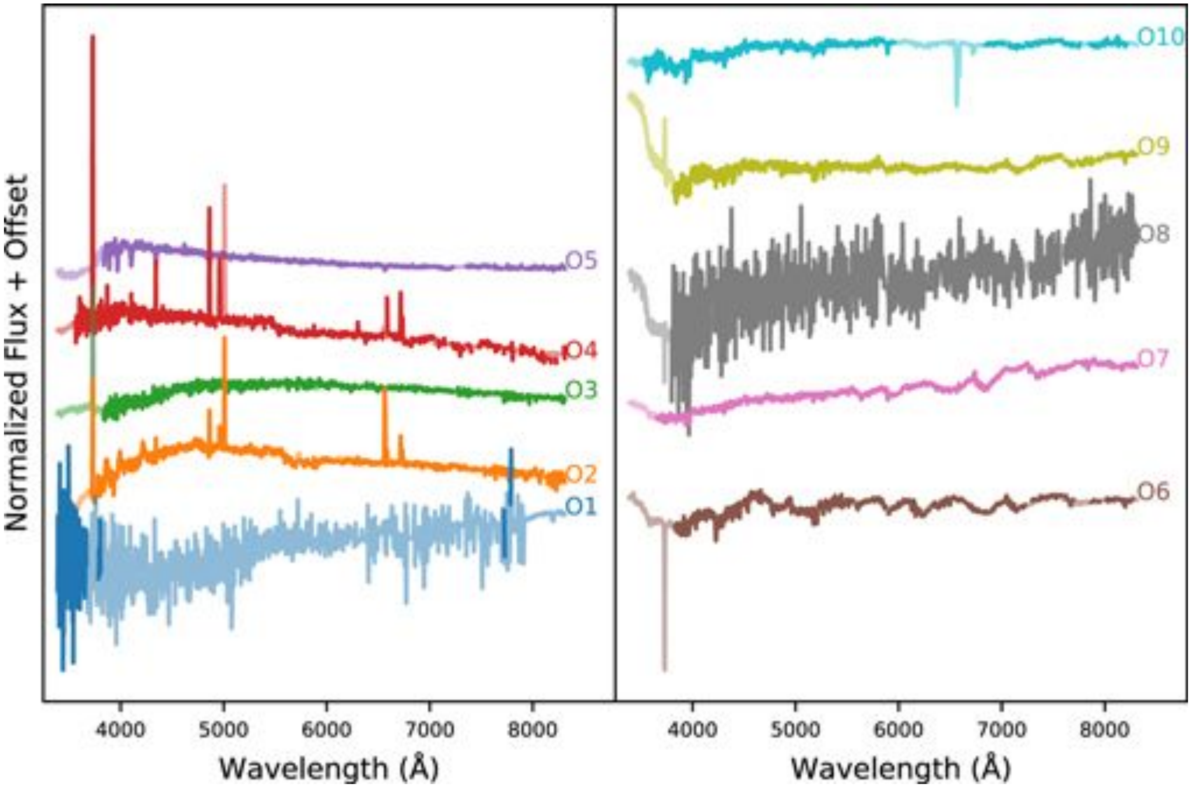
VAE synthetic spectra to understand latent variables (e.g. star formation) by following galaxy distributions.

AstroCLIP (Lanusse et al. 2023) uses images and spectra of galaxies



Outlier detection

local outlier factor (LOF) algorithm (Breunig et al. [2000](#)) is used to identify outliers. The algorithm estimates the local density of each point by using *k* nearest neighbors and then identifies points with densities much lower than their neighbors' as outliers.



Liang et al. 2023 find more interesting spectra in DESI using auto-encoders and normalizing flows

Combination of Probabilistic Modeling and Neural Networks

VAEs merge two powerful concepts: probabilistic graphical models and deep learning. This combination allows VAEs to leverage the strengths of both worlds:

- **Probabilistic Graphical Models**: These models handle uncertainty and variability in data by modeling probability distributions.
- **Deep Learning**: Neural networks, particularly deep architectures, excel at learning complex patterns and representations from high-dimensional data.

Latent Space Representation

VAEs introduce a stochastic latent space where data is encoded. This latent space has several advantages:

- **Smooth Interpolations**: Because the latent space is continuous, VAEs can generate smooth transitions between different data points, making them suitable for tasks like image morphing and style transfer.
- **Structured Latent Space**: The latent space often captures meaningful variations in the data, such as different features in images (e.g., facial expressions, orientations).

Principled Approach to Generative Modeling

VAEs provide a rigorous probabilistic foundation for generative modeling:

- **Encoder-Decoder Architecture**: The encoder maps input data to a latent space distribution (typically Gaussian), and the decoder reconstructs the data from samples drawn from this distribution.
- **Variational Inference**: The variational approach approximates complex posterior distributions, making inference tractable and efficient.

Training with Variational Inference

The core innovation in VAEs is their training methodology, which uses variational inference:

- **Evidence Lower Bound (ELBO)**: VAEs maximize the ELBO, a lower bound on the log-likelihood of the data. This involves balancing two terms:
  - **Reconstruction Loss**: Measures how well the decoder reconstructs the input data.
  - **KL Divergence**: Regularizes the latent space to match a prior distribution (usually a standard Gaussian).

Scalability and Flexibility

VAEs are scalable and flexible, making them applicable to various types of data:

- **Different Data Types**: VAEs have been adapted to handle images, text, audio, and more.
- **Complex Architectures**: Extensions like Convolutional VAEs (for images) and Recurrent VAEs (for sequences) allow VAEs to handle complex, high-dimensional data efficiently.

Applications and Impact

VAEs have had a profound impact on numerous applications:

- **Image Generation**: VAEs can generate new, realistic images after being trained on a dataset of images.
- **Data Augmentation**: VAEs can augment training datasets by generating new examples, improving the performance of downstream tasks.
- **Anomaly Detection**: By learning the normal data distribution, VAEs can identify anomalies as data points that do not fit the learned distribution.
- **Semi-Supervised Learning**: VAEs can leverage both labeled and unlabeled data, improving performance when labeled data is scarce.

Interpretability

The latent space of VAEs can be interpreted and manipulated:

- **Latent Variables**: Each dimension in the latent space can correspond to specific features or variations in the data.
- **Disentanglement**: With appropriate modifications (e.g., β-VAE), VAEs can disentangle latent factors, leading to more interpretable models. Adjustable hyperparameter β balances latent channel capacity and independence constraints with reconstruction accuracy.

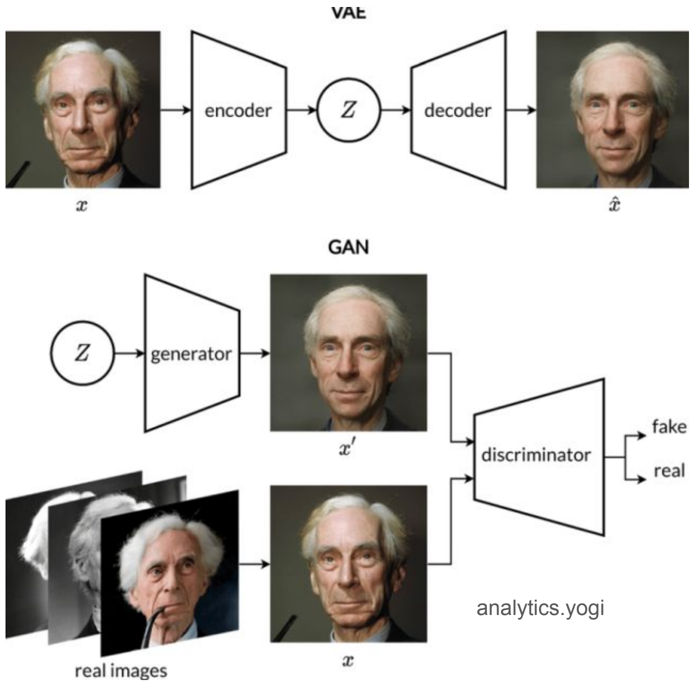
Overview of GANs

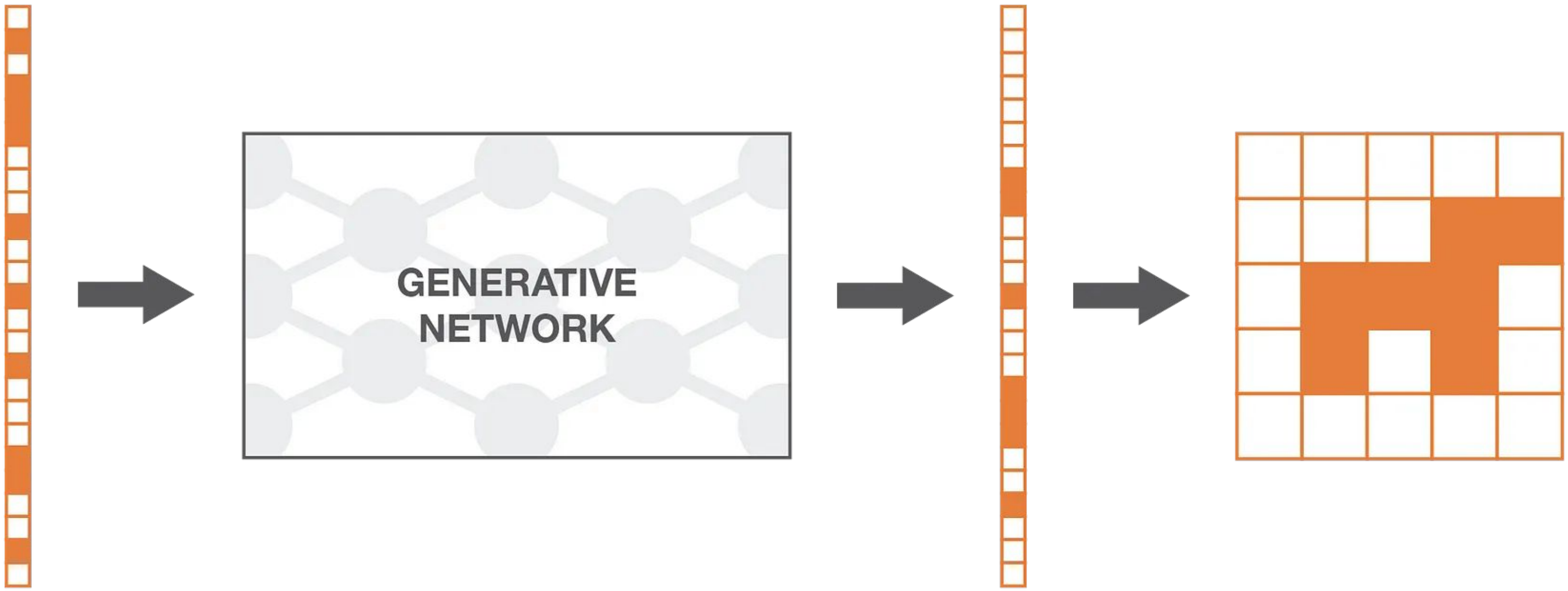
Basic concept

- Generator creates fake data
- Discriminator distinguishes between real and fake data

Generator and Discriminator

- Training involves both networks in a minimax game
- Generator tries to fool the Discriminator
- Discriminator tries to identify real vs. fake data
- Loss functions: Generator loss and Discriminator loss





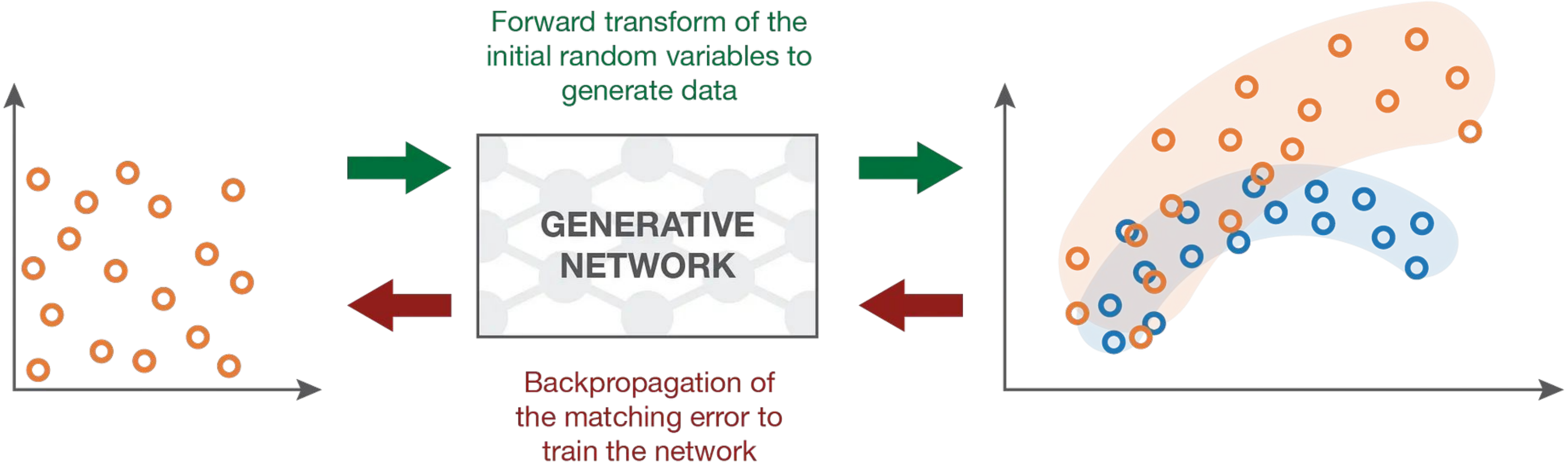
Input random variable (drawn from a simple distribution, for example uniform).

The generative network transforms the simple random variable into a more complex one.

Output random variable (should follow the targeted distribution, after training the generative network).

The output of the generative network once reshaped.

https://towardsdatascience.com/understanding-generative-adversarial-networks-gans-cd6e4651a29

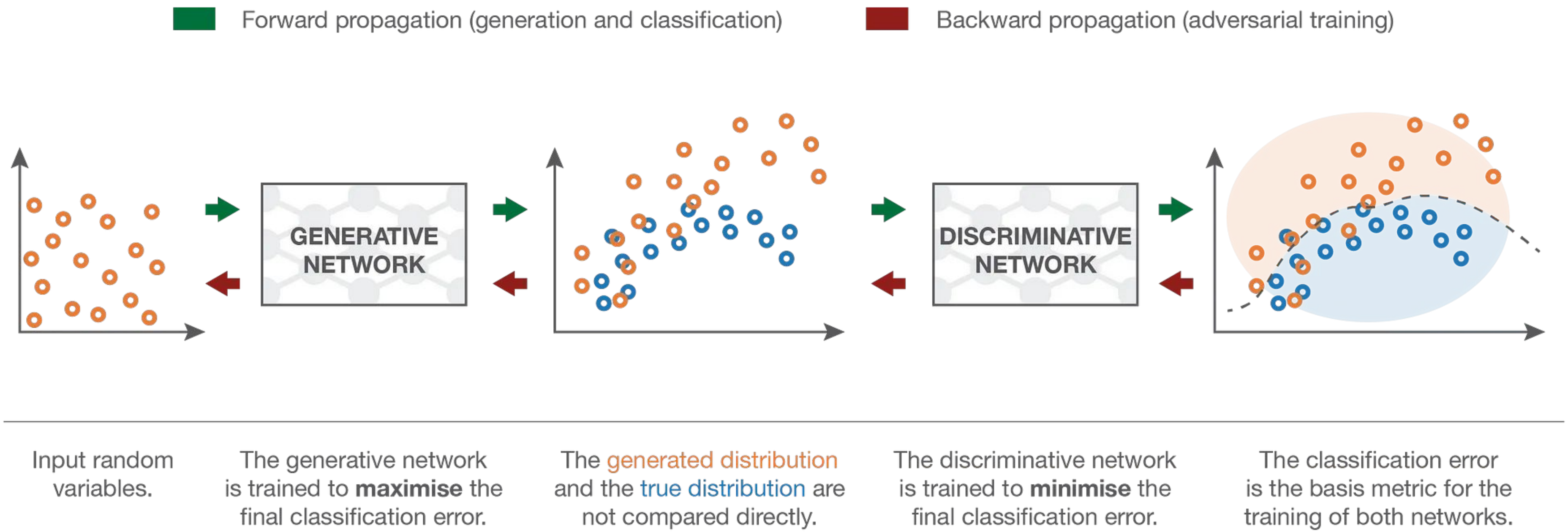


Input random variables (drawn from a uniform).

Generative network to be trained.

The generated distribution is compared to the true distribution and the “matching error” is backpropagated to train the network.

For comparing dogness, Maximum Mean Discrepancy (MMD) is used.



Instead of using backpropagation to train the network, it is used to draw the two distributions closer through adversarial training

- a generative network G(.) that takes a random input z with density p_z and returns an output x_g = G(z) that should follow (after training) the targeted probability distribution
- a discriminative network D(.) that takes an input x that can be a “true” one (x_t, whose density is denoted p_t) or a “generated” one (x_g, whose density p_g is the density induced by the density p_z going through G) and that returns the probability D(x) of x to be a “true” data

Loss function for the GAN

$$\begin{aligned} E(G, D) &= \frac{1}{2}\mathbb{E}_{x\sim p_t}[1 - D(x)] + \frac{1}{2}\mathbb{E}_{z\sim p_z}[D(G(z))] \\ &= \frac{1}{2}\left(\mathbb{E}_{x\sim p_t}[1 - D(x)] + \mathbb{E}_{x\sim p_g}[D(x)]\right) \end{aligned}$$

$$\max_G \left(\min_D E(G, D)\right)$$

Possible challenges

1. **Common Issues**:
   - Mode collapse
   - Non-convergence
   - Vanishing gradients
2. **Techniques to Overcome Challenges**:
   - Feature matching
   - Mini-batch discrimination
   - Progressive growing
   - Use of spectral normalization

Causes of mode collapse

1. **Training Dynamics**: The adversarial nature of GANs can lead to instability in training. If the discriminator becomes too strong, it can overpower the generator, leading to mode collapse.
2. **Imbalanced Updates**: If the generator or discriminator is updated too frequently compared to the other, it can cause training instability and mode collapse.
3. **Architectural Choices**: Certain choices in the architecture of the generator and discriminator can predispose the model to mode collapse.

Symptoms

Lack of Diversity: The generated samples show little variety, often producing similar outputs even for different inputs.

Repeated Patterns: The generator tends to produce the same few patterns repeatedly, failing to capture the full range of the target distribution.

Solutions

Improving Training Stability: Techniques like batch normalization, gradient penalty, and spectral normalization can help stabilize training.

Regularization: Applying regularization techniques to the generator and discriminator can help maintain a balance in their updates.

Architectural Changes: Modifying the network architecture, such as using different activation functions or adding more layers, can sometimes alleviate mode collapse.

Multiple Discriminators: Using multiple discriminators to evaluate the generator's output can help provide a more diverse feedback signal.

Unrolled GANs: Unrolling the optimization of the discriminator by several steps can help the generator receive a more stable gradient signal.

Minibatch Discrimination: Including information about the entire batch in the discriminator's decision can help detect and penalize mode collapse.

Variants of GANs

DCGAN

- Uses convolutional layers in both Generator and Discriminator
- Mode collapse to a lesser extent

Conditional GAN

- Uses additional information (labels) as input
- Requires labeled data

CycleGAN

- Translates images from one domain to another
- Learns mappings without paired images; uses two generators and two discriminators

Try a DCGAN for MNIST

https://colab.research.google.com/drive/1zz-4AyXrOUKfUe_kd3XduD4Uny38UFvI#scrollTo=-RYD1xQdB6zB

Data Preparation: The MNIST dataset is loaded and normalized to the range [-1, 1].

Model Definitions:

- **Generator**: Takes a noise vector and generates a 28x28 image.
- **Discriminator**: Takes a 28x28 image and outputs the probability that the image is real.

Loss Functions:

- **Generator Loss**: Measures how well the generator fools the discriminator.
- **Discriminator Loss**: Measures how well the discriminator distinguishes real images from fake images.

Optimizers: Adam optimizers are used for both models.
Training: The training loop iterates over the dataset, updating the generator and discriminator. Images are generated and saved at each epoch to visualize progress.

Checkpoints: Model checkpoints are saved every 15 epochs.

(One slide) overview of Tra

Self-attention mechanism

- Allows the model to focus on different parts of the input sequence
- Key innovation enabling deep contextual understanding

Applications

- Text generation (e.g., GPT-3)
- Text translation (e.g., BERT)
- Various other NLP tasks

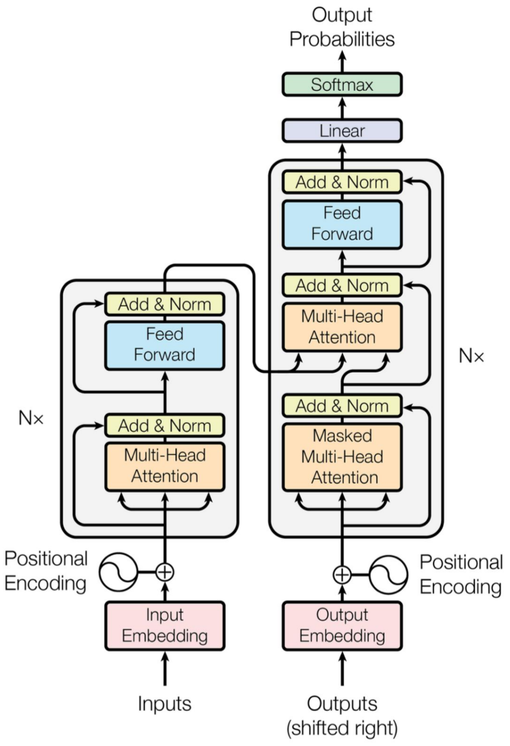
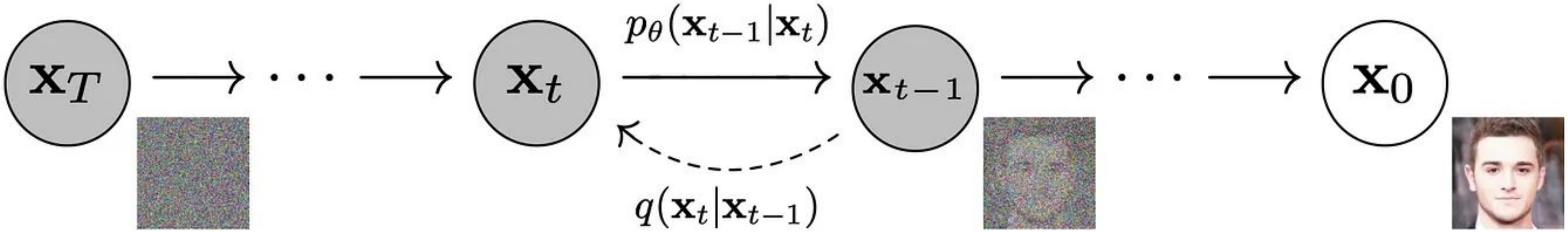


Figure 1: The Transformer - model architecture.

Diffusion

Going from Noise (entropy) to structure

Ho, Jain, Abbeel (2020) https://arxiv.org/abs/2006.11239



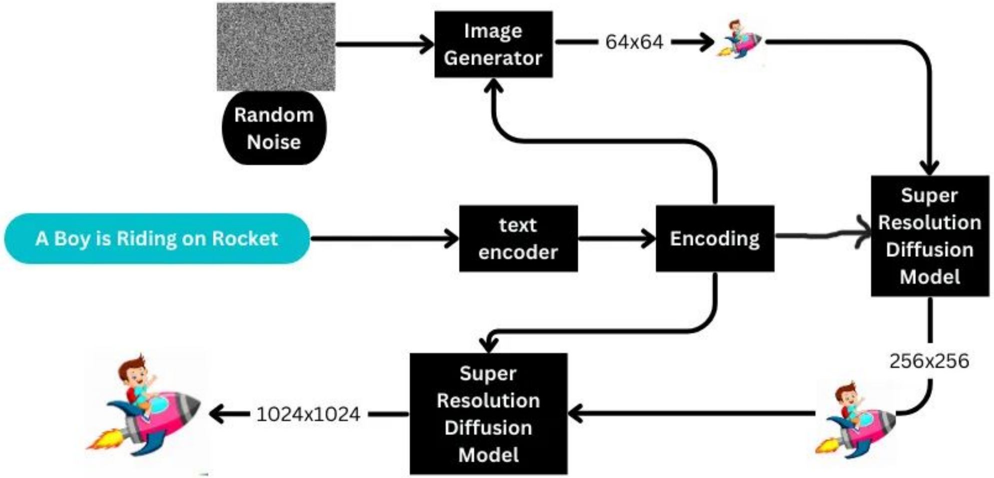
Guided by the text prompt

A low-res image is generated from noise which can then be improved by removal of noise and superresolution.

Google's IMAGEN, for example.

https://arxiv.org/pdf/2205.11487

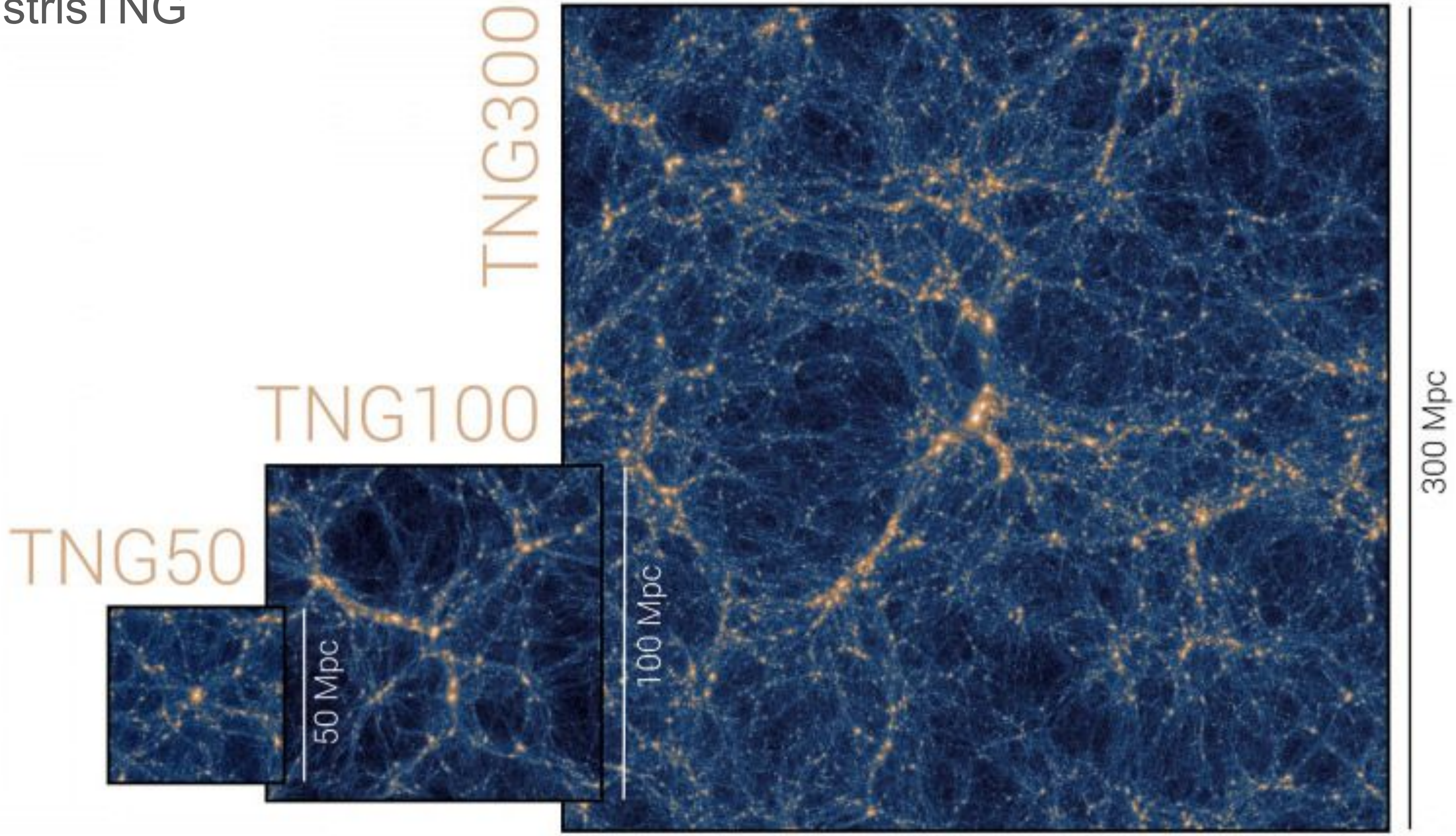
Model	FID-30K	Zero-shot FID-30K
AttnGAN [76]	35.49	
DM-GAN [83]	32.64	
DF-GAN [69]	21.42	
DM-GAN + CL [78]	20.79	
XMC-GAN [81]	9.33	
LAFITE [82]	8.12	
Make-A-Scene [22]	7.55	
DALL-E [53]		17.89
LAFITE [82]		26.94
GLIDE [41]		12.24
DALL-E 2 [54]		10.39
Imagen (Our Work)		**7.27**



https://shyampatel1320.medium.com/introduction-to-diffusion-models-and-imagen-the-magic-behind-text-to-image-generation-24221532580d#:~:text=The%20diffusion%20model%20is%20an,and%20generating%20new%20images%20vary.

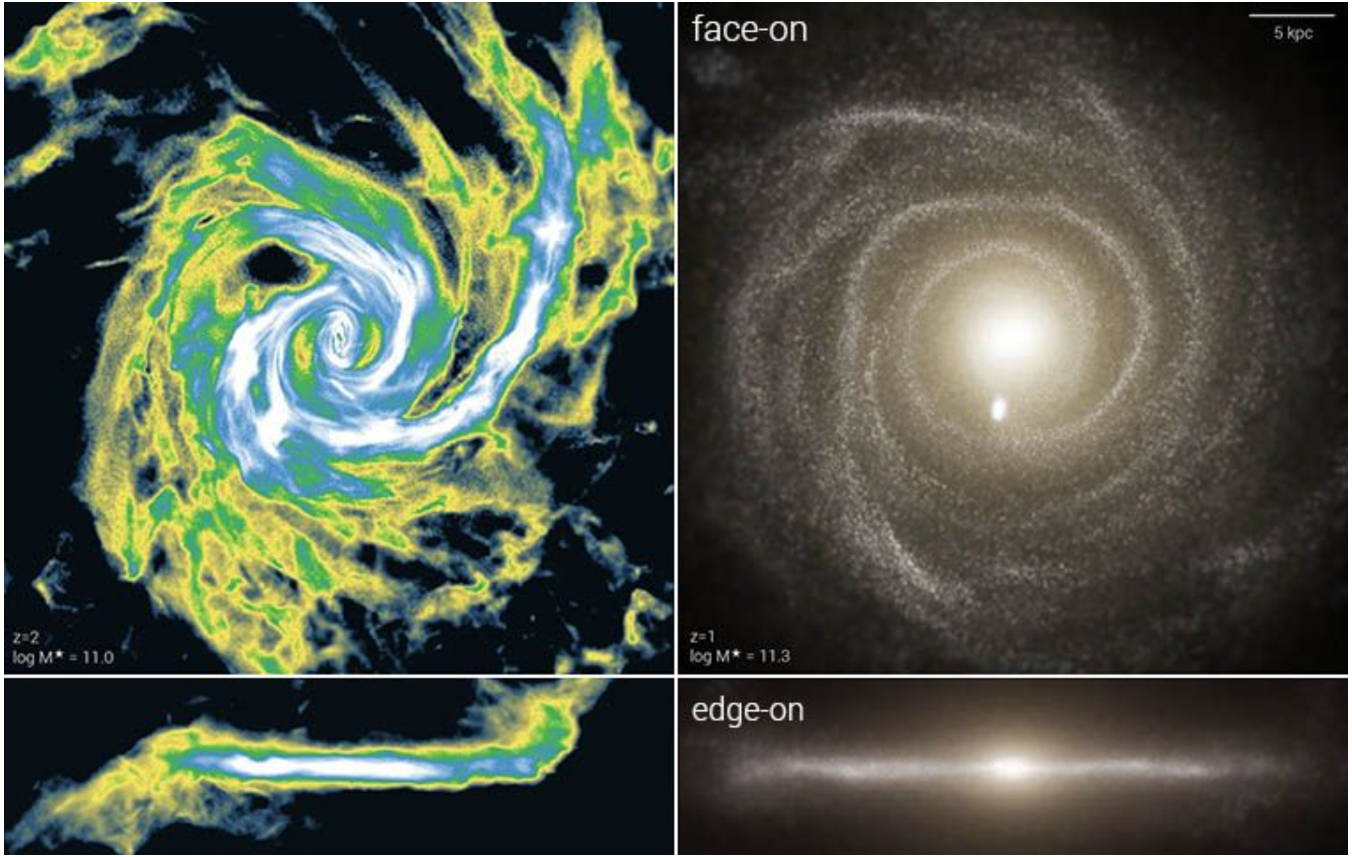
Frechet Inception Distance (FID) dist between real and generated images

IllustrisTNG



IllustrisTNG

Some of these do not use generative models. But then that is an opportunity! Make sure proper physics is incorporated.



Interpretability and explainability

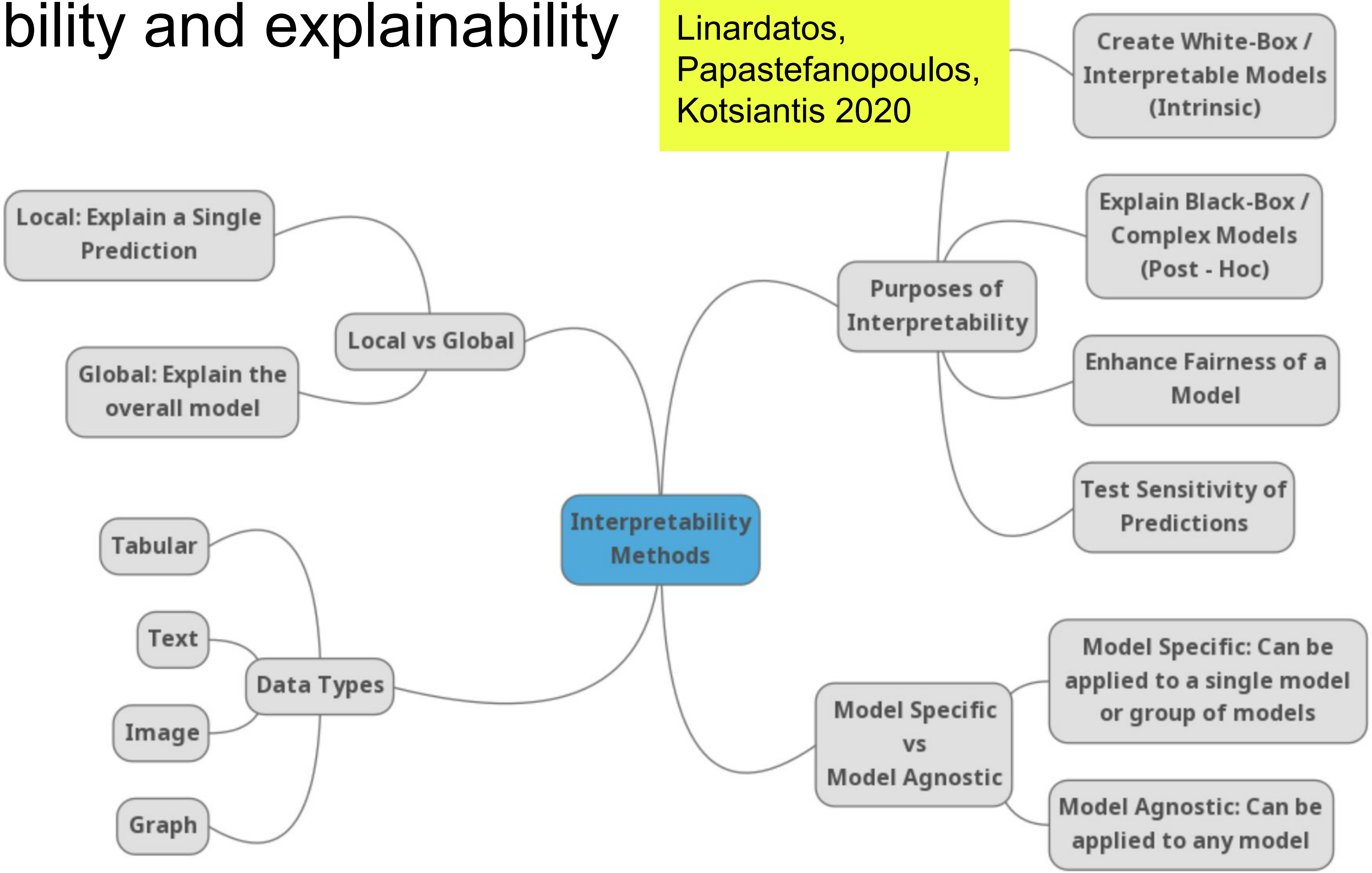
Linardatos, Papastefanopoulos, Kotsiantis 2020

Post-hoc

LIME: Local Interpretable Model-agnostic Explanations

SHAP: Shapley Additive Explanations

Fairness



ZARTH and its wild (ambiguous) classes

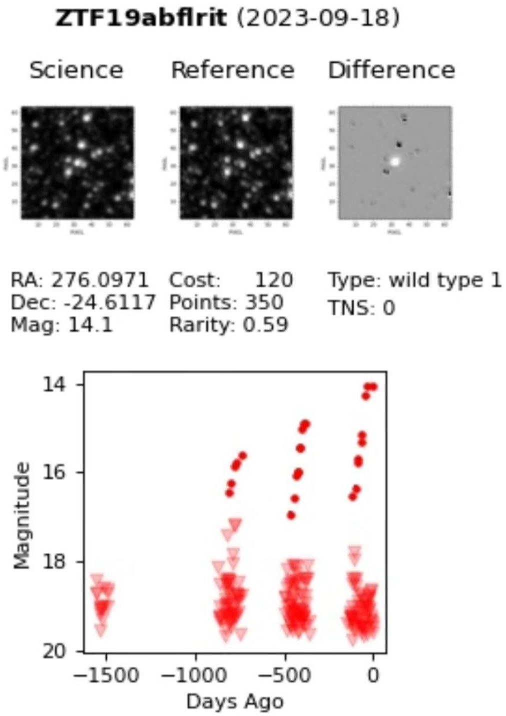
Hosted

Nuclear

Orphan

Variable

20-200 fresh ZTF transients every good night
Many gamification elemen
Points for catching
Leaderboards
Streaks
Badges coming soon



Wild Type 1
Wild Type 2

Game currently for outreach only for general public
Astronomy students can lear lot
And also provide feedback

Ideal to introduce in the classroom

Applications of Generative Models and Ethics

- **Applications of Generative AI**
  - In art and creativity (music, drawing)
  - In science (drug discovery, material science)
- **Session 2: Ethical Considerations**
  - Bias and fairness in AI models
  - Ethical implications of deepfakes and AI in surveillance

Trust and Ethics (Brian Green - Markakula Center for Applied Ethics)

What does it mean for something to be “trustworthy”? At the very least, it must be both technically trustworthy - it does what it is supposed to do - and ethically trustworthy - it does not violate ethical ideals necessary for trust (such as violating rights, deceiving, harming, exploiting users, etc.).

Tech empowers people to do new things. At the forward edges of human action people can act in ways that laws might not cover, but ethics does

Technology and Trust **(Brian Green - Markakula Center for Applied Ethics)**

1) Technological products should be technically trustworthy: • They are tools that should do what they are supposed to do

2) Technological products should be ethically trustworthy: • They should have the user’s best interests and the common good in mind, not exploit, deceive, violate, or otherwise harm people

The above are the minimum! Necessary, but not sufficient, for trust. Even if both are the case, technology can still create social distrust

Why Does Tech Harm Social Trust? **(Brian Green - Markakula Center for Applied Ethics)**

More Technology = More Power

More Power = More Choices

More Choices = More Responsibility

More Responsibility = More Need for Ethics

We were previously involuntarily constrained by our weakness • Now we must learn to be voluntarily constrained by our judgment • In other words, technological power turns socio-technical constants into variables (B. Srinivasan)

Shallow Summary

Generative AI entering all walks of life

VAEs, and transformers are very effective

Need to be mindful of unethical uses as well as distorted outputs.