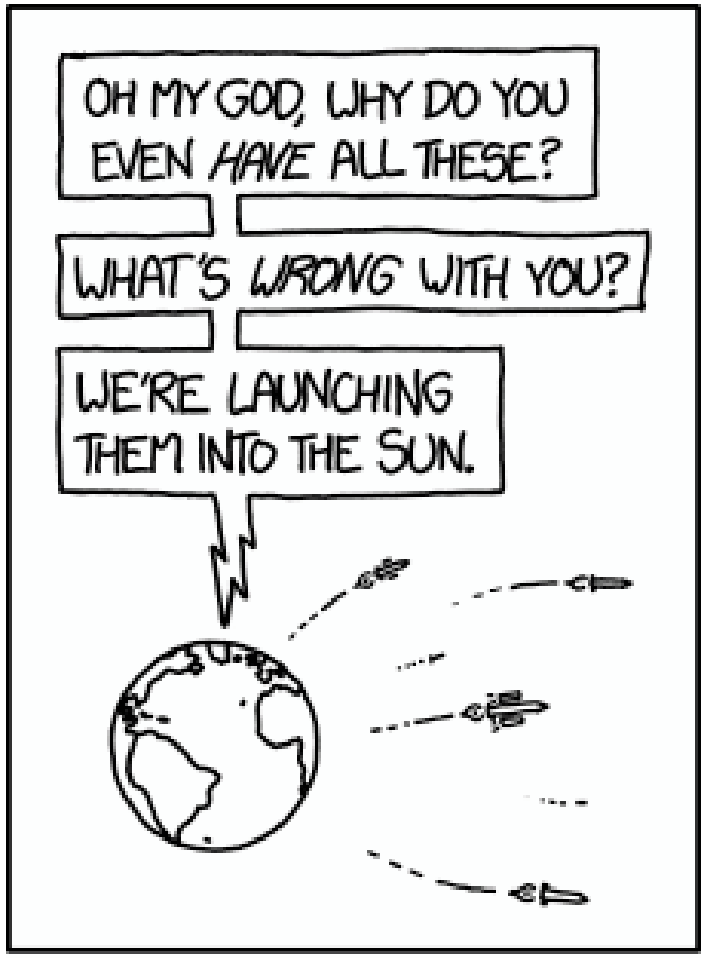
Generative AI:
media (world) generation



THE MOMENT THE COMPUTERS CONTROLLING OUR NUCLEAR ARSENALS BECAME SENTIENT

Autoencoder

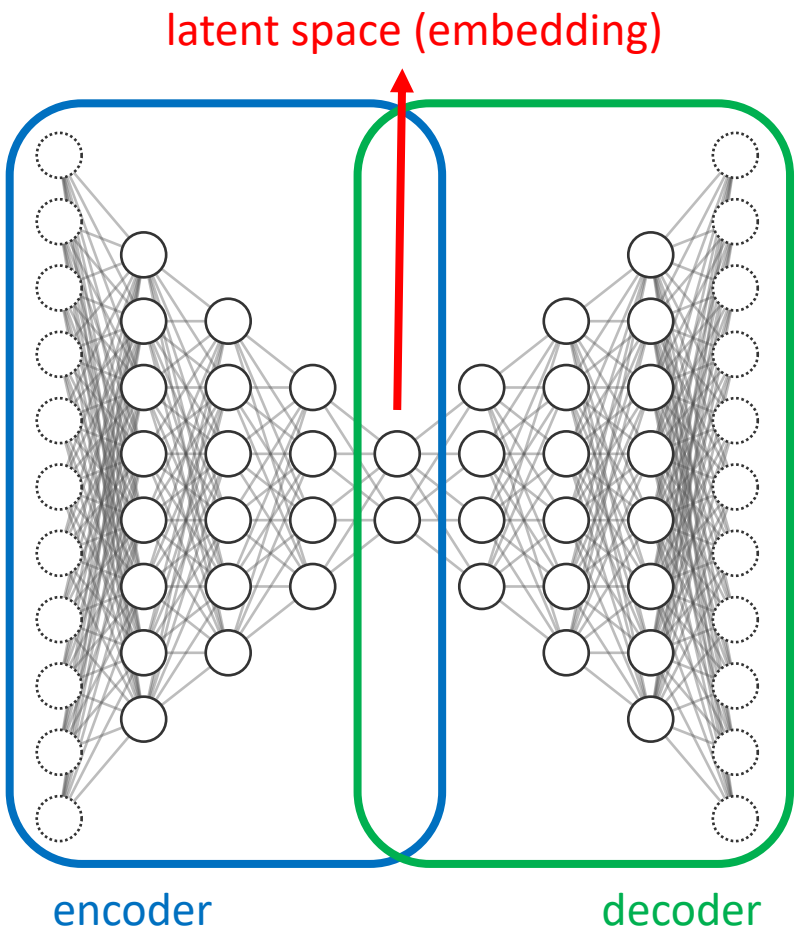
An **autoencoder** is a type of neural network used for **unsupervised learning**, designed to encode input data into a lower-dimensional representation and then reconstruct it back to its original form.

It consists of two main components: the **encoder**, which compresses the input into a latent space, and the **decoder**, which reconstructs the original input from this compressed representation.

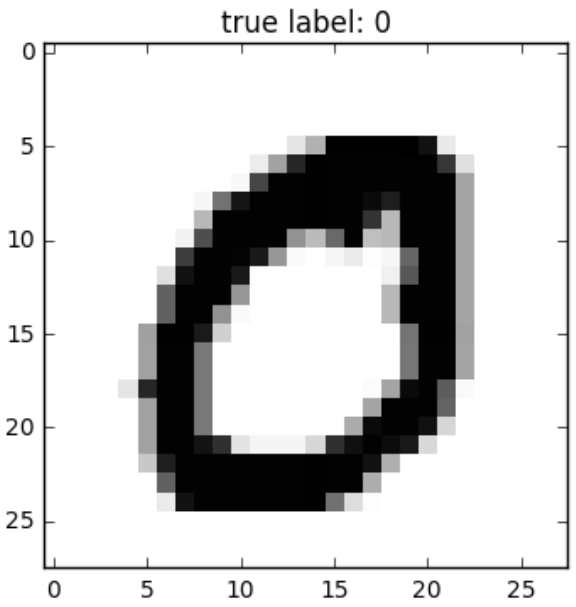
Autoencoder

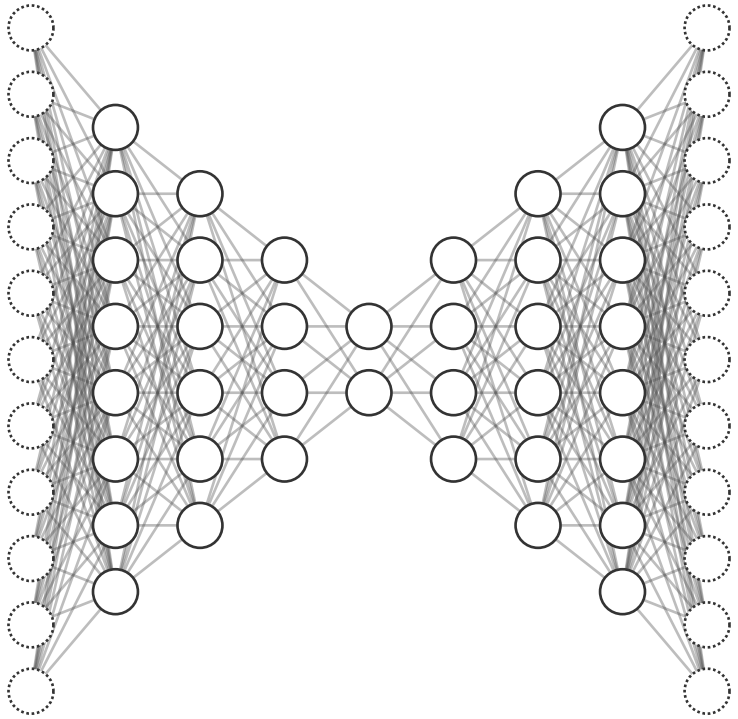
The encoder compresses the data from a higher-dimensional space to a lower-dimensional space (also called the **latent space**)

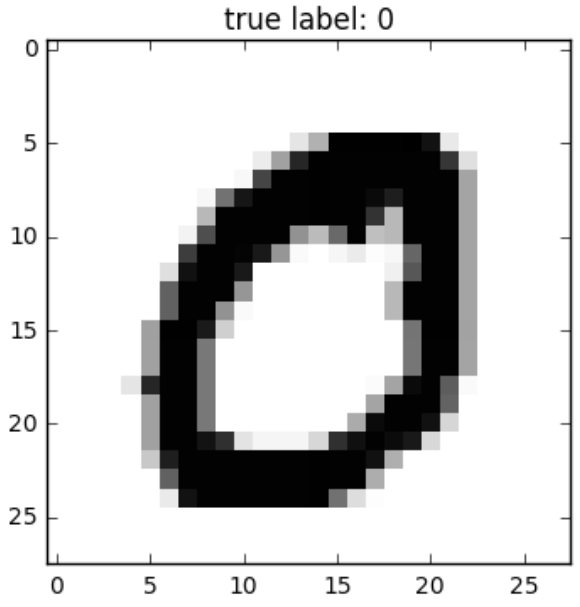
The decoder does the opposite i.e., convert the latent space back to higher-dimensional space.



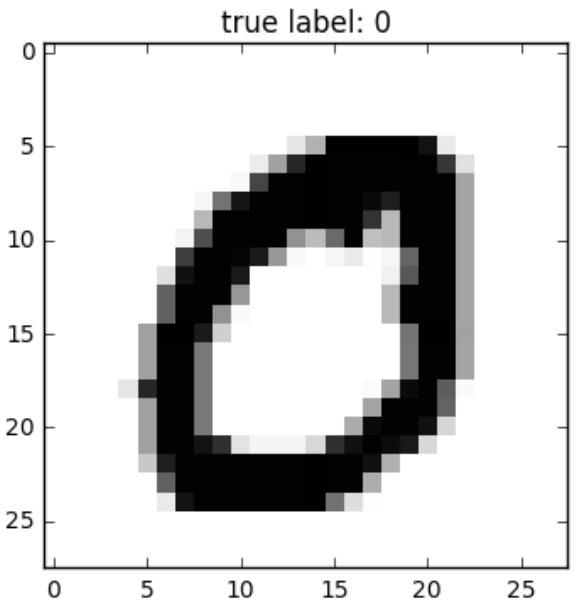
Autoencoder

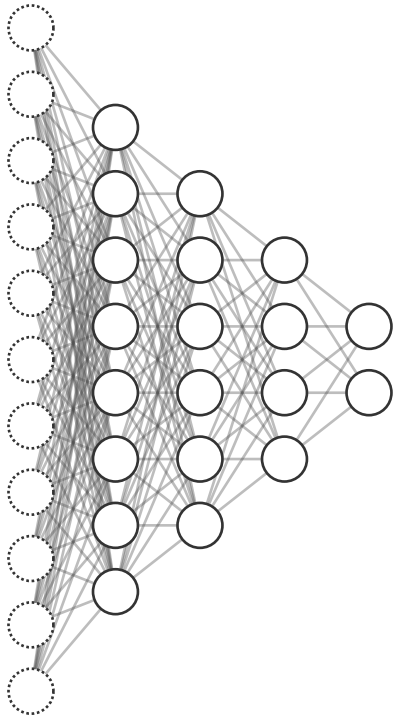


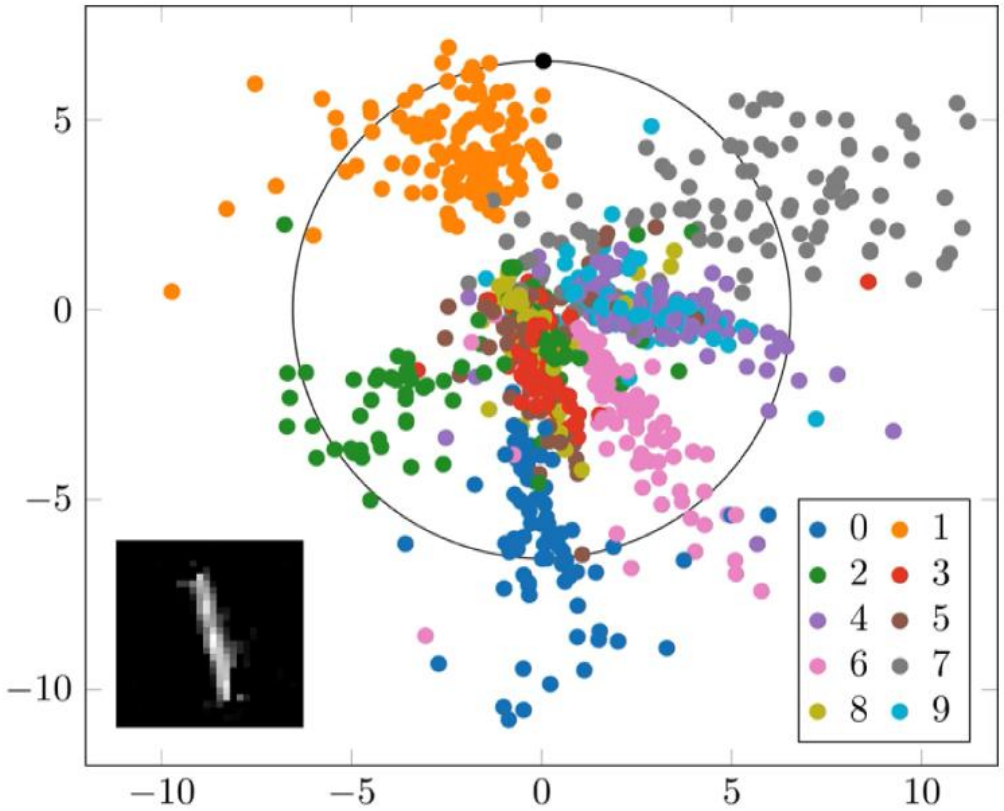




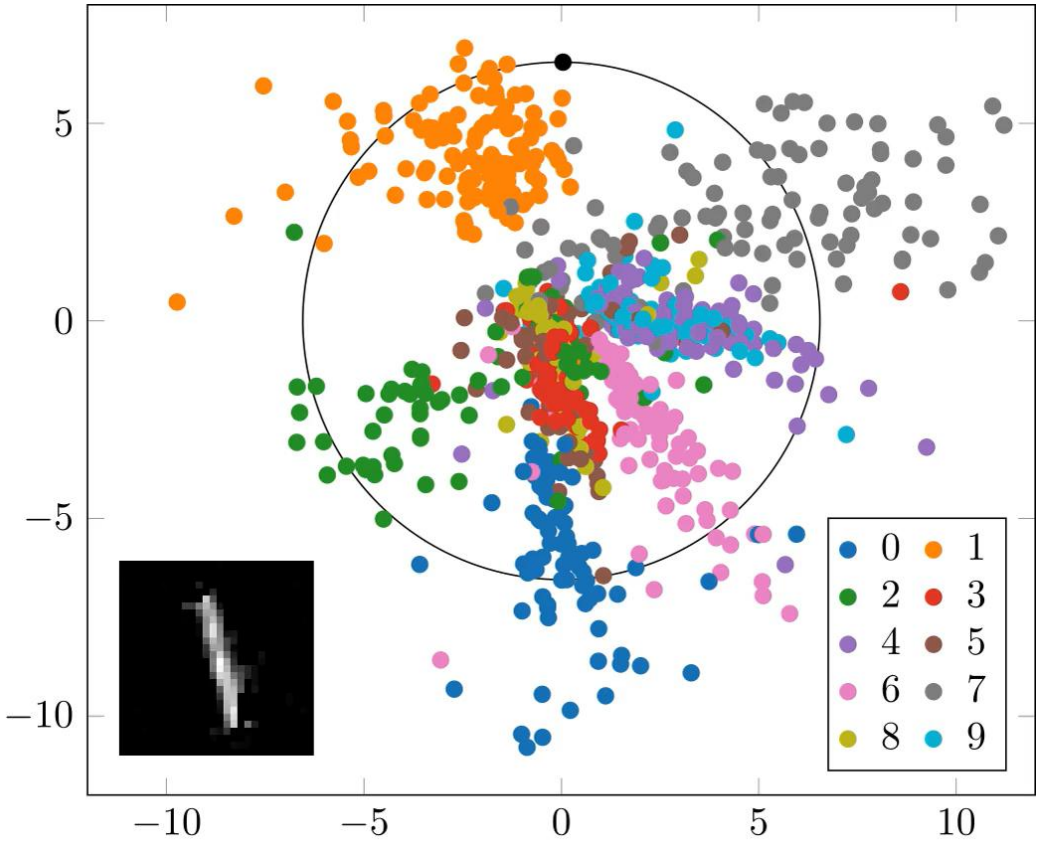
Autoencoder

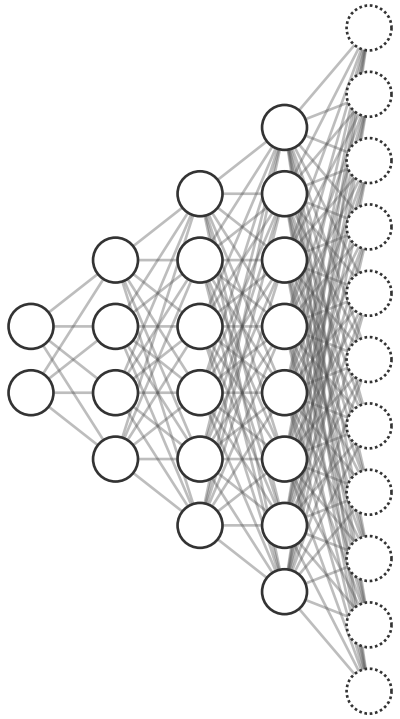


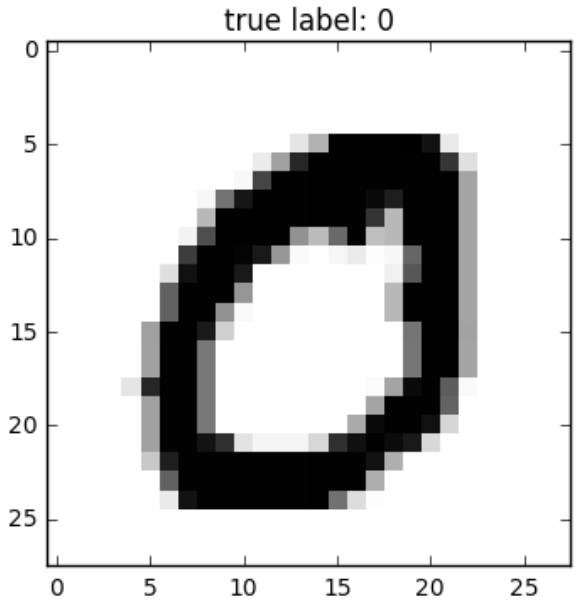




Autoencoder



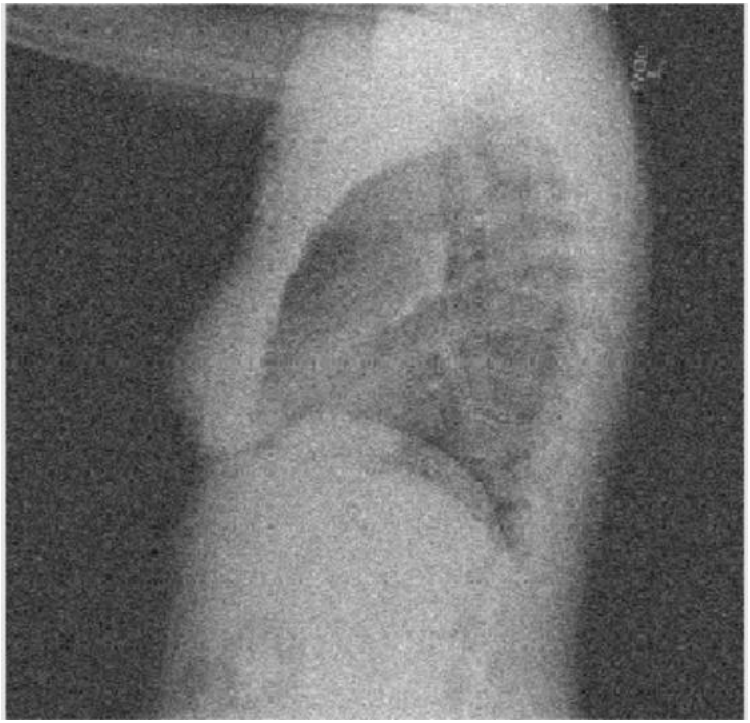


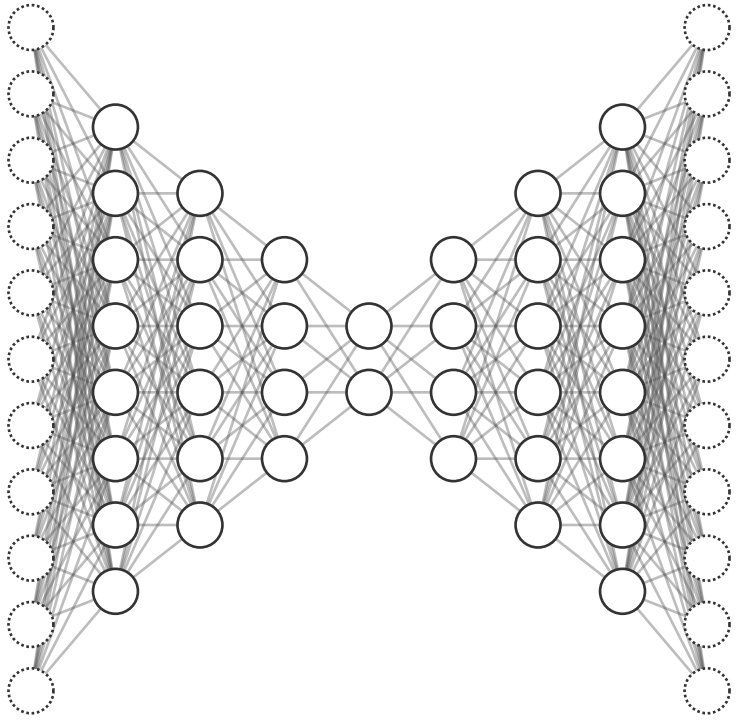


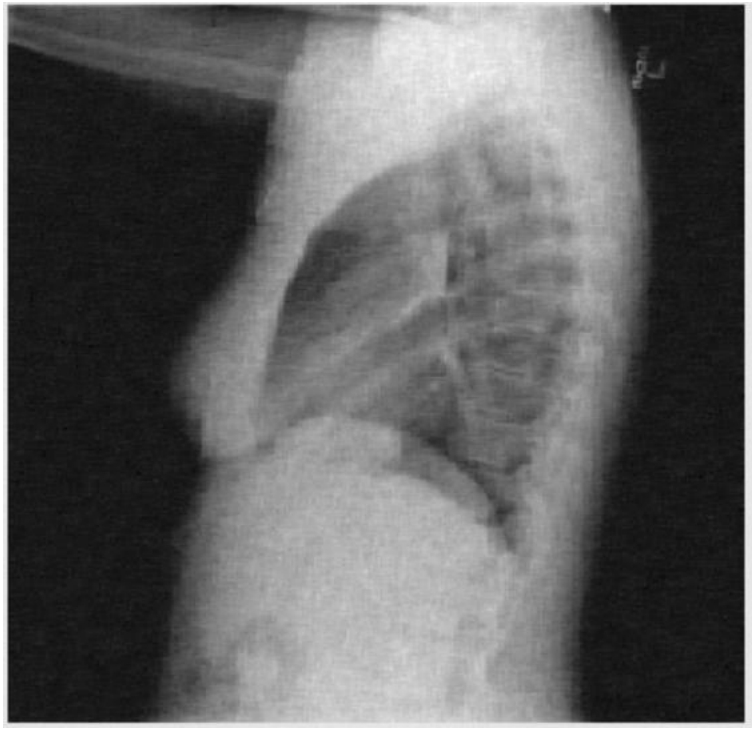
Denoising autoencoder

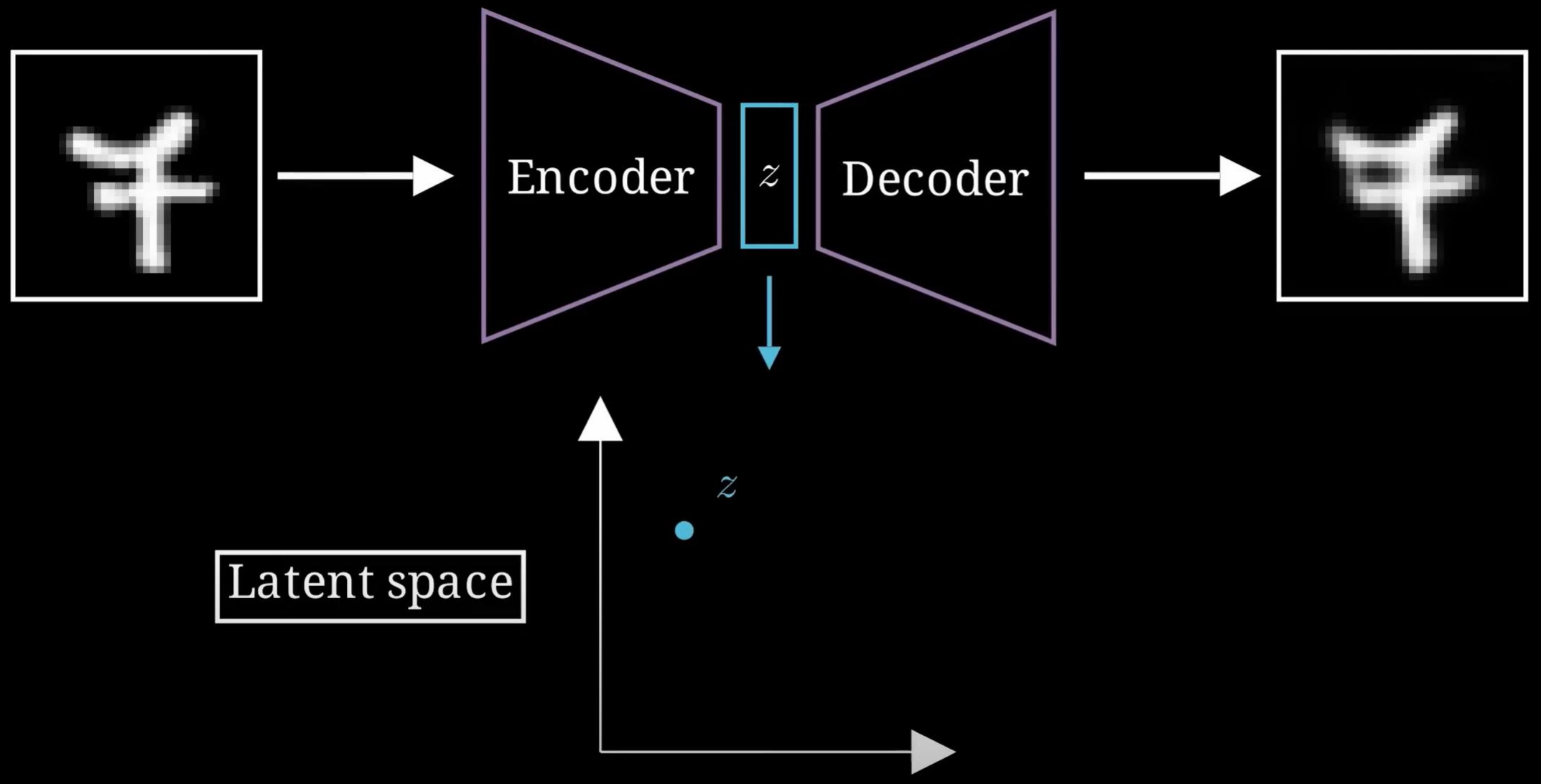
Autoencoders can easily be trained to **denoise** image data, expression data, …

The principle is to force the latent space layer capture the most robust features by reconstructing a clean input from a corrupted one.








Encoder
z
Decoder
z
Latent space

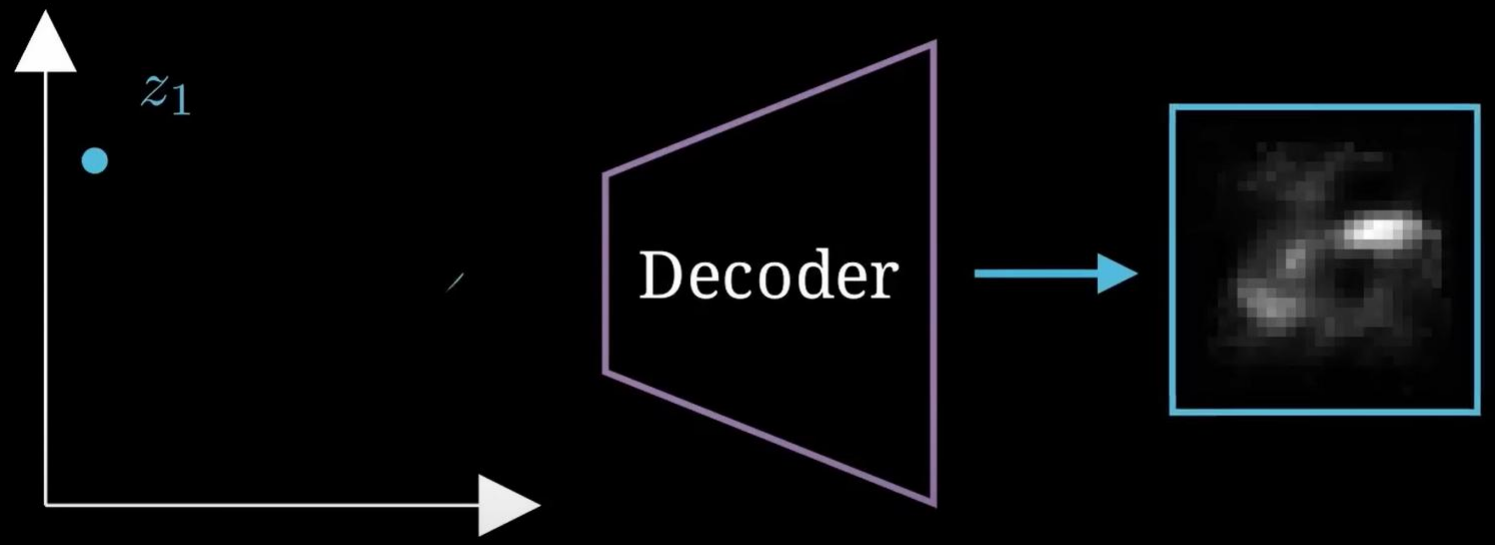
Autoencoder

Sparse and unstructured latent space: Normal autoencoders encode each input into a **single fixed point** in the latent space without enforcing a continuous distribution.

As a result, **interpolating** between different points often leads to unrealistic or meaningless outputs.

Since the encoder directly maps inputs to arbitrary points, sampling from the latent space (to generate new data) is unreliable.

There is no guarantee that points between learned encodings correspond to meaningful variations of the data.

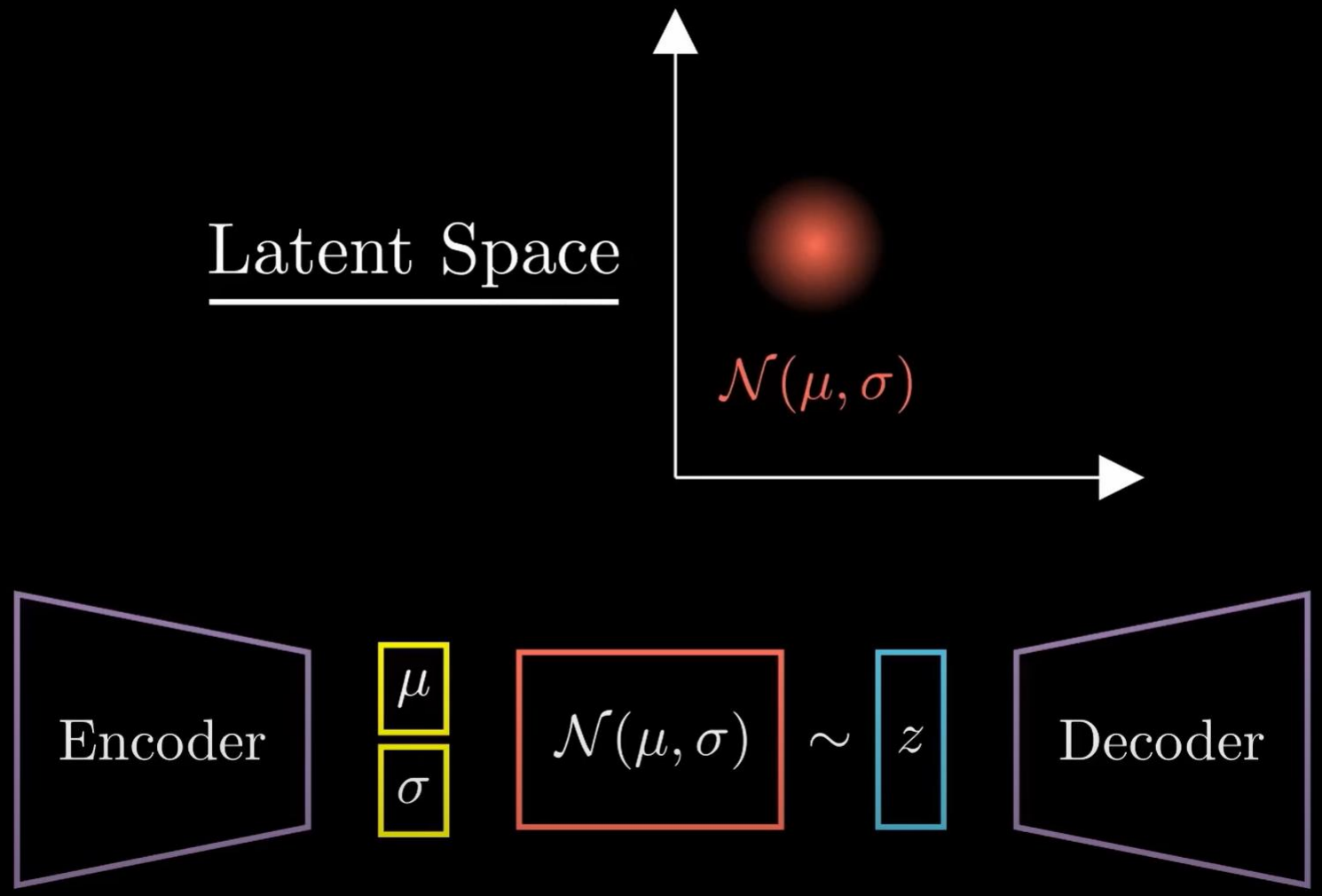

z_1
Decoder

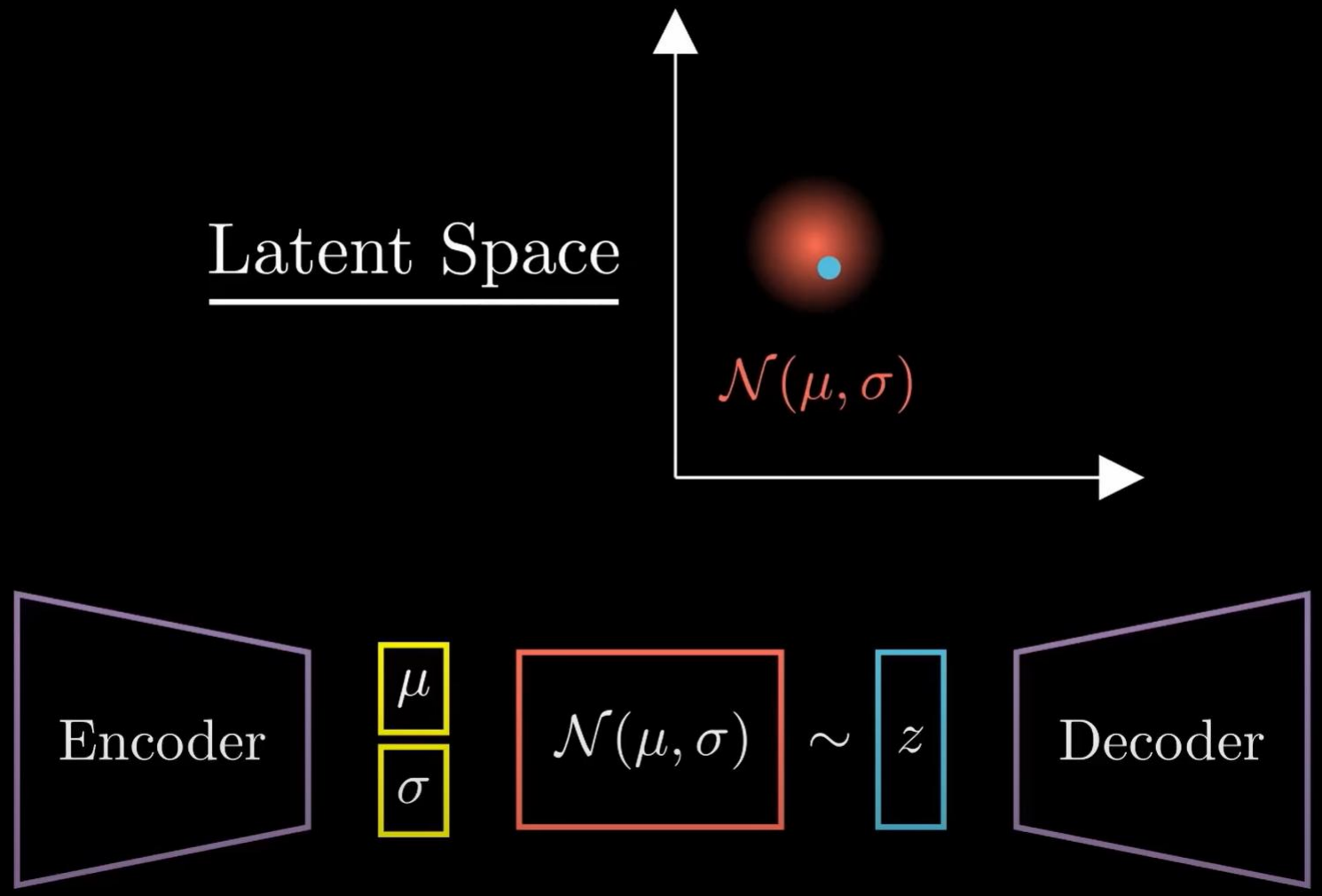
Variational autoencoder (VAE)

Latent space representation: A VAE learns a **probabilistic distribution** of the input data in a lower-dimensional latent space, rather than mapping inputs to fixed points like traditional autoencoders.

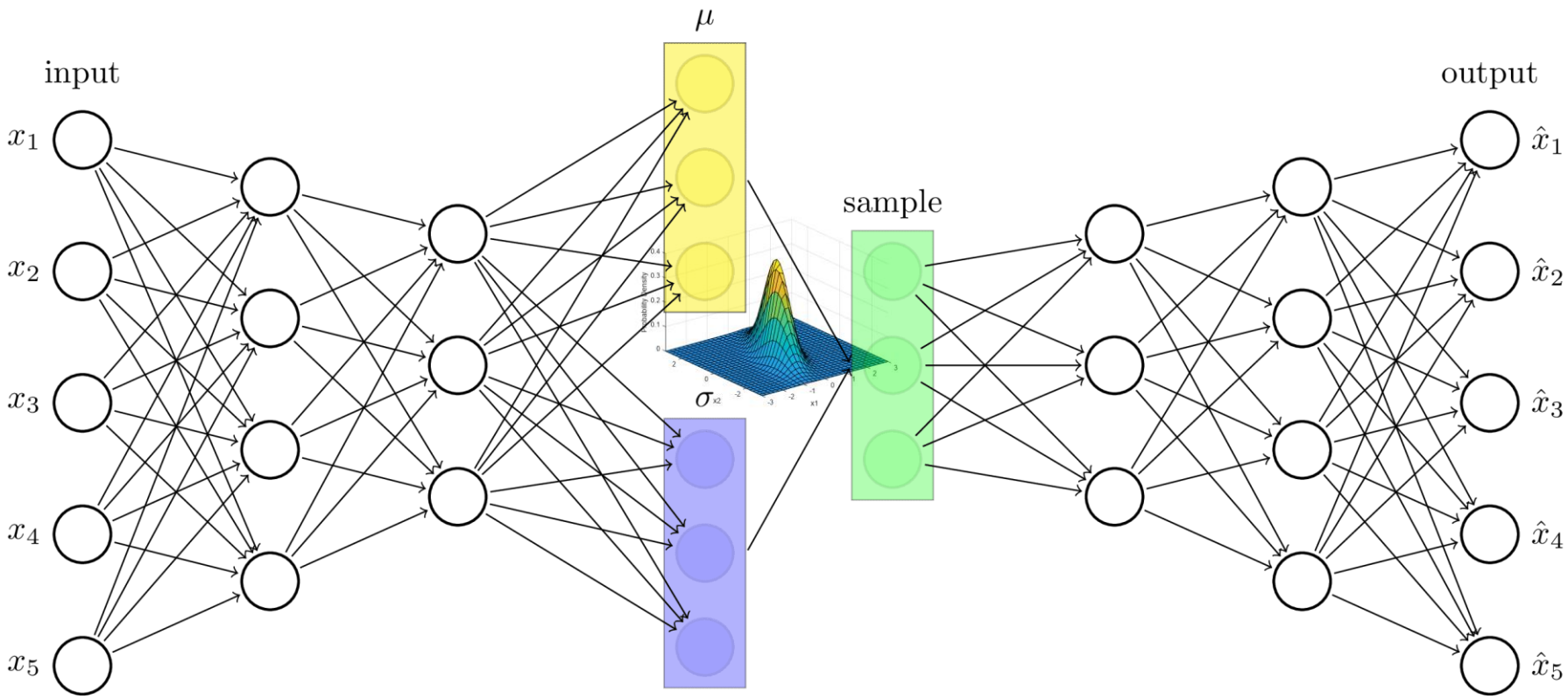
The encoder transforms inputs into a probability distribution (mean & variance) in the latent space, while the decoder reconstructs data by sampling from this distribution.

VAEs use a Kullback-Leibler (KL) divergence loss to ensure that the learned latent distribution remains close to a **standard normal distribution**, promoting smoothness and continuity in the latent space.


Latent Space
$\mathcal{N}(\mu, \sigma)$
Encoder
μ
σ
$\mathcal{N}(\mu, \sigma)$
$\sim$
z
Decoder


Latent Space
$\mathcal{N}(\mu, \sigma)$
Encoder
μ
σ
$\mathcal{N}(\mu, \sigma)$
$\sim$
z
Decoder

Variational autoencoder (VAE)



VAE: applications

Dimensionality reduction: VAEs can compress high-dimensional biomedical data (e.g., gene expression, medical images) into a lower-dimensional latent space while preserving meaningful features.

Denoising & data imputation: VAEs can reconstruct missing or corrupted biomedical data, improving the quality of noisy datasets like MRI scans, ECG signals, or gene expression profiles.

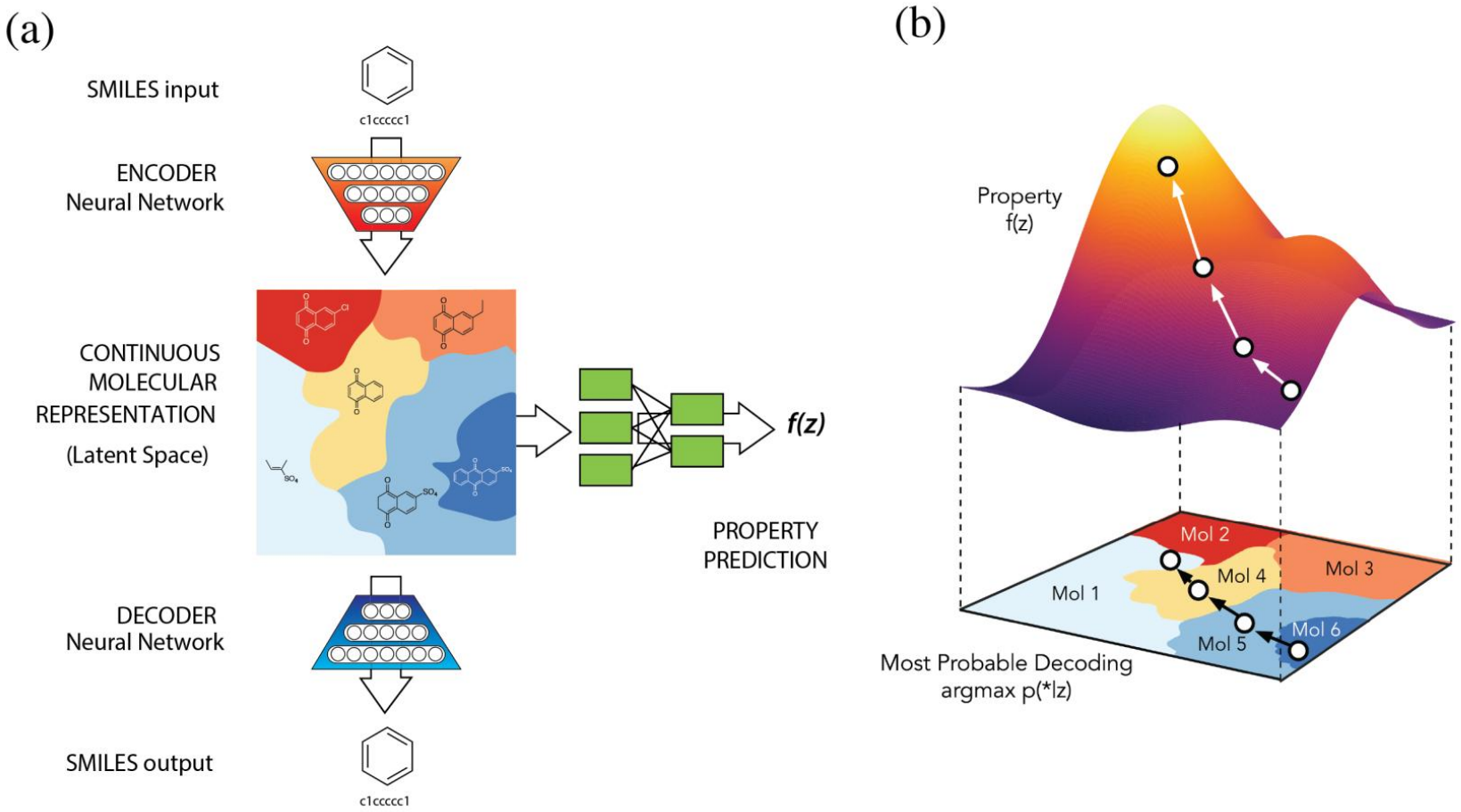
Anomaly detection: By learning normal data distributions, VAEs can identify anomalies in biomedical datasets, aiding in early disease detection (e.g., identifying tumors in medical imaging).

Data privacy: VAEs can be used for privacy-preserving data sharing by generating synthetic patient data that retain statistical properties without exposing real patient records.

Gómez-Bombarelli et al. ACS Central Science (2019)

VAE: drug discovery

Representing chemicals in latent space to search for chemicals that optimize some function (e.g. maximize activity or minimize toxicity).



Gómez-Bombarelli et al. ACS Central Science (2019)

Generative adversarial network (GAN)

A VAE learns a probabilistic distribution of the input data in a lower-dimensional latent space, rather than mapping inputs to fixed points like traditional autoencoders.

GANs generate sharper, more detailed images than VAEs, which often produce blurry results due to probabilistic constraints.

GANs do not impose a Gaussian prior, allowing for richer and more diverse data generation compared to VAEs.

GANs excel in generating high-quality synthetic biomedical images and unsupervised feature learning, whereas VAEs are better for structured latent representations and anomaly detection.

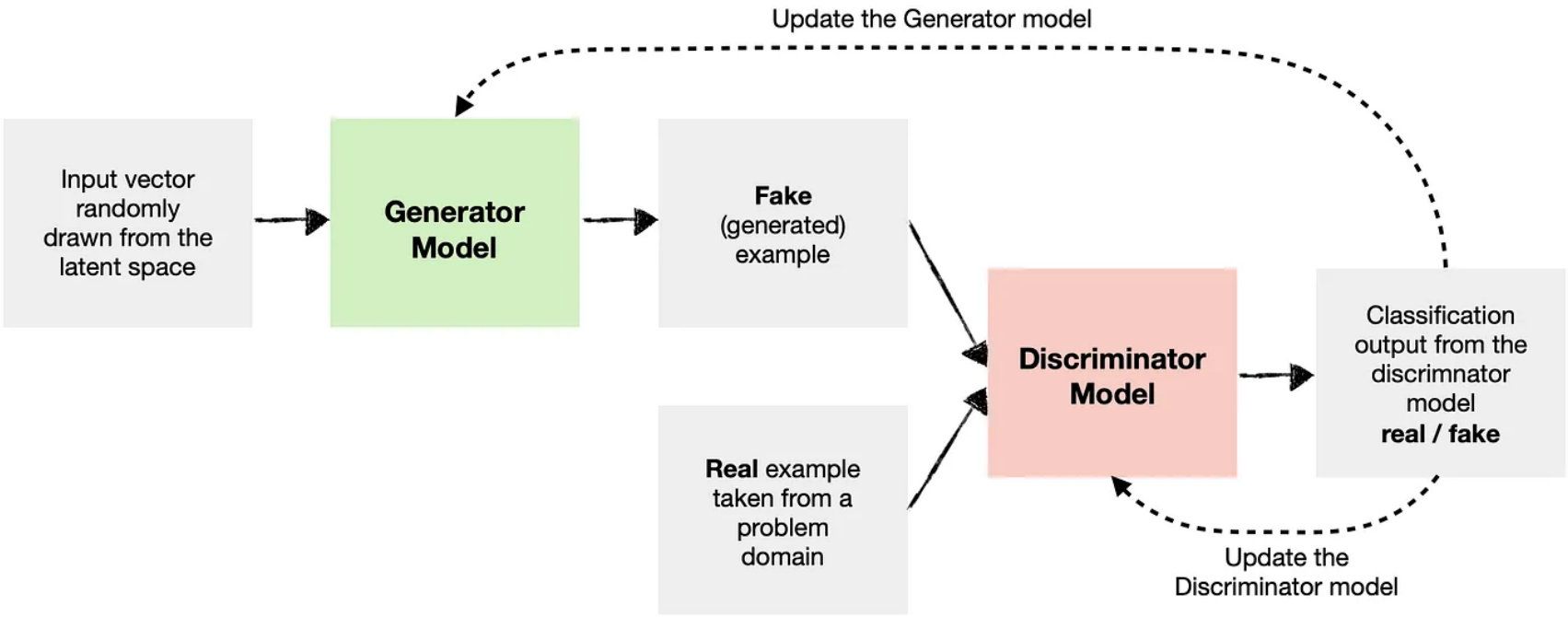
Generative adversarial network (GAN)

The **generator** CNN learns to generate new data (i.e., fake data) like that of the problem domain.

The **discriminator** CNN learns to identify whether the provided example is fake (comes from a generator) or real (comes from the actual data domain).

Both the generator and the discriminator are **trained together** by minimizing one loss function.

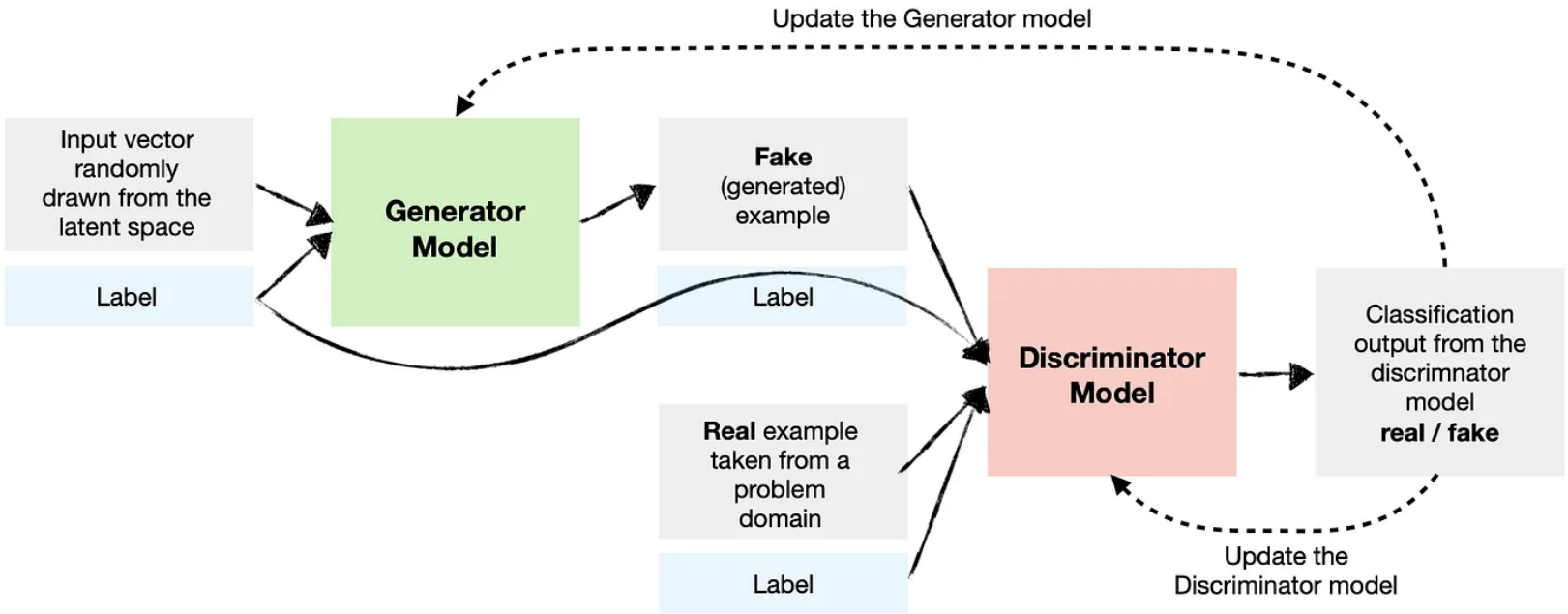
.



Conditional GAN (cGAN)

A cGAN extends standard GANs by incorporating **extra input** (e.g., class labels, text descriptions, or images) to **guide** the generation process, ensuring more controlled outputs.

The generator takes both random noise and a conditioning variable to produce relevant outputs, while the discriminator evaluates whether the generated sample matches the conditioning input.



Conditional GAN (cGAN)

Conditioning allows the GAN to generate medical images that match specific disease conditions, patient attributes, or imaging modalities, ensuring the synthetic data is medically meaningful rather than random.

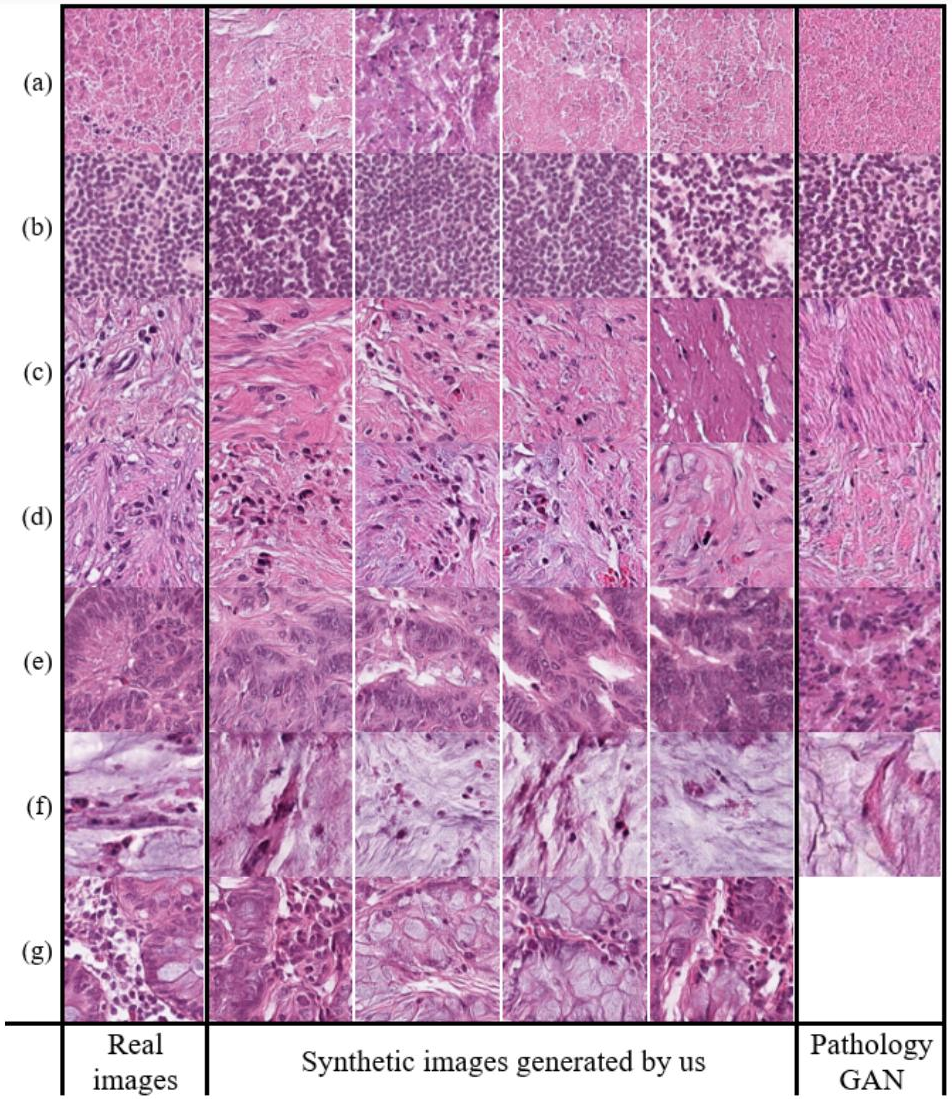


Fig. 3: Examples of real and synthetic images for NCT-CRC-HE-100K colorectal cancer dataset: (a) DEB, (b) LYM, (c) MUS, (d) STR, (e) TUM, (f) MUC, and (g) NORM.

Source A: gender, age, hair length, glasses, pose

Source B: everything else

Result of combining A and B

Diffusion model

Diffusion models generate data by **gradually transforming pure noise into structured outputs** through a series of small, incremental denoising steps.

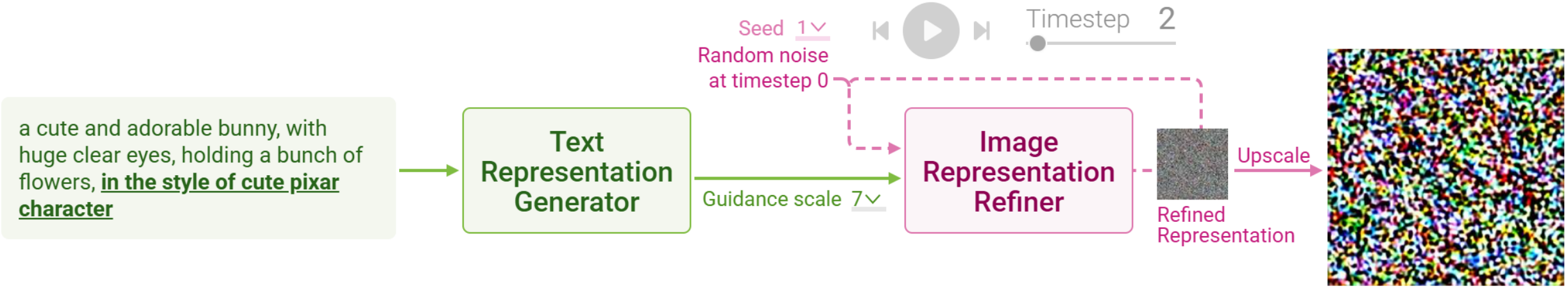
They leverage a probabilistic model that learns the underlying data distribution by reversing the forward process of adding noise.

By refining details over multiple iterations, diffusion models can produce highly detailed and realistic images.

Their training strategy focuses on **learning to denoise**, which often results in stable and reliable convergence during the learning process.

Diffusion model

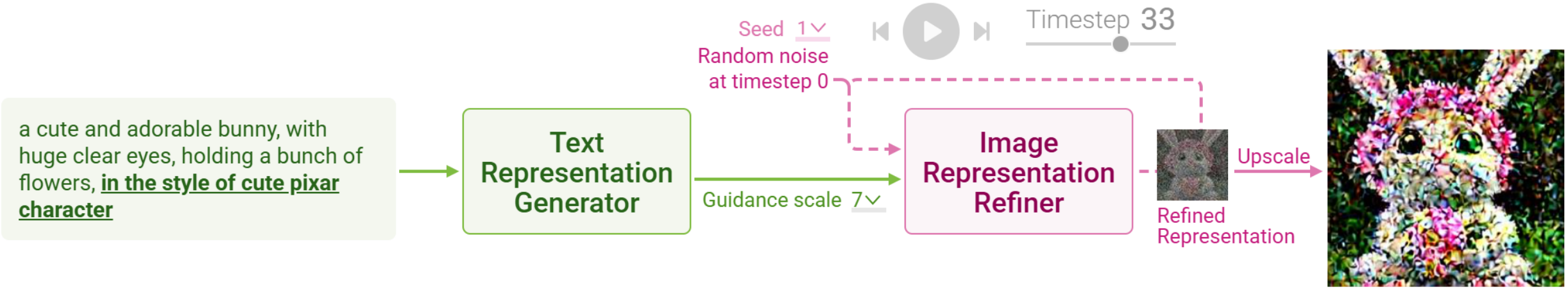
The process involves a **Text Representation Generator** that converts the input prompt into an initial representation, followed by an **Image Representation Refiner** that enhances the image over multiple steps using noise reduction and a guidance scale.



https://poloclub.github.io/diffusion-explainer/

Diffusion model

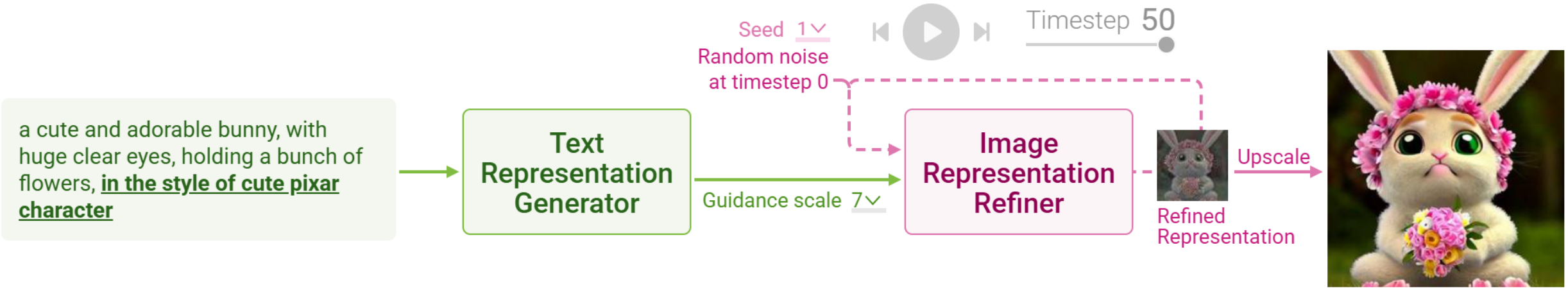
The process involves a **Text Representation Generator** that converts the input prompt into an initial representation, followed by an **Image Representation Refiner** that enhances the image over multiple steps using noise reduction and a guidance scale.



https://poloclub.github.io/diffusion-explainer/

Diffusion model

The process involves a **Text Representation Generator** that converts the input prompt into an initial representation, followed by an **Image Representation Refiner** that enhances the image over multiple steps using noise reduction and a guidance scale.



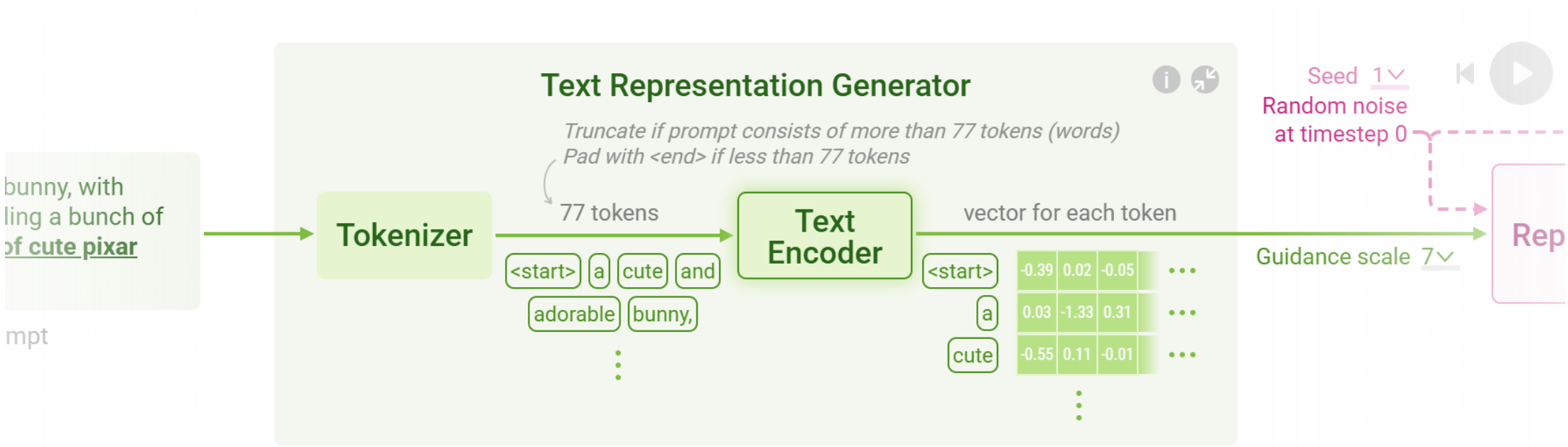
https://poloclub.github.io/diffusion-explainer/

Diffusion model

The **Text Encoder** converts each token into a numerical representation (embedding).

Each word is mapped to a high-dimensional vector, as shown in the matrix with numerical values for words like "cute" and "bunny."

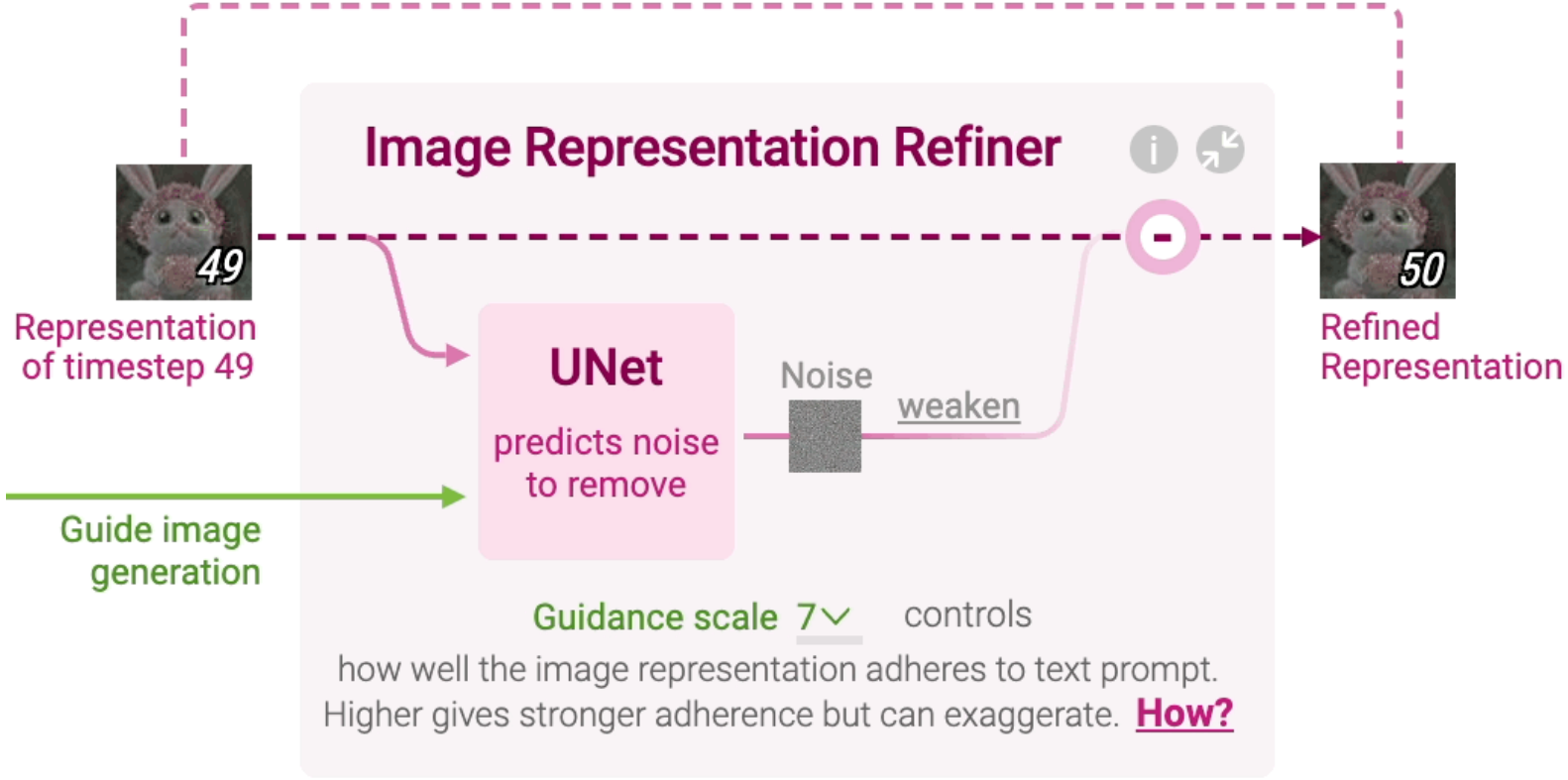
The generated token vectors are fed into the **Image Representation Refiner** (next stage), which uses them to guide the creation of an image.



https://poloclub.github.io/diffusion-explainer/

Diffusion model

The **Image Representation Refiner** uses a U-Net model to predict and remove noise from an intermediate image. This process gradually improves the image quality by iteratively reducing noise.

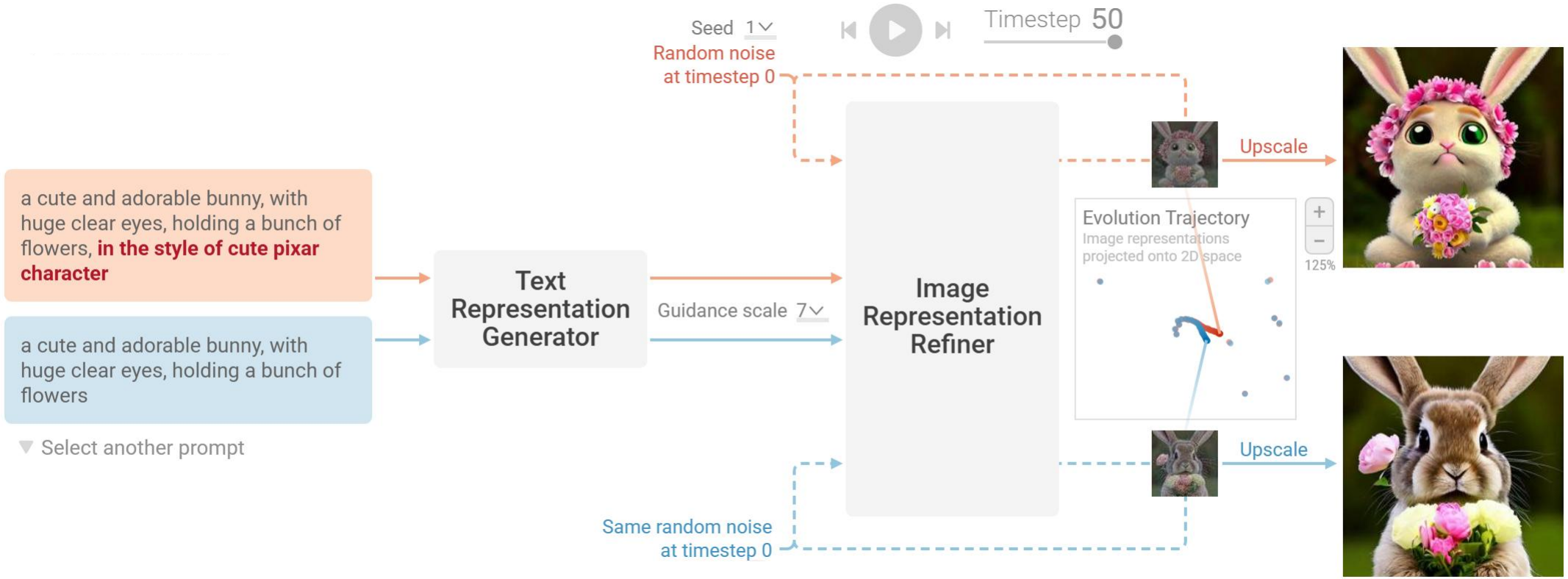


https://poloclub.github.io/diffusion-explainer/

Contrastive Language-Image Pretraining (CLIP)

CLIP helps diffusion models understand how well an image matches a given text prompt by embedding both text and images into a shared feature space.

The diffusion model uses this **guidance** to iteratively refine the image toward better alignment with the prompt.

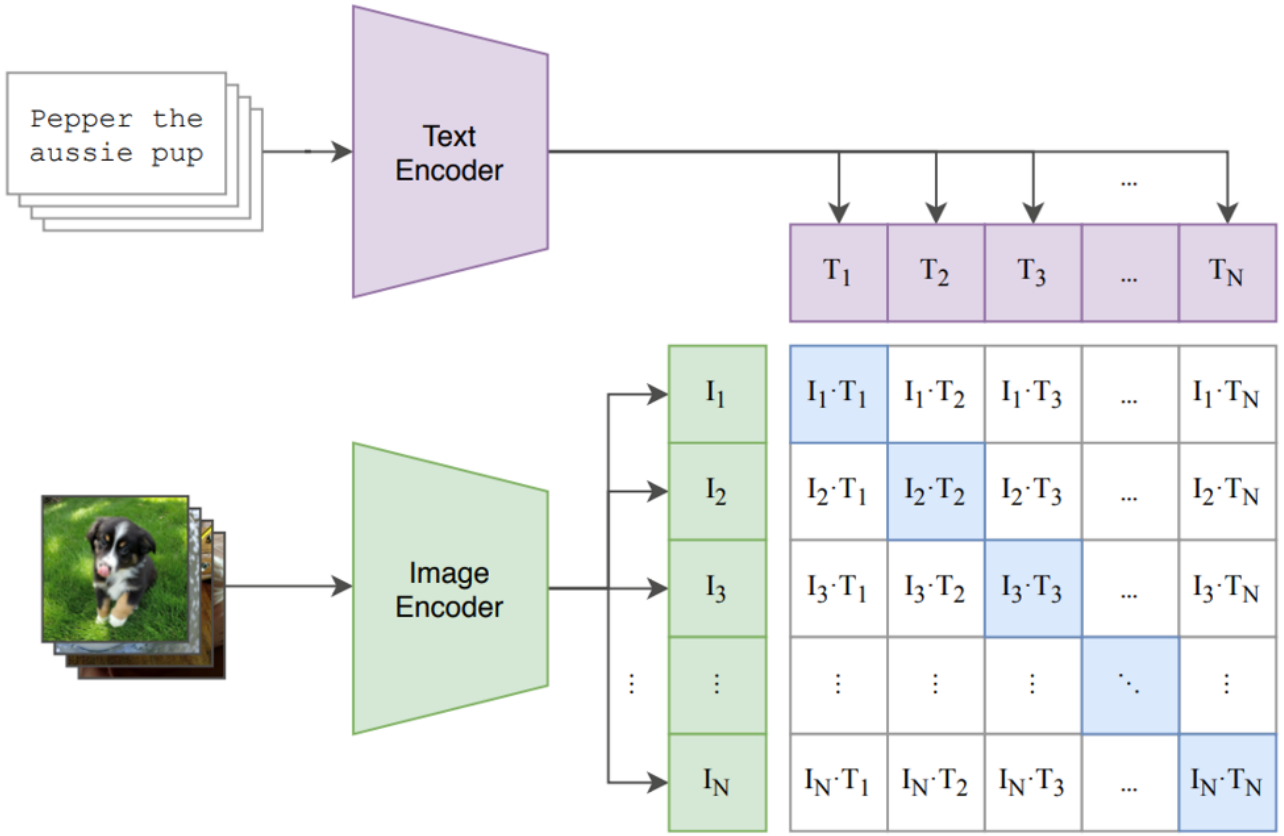


https://poloclub.github.io/diffusion-explainer/

Contrastive Language-Image Pretraining (CLIP)

CLIP is trained on a large dataset of image-text pairs to learn a shared embedding space, where both images and textual descriptions are represented as vectors **in that same space**.

It uses a **contrastive loss** that aligns the embeddings of matching image-text pairs while pushing non-matching pairs apart, effectively capturing semantic relationships.



Radford, A. et al. (2021). Learning Transferable Visual Models From Natural Language Supervision, ICML, PMLR 139, 8748–8763

Data: LAION datasets

LAION-400M comprises about 400 million image–text pairs, while LAION-5B scales up to roughly 5 billion pairs, making them some of the largest open datasets for text-to-image tasks.

Both datasets are sourced from extensive web crawls, ensuring a wide variety of visual content and textual descriptions across numerous domains and topics.

These datasets feature a diverse range of languages—with approximately 50% of the pairs in English and the remainder spanning over 80 other languages—facilitating the training of globally applicable models.

french cat

french cat

How to tell if your feline is french. He wears a b...

イケメン猫モデル「トキ・ナンタケット」がかっこいい - NAVER まとめ

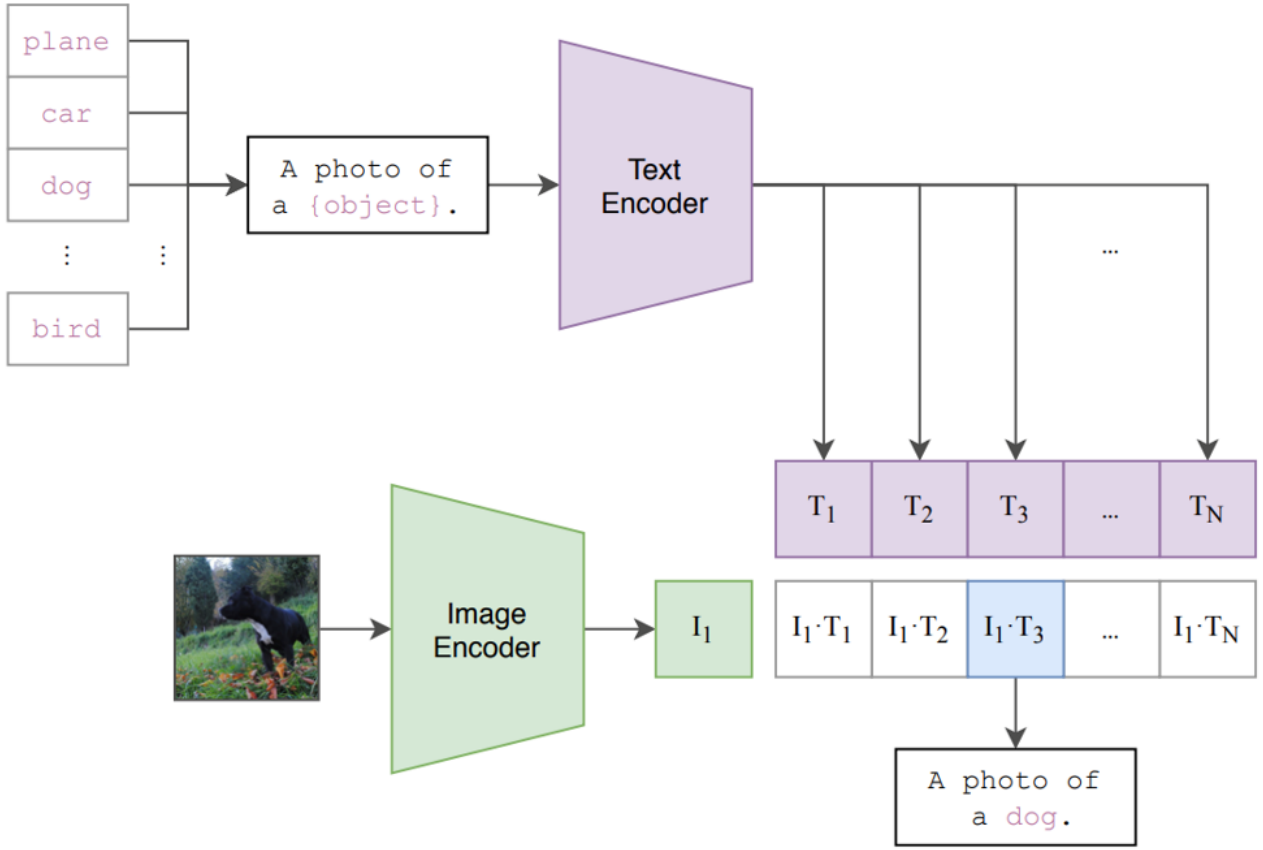
Hilarious pics of funny cats! funnycatsgif.com

Schuhmann, C., et al. (2021). LAION-400M: Open Dataset of 400 Million Image-Text Pairs. Zenodo. doi:10.5281/zenodo.3779943

Zero-shot image classification with CLIP

CLIP learns a **shared representation** for images and text by training on large amounts of image–text pairs, enabling it to match an unseen image with relevant textual descriptions without explicit training on specific categories.

By comparing an image's embedding to embeddings of class labels or prompts, CLIP can identify the most semantically similar label, allowing it to classify images into new categories it was **never directly trained on**.

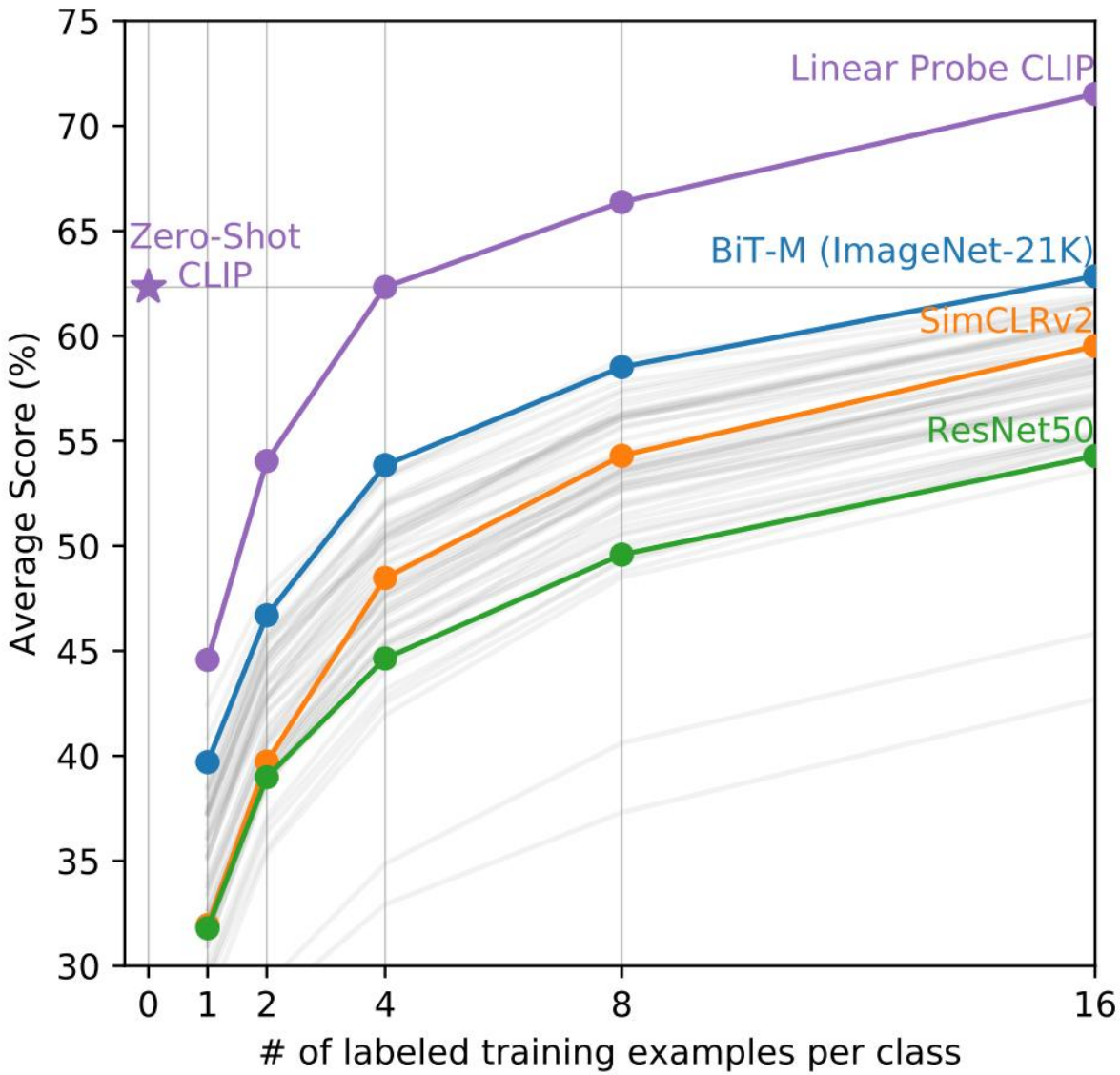


Radford, A. et al. (2021). Learning Transferable Visual Models From Natural Language Supervision, ICML, PMLR 139, 8748–8763

Zero-shot image classification with CLIP

CLIP learns a **shared representation** for images and text by training on large amounts of image–text pairs, enabling it to match an unseen image with relevant textual descriptions without explicit training on specific categories.

By comparing an image's embedding to embeddings of class labels or prompts, CLIP can identify the most semantically similar label, allowing it to classify images into new categories it was **never directly trained on**.



Radford, A. et al. (2021). Learning Transferable Visual Models From Natural Language Supervision, ICML, PMLR 139, 8748–8763

Zero-shot image classification with CLIP

GitHub

CLIP_image_classification.ipynb

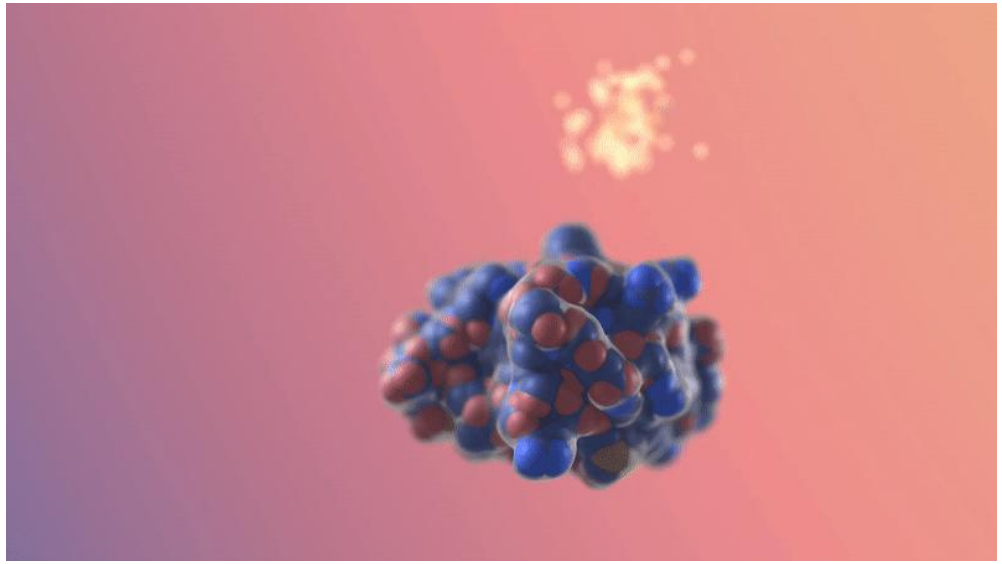
Diffusion models: RFdiffusion

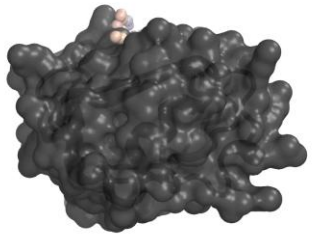
RFdiffusion begins with random noise and progressively refines it through a reverse diffusion process to form coherent protein structures.

It integrates structural priors from pre-trained ensuring generated conformations adhere to realistic protein geometry.

The process is conditioned on specific design objectives—like binding sites or active motifs—to tailor the protein output to functional requirements.

By combining diffusion-based refinement with learned protein structure knowledge, RFdiffusion enables the generation of novel, physically plausible protein designs.



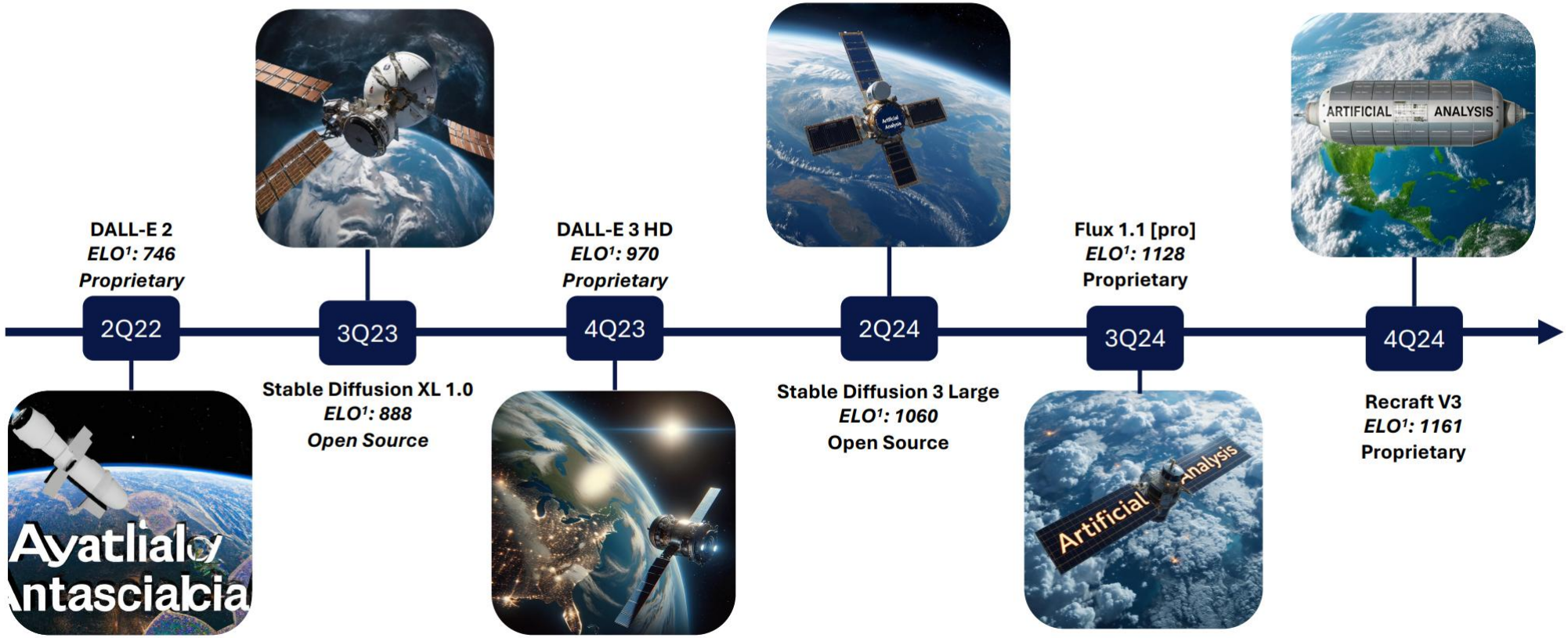


Watson, J.L., et al. (2023). De novo design of protein structure and function with RFdiffusion. Nature, 620, 687–694.

Image generation quality progressed rapidly in 2024, with notable leaps in photorealism, prompt adherence, and text rendering

Image Generation Quality, Over Time

Prompt: The words 'Artificial Analysis' emblazoned on a next-generation spacecraft orbiting a breathtaking Earth view



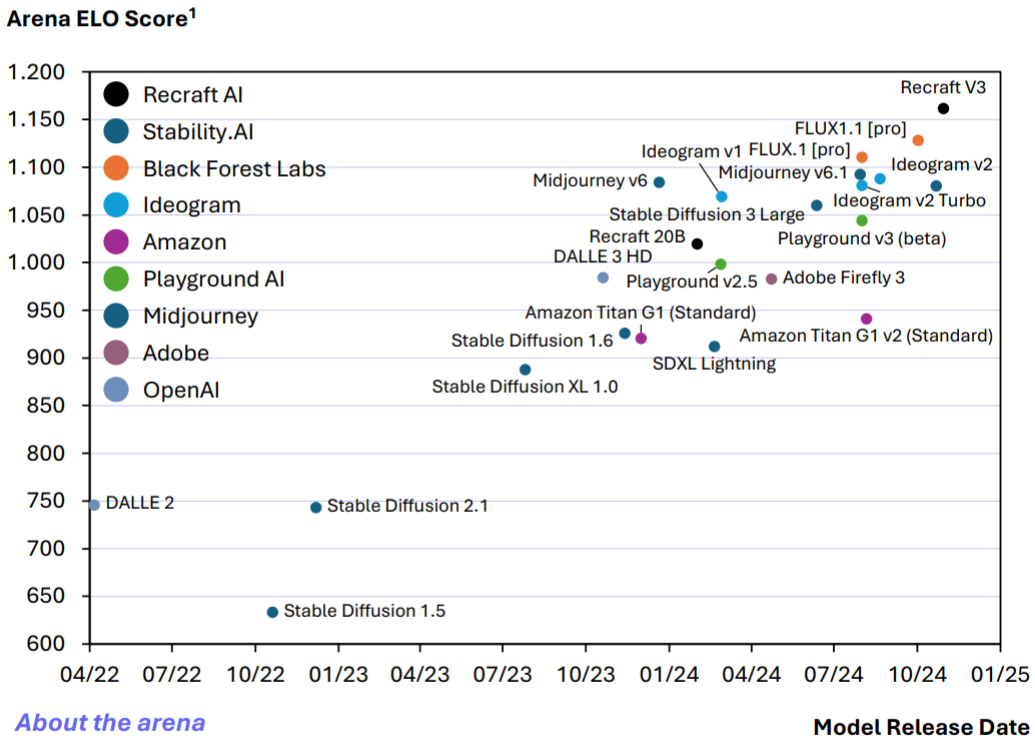
Source: Artificial Analysis Image Arena

1. Artificial Analysis Image Arena ELO, calculated as of 16 December 2024 based on 1.6 million votes from Artificial Analysis users

Artificial Analysis

Progress and competition for image models accelerated over the course of 2024; The top 5 models in the Artificial Analysis Image Arena all launched since Q3 2024

Image Arena ELO vs. Model Release Date



About the arena

Image Arena Leaderboard – Top 5 Models

Creator	Model	Release Date	Arena ELO
Recraft AI	Recraft v3	OCT 2024	1161
Black Forest labs	FLUX1.1 [pro]	OCT 2024	1128
Black Forest labs	FLUX.1 [pro]	AUG 2024	1110
Midjourney	Midjourney v6.1	JUL 2024	1092
Ideogram	Ideogram v2	AUG 2024	1088

The Artificial Analysis Image Arena uses crowdsourced preferences to assess text to image model quality. As of December 2024, over 1.5 million preferences have been gathered. These preferences are used to compute Arena ELO scores, which reflect the relative quality of image models.

1. As at 4 October 2024. 2. FLUX 1.1 [pro] was tested on the Image Arena during 3Q24 under the pseudonym Blueberry

Artificial Analysis