Autonomous and Adaptive Systems

Generative Learning

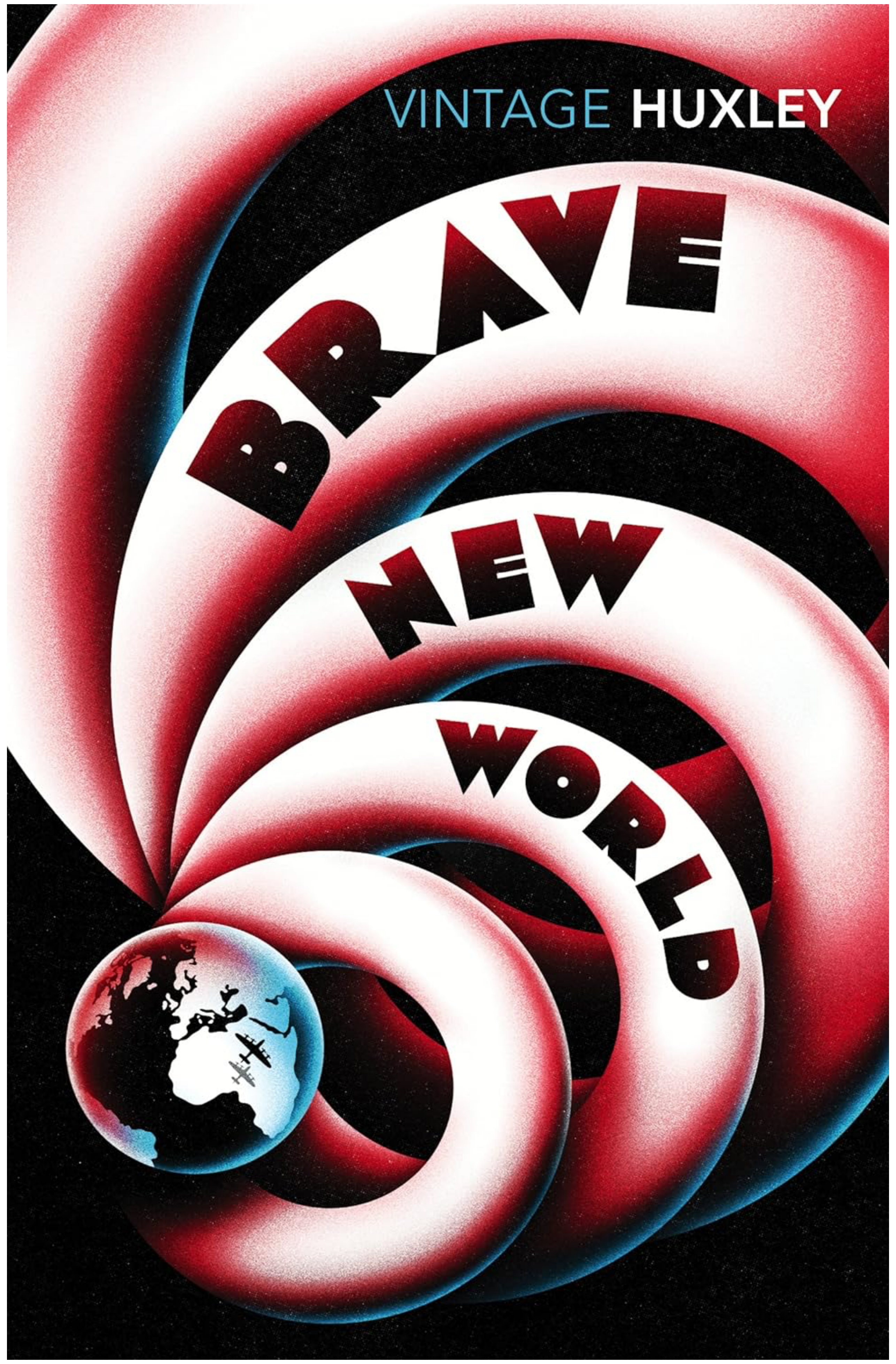
Mirco Musolesi

mircomusolesi@acm.org

Generative Learning

- In this lecture we will focus on machines that create and autonomously plan.
- In the recent years, many systems that exhbit the capacity of creating new artefacts and autonomously “invent” new solutions have been presented.
- And we are just at beginning…

A Brave New World



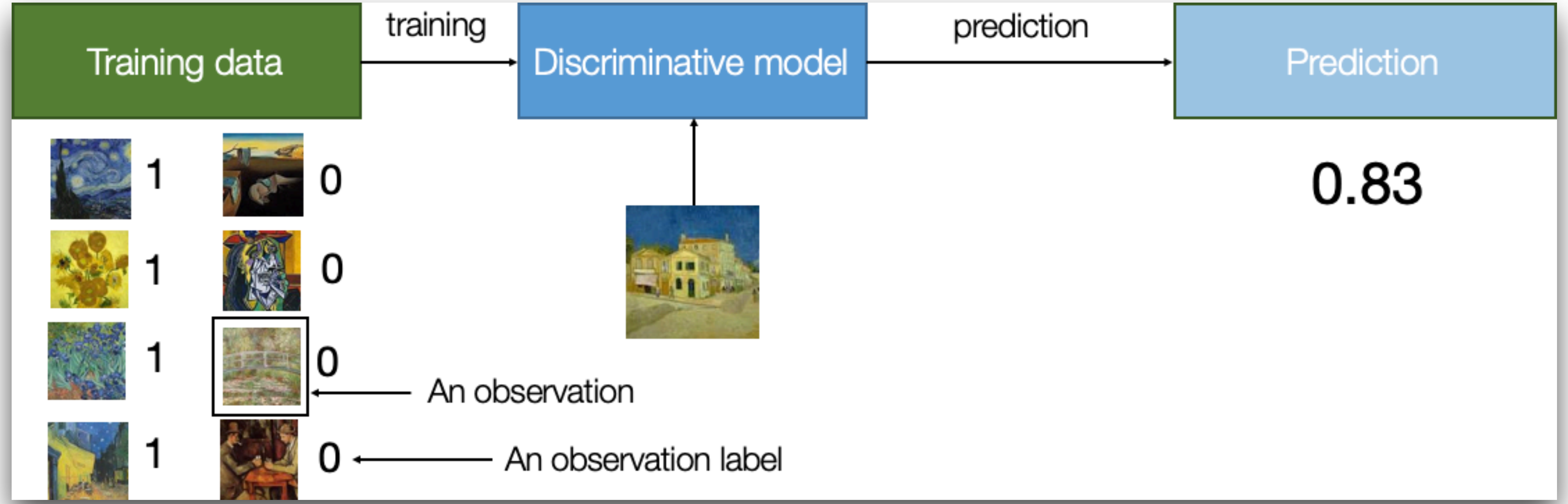
Generative Modeling

- A generative model describes how a dataset is generated for example through a probabilistic description. Through sampling of this model, we generate new data.

- The goal is to generate data are variations of the existing ones, but not “too far” from the original dataset.

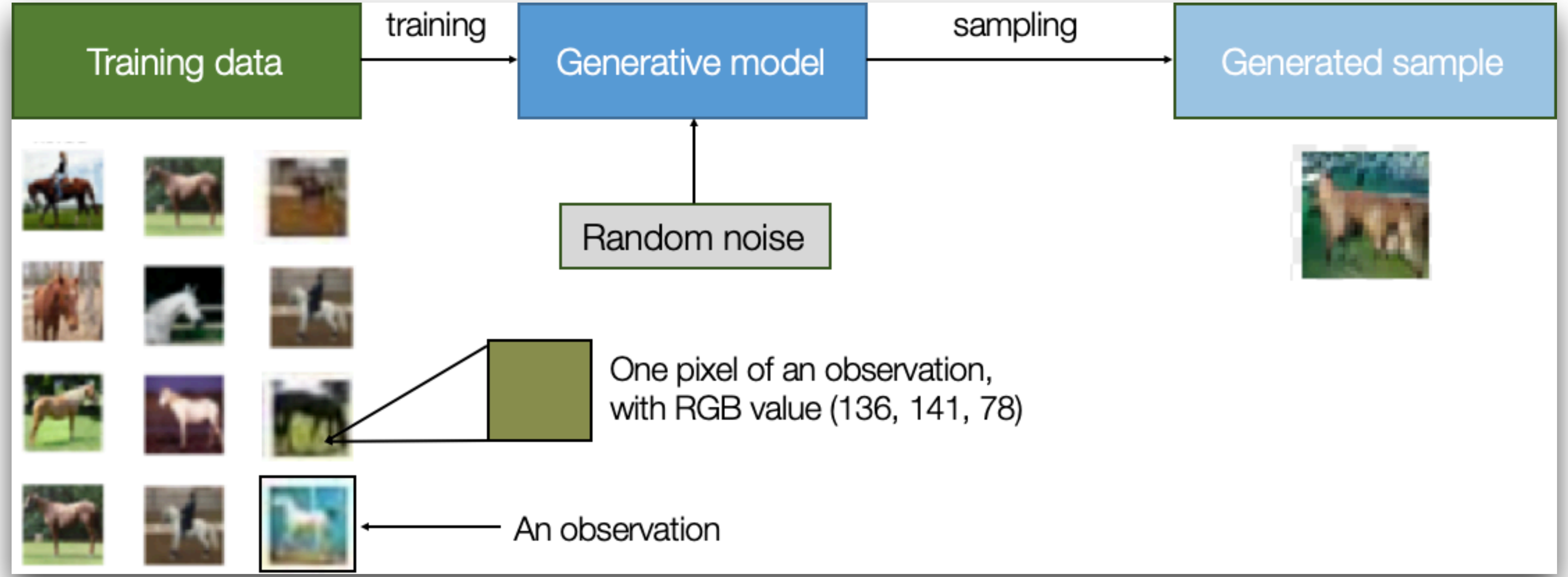
- A generative model is usually probabilistic rather than deterministic in nature.

Discriminative Model



Source: David Foster. Generative Deep Learning. O'Reilly. 2019.

Generative Model



Source: David Foster. Generative Deep Learning. O'Reilly. 2019.

Generative Modelling Framework

- Given a dataset $\mathbf{X}$, we assume that the observation has been generated according to some *unknown* distribution p_{data}.
- The goal is to create a generative model p_{model} that can be used to generate samples that look like they were drawn from p_{data}.
- We achieved our goal if the generated data are also suitably different from the observations in $\mathbf{X}$.
  - The model should not simply reproduce the things that have already been seen.

Variational Autoencoders

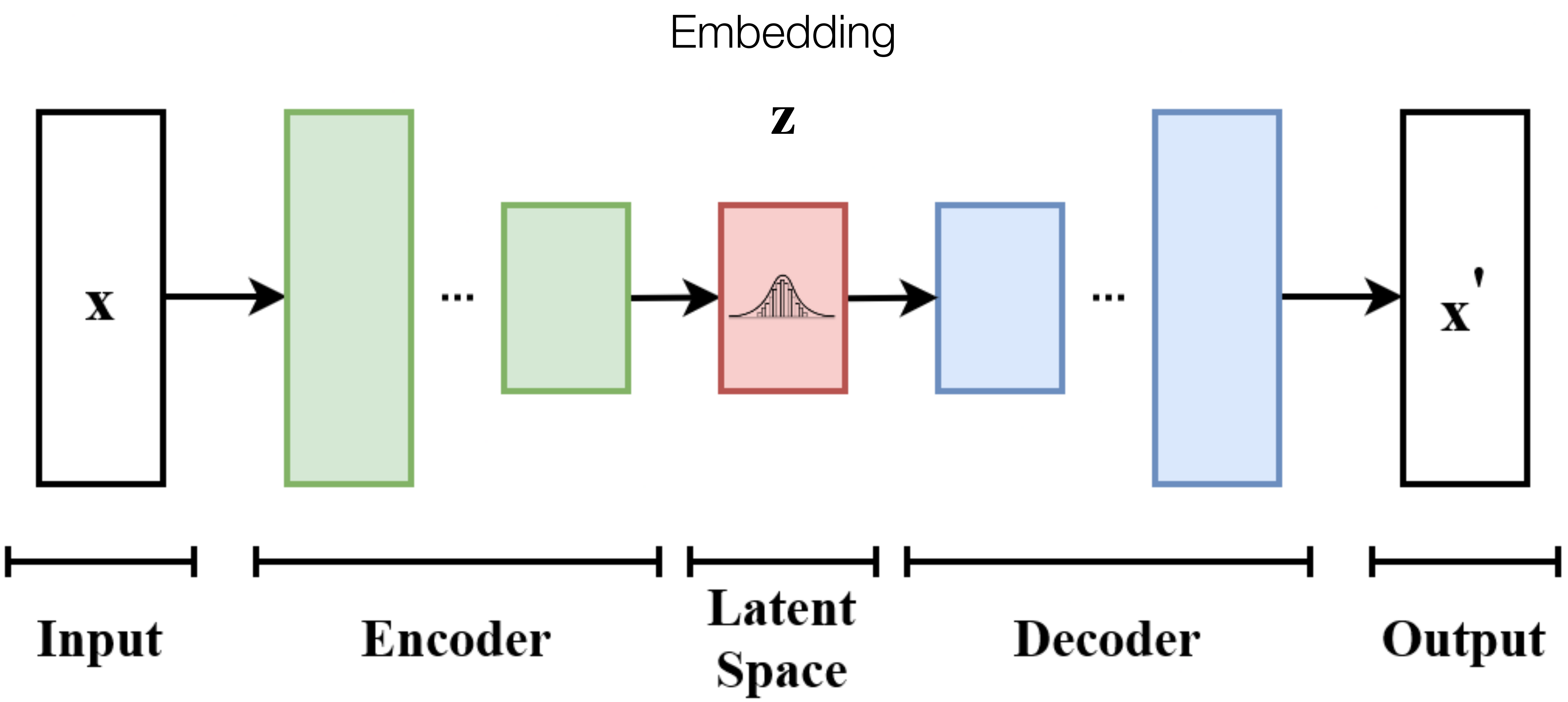
- An *autoencoder* is a neural network that is trained to perform the task of encoding and decoding an item, such that the output from this process is close to the original item as much as possible.

- An autoencoder is composed of two parts:

  - An *encoder network* that compresses high-dimensional input data such an image into a lower-dimensional embedding vector.

  - A *decoder network* that decompresses a given embedding vector back to the original.

Variational Autoencoders



Source: Wikimedia

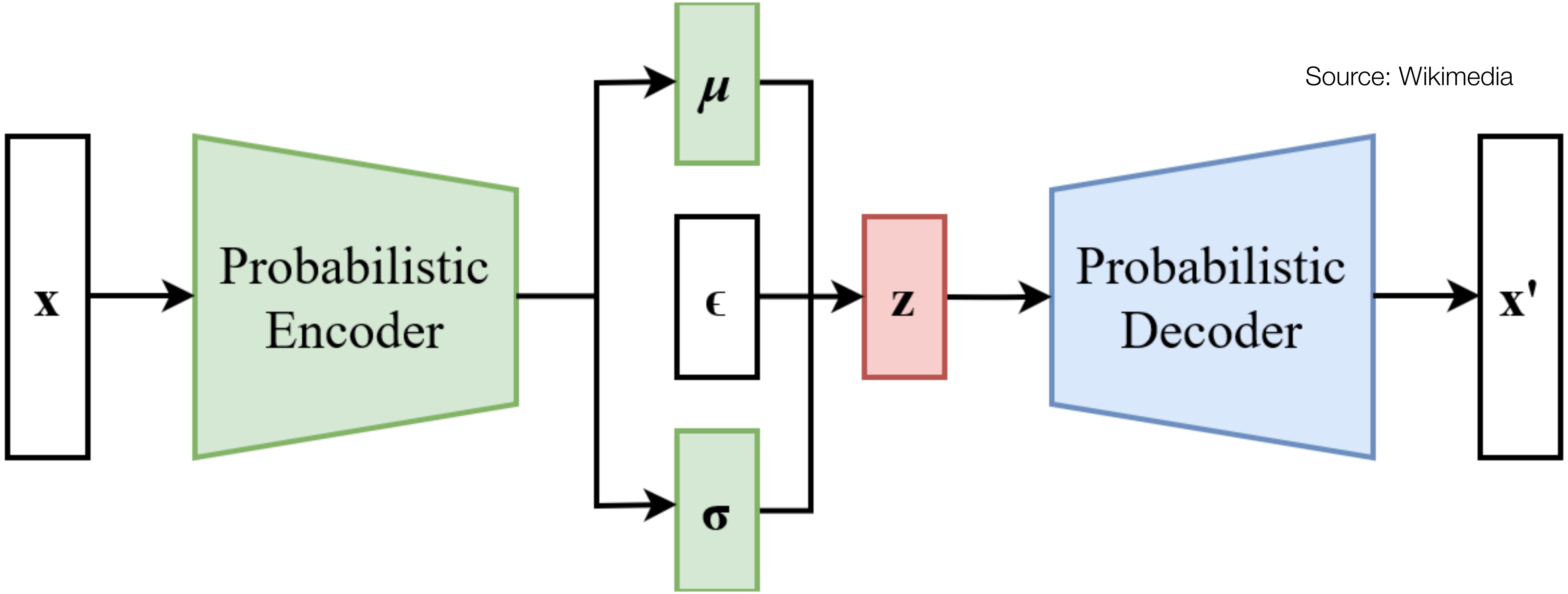
Variational Autoencoders

- The embedding $\mathbf{z}$ is a compression of the original input into a lower-dimensional latent space.

- By sampling the latent space, we can generate new outputs by passing the sample to the decoder, since the decoder has learned how to convert points into “realistic” outputs.

Variational Autoencoders

Source: Wikimedia

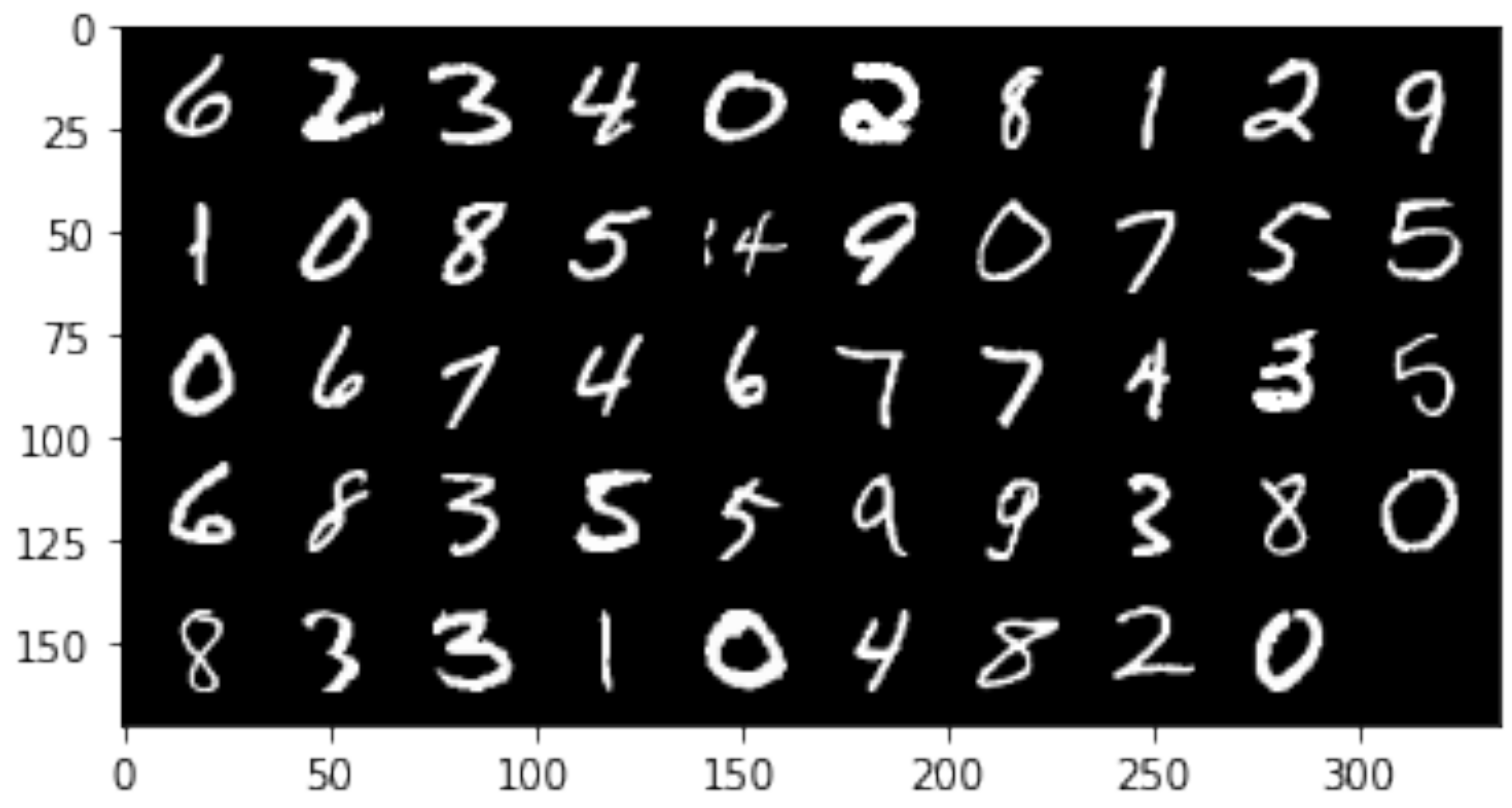


The encoder takes each input image and encode it into two vectors that defines a multivariate normal distribution in the latent space with mean μ and variance σ

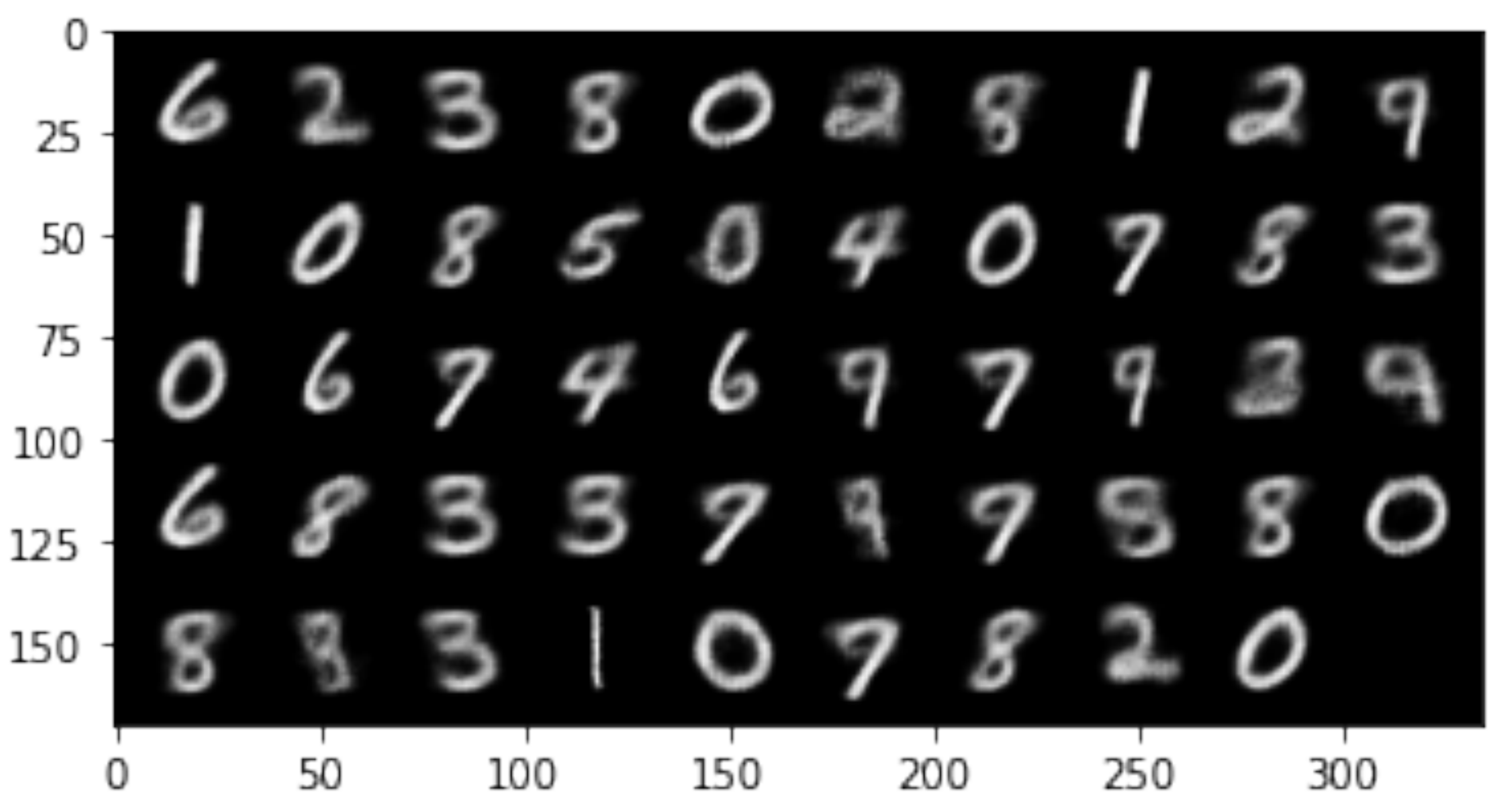
ϵ is sampled from a normal distribution $(\mathbf{0},\mathbf{I})$

We calculate $\mathbf{z}$ as follows $\mathbf{z} = \mu + \sigma\epsilon$

Variational Autoencoders



Original



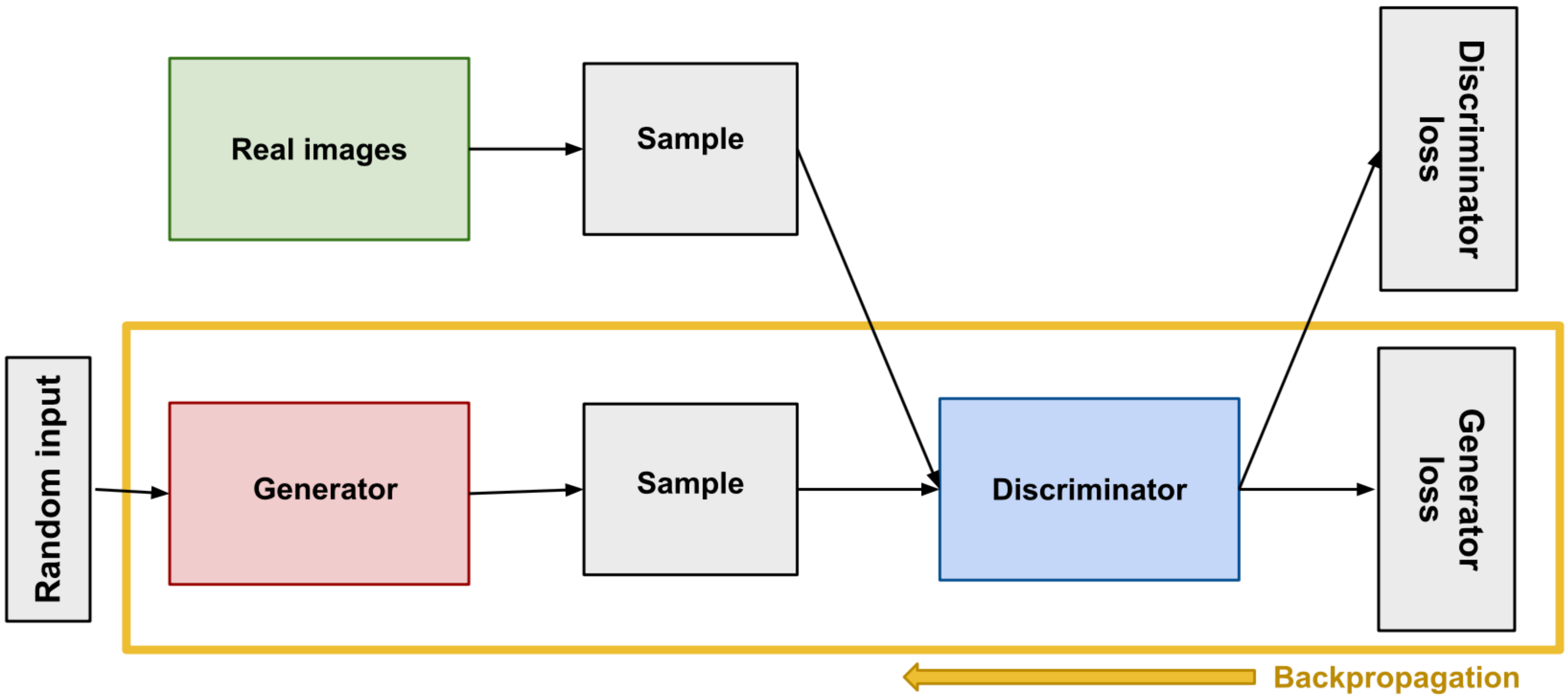
VAE Reconstruction

Colab notebook: https://colab.research.google.com/github/smartgeometry-ucl/dl4g/blob/master/variational_autoencoder.ipynb

Generative Adversarial Networks (GANs)

- A Generative Adversarial Network (GAN) is a class of machine learning techniques in which two neural networks play against each other.
- The *generative network* generates candidates, while the *discriminative network* evaluate them.
- The generative network tries to create new samples that look similar from the true data (an original distribution, for example portraits). The goal of the discriminator is to identify if the data given in input are from the original distribution or not.
- The generative network's training objective is to increase the error rate of the discriminative network (i.e., to fool the discriminative network)
- Indeed, the discriminative network's training objective is to minimise its error rate in discriminating the input.
- This is used for images, videogame generation, scientific images, etc.

Generative Adversarial Networks (GANs)



Source: https://developers.google.com/machine-learning/gan/generator

Generative Adversarial Nets

Ian J. Goodfellow,[*] Jean Pouget-Abadie,[†] Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair,[‡] Aaron Courville, Yoshua Bengio[§]
Département d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC H3C 3J7

Abstract

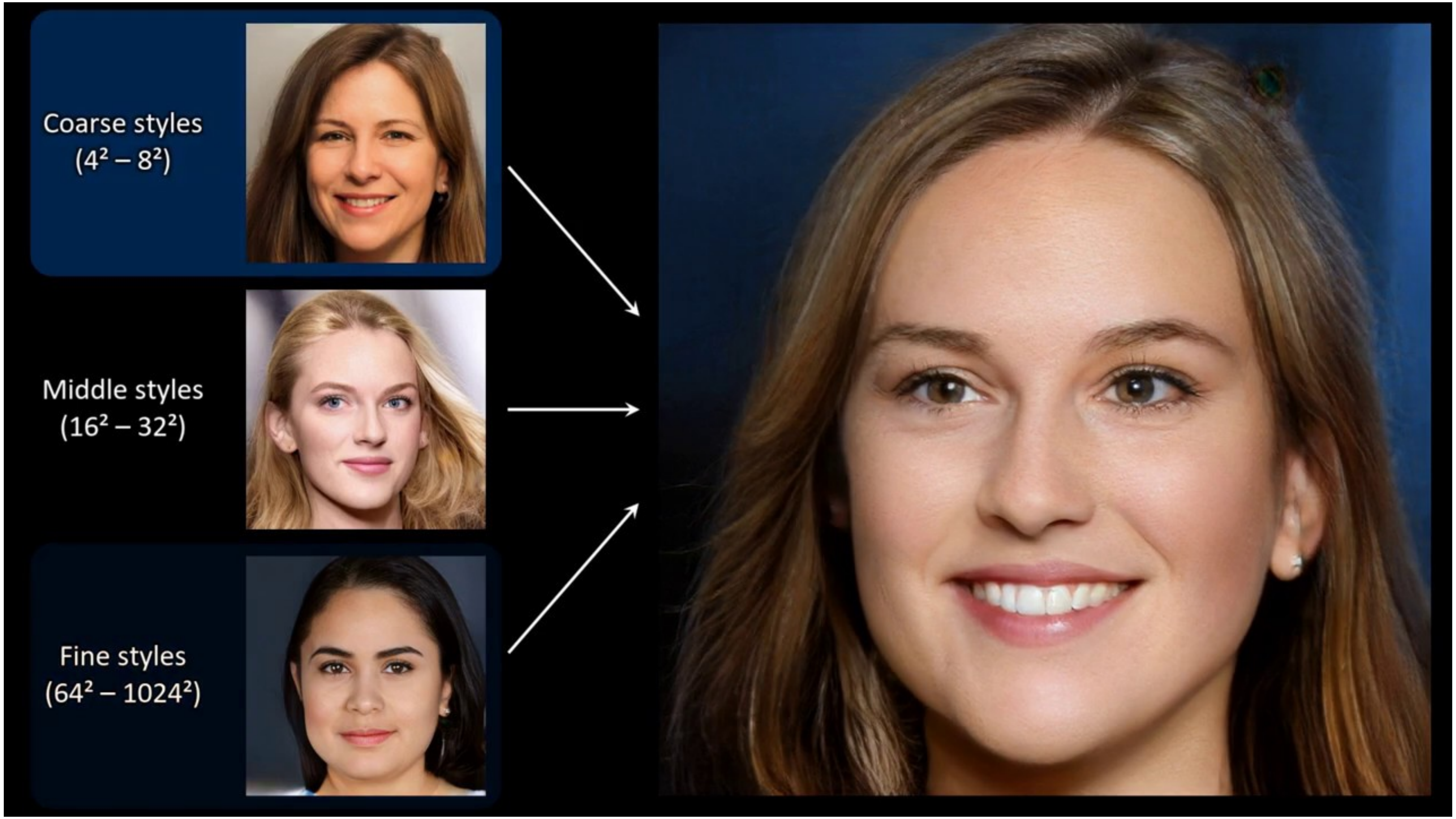
We propose a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D, a unique solution exists, with G recovering the training data distribution and D equal to $\frac{1}{2}$ everywhere. In the case where G and D are defined by multilayer perceptrons, the entire system can be trained with backpropagation. There is no need for any Markov chains or unrolled approximate inference networks during either training or generation of samples. Experiments demonstrate the potential of the framework through qualitative and quantitative evaluation of the generated samples.

Figure 2: Visualization of samples from the model. Rightmost column shows the nearest training example of the neighboring sample, in order to demonstrate that the model has not memorized the training set. Samples are fair random draws, not cherry-picked. Unlike most other visualizations of deep generative models, these images show actual samples from the model distributions, not conditional means given samples of hidden units. Moreover, these samples are uncorrelated because the sampling process does not depend on Markov chain mixing. a) MNIST b) TFD c) CIFAR-10 (fully connected model) d) CIFAR-10 (convolutional discriminator and “deconvolutional” generator)

Generation of Images using Generative Adversarial Networks (GANs)

Source: David Foster. Generative Deep Learning. O'Reilly. 2019.

StyleGAN


Coarse styles
(4² – 8²)
Middle styles
(16² – 32²)
Fine styles
(64² – 1024²)

A Style-Based Generator Architecture for Generative Adversarial Networks

Tero Karras
NVIDIA
tkarras@nvidia.com

Samuli Laine
NVIDIA
slaine@nvidia.com

Timo Aila
NVIDIA
taila@nvidia.com

Abstract

We propose an alternative generator architecture for generative adversarial networks, borrowing from style transfer literature. The new architecture leads to an automatically learned, unsupervised separation of high-level attributes (e.g., pose and identity when trained on human faces) and stochastic variation in the generated images (e.g., freckles, hair), and it enables intuitive, scale-specific control of the synthesis. The new generator improves the state-of-the-art in terms of traditional distribution quality metrics, leads to demonstrably better interpolation properties, and also better disentangles the latent factors of variation. To quantify interpolation quality and disentanglement, we propose two new, automated methods that are applicable to any generator architecture. Finally, we introduce a new, highly varied and high-quality dataset of human faces.

(e.g., pose, identity) from stochastic variation (e.g., freckles, hair) in the generated images, and enables intuitive scale-specific mixing and interpolation operations. We do not modify the discriminator or the loss function in any way, and our work is thus orthogonal to the ongoing discussion about GAN loss functions, regularization, and hyperparameters [24, 45, 5, 40, 44, 36].

Our generator embeds the input latent code into an intermediate latent space, which has a profound effect on how the factors of variation are represented in the network. The input latent space must follow the probability density of the training data, and we argue that this leads to some degree of unavoidable entanglement. Our intermediate latent space is free from that restriction and is therefore allowed to be disentangled. As previous methods for estimating the degree of latent space disentanglement are not directly applicable in our case, we propose two new automated metrics — perceptual path length and linear separability — for quantifying these aspects of the generator. Using these metrics, we show that compared to a traditional generator architecture,

Transformers

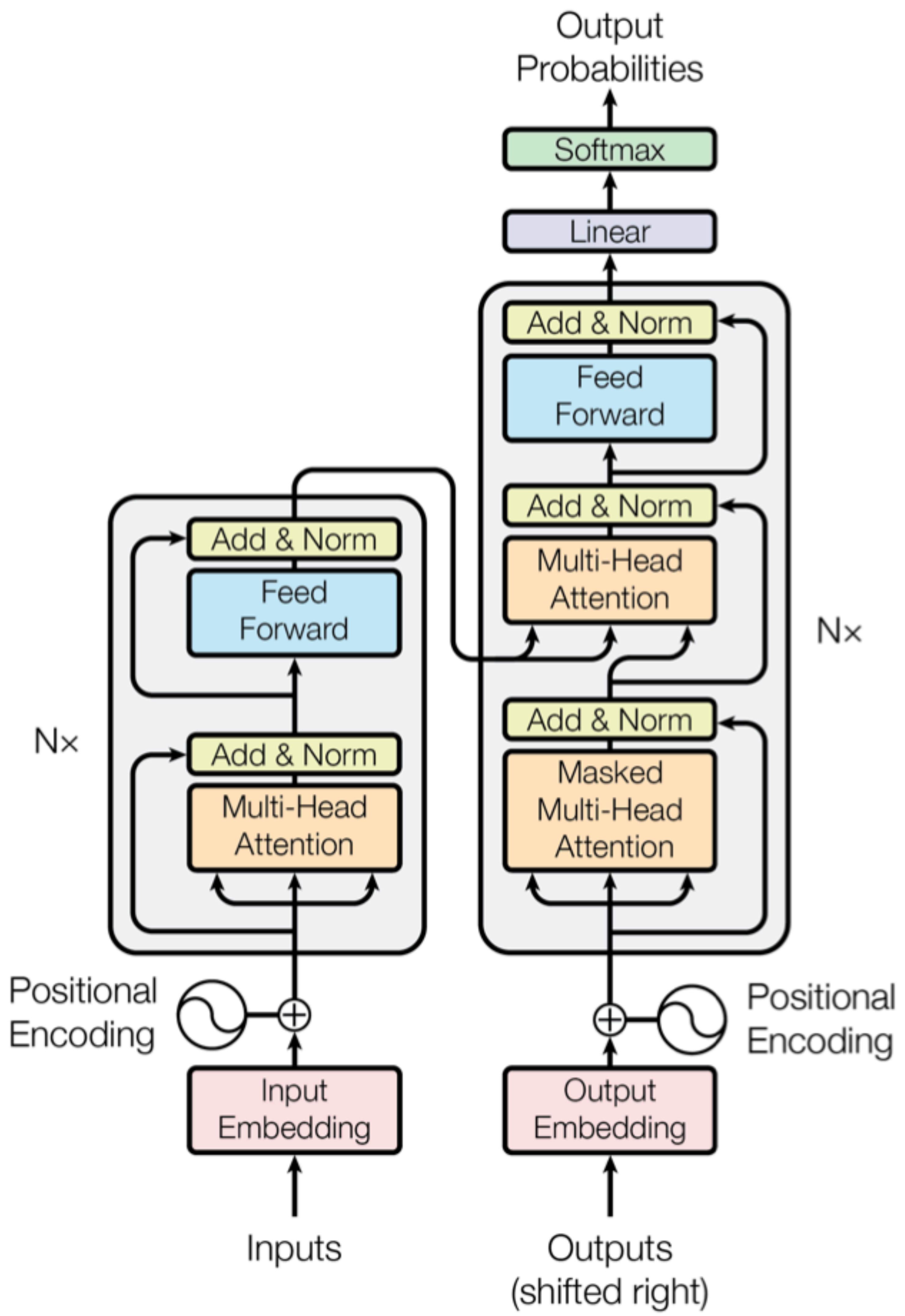
Attention Is All You Need

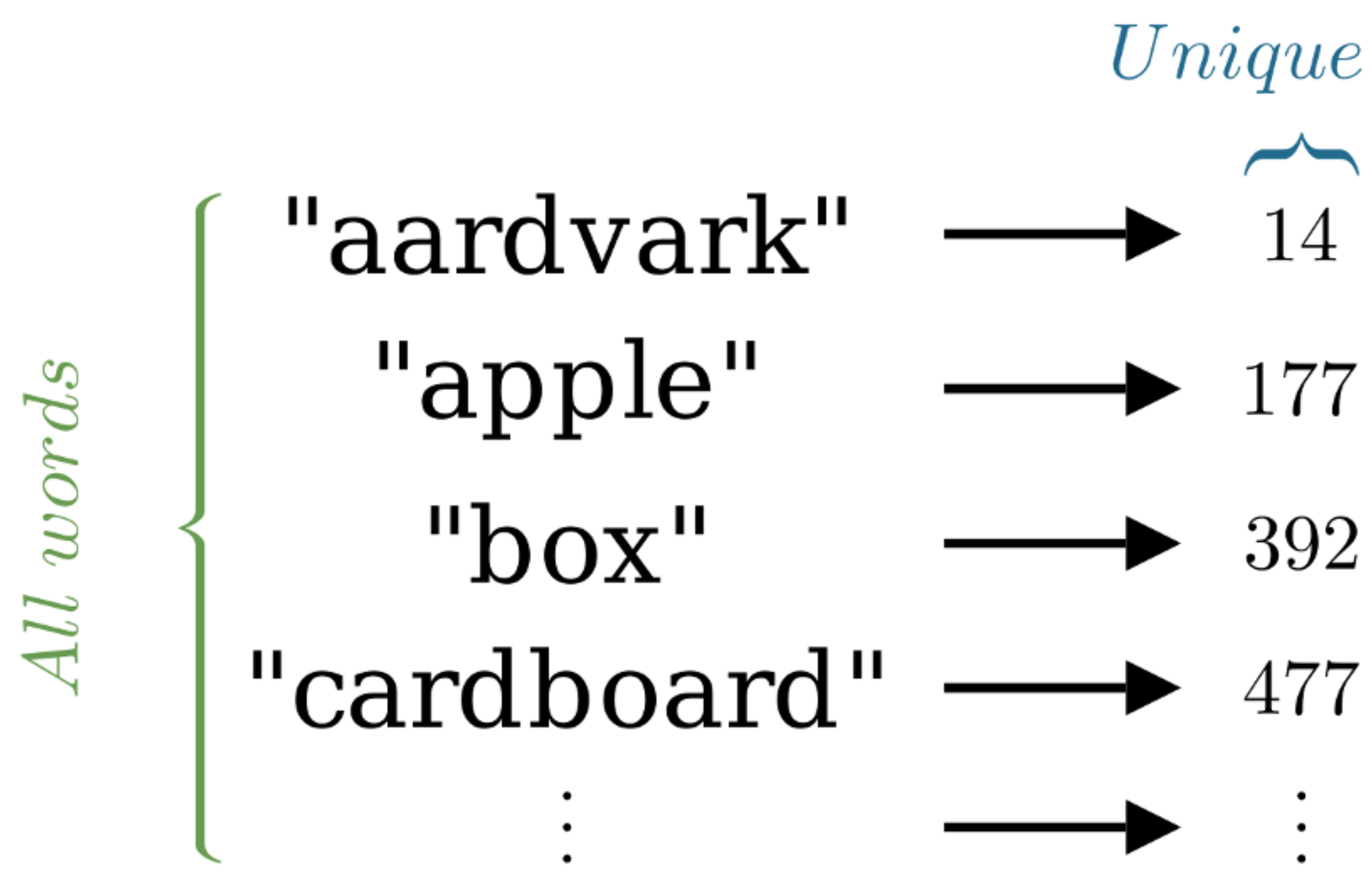
Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

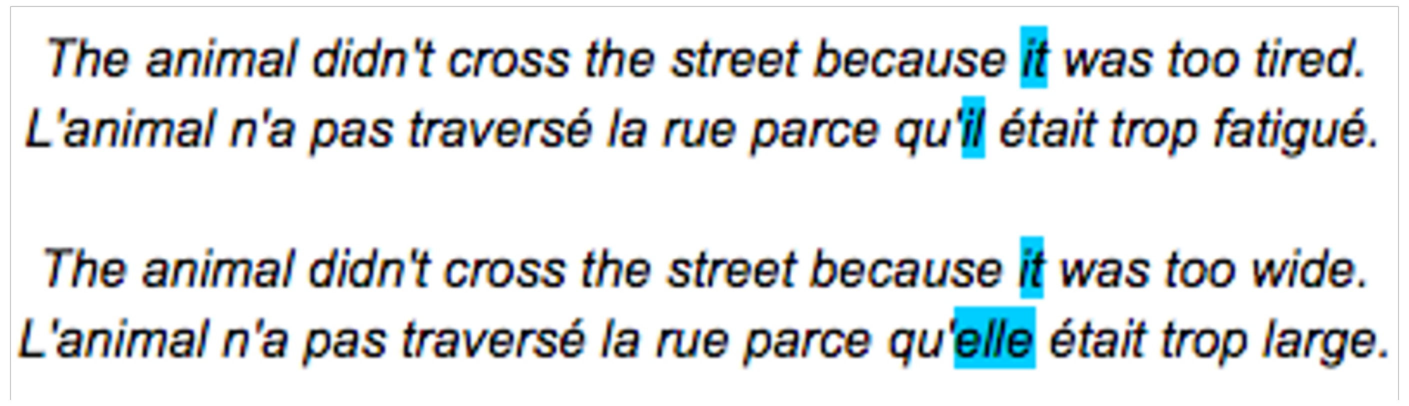
Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

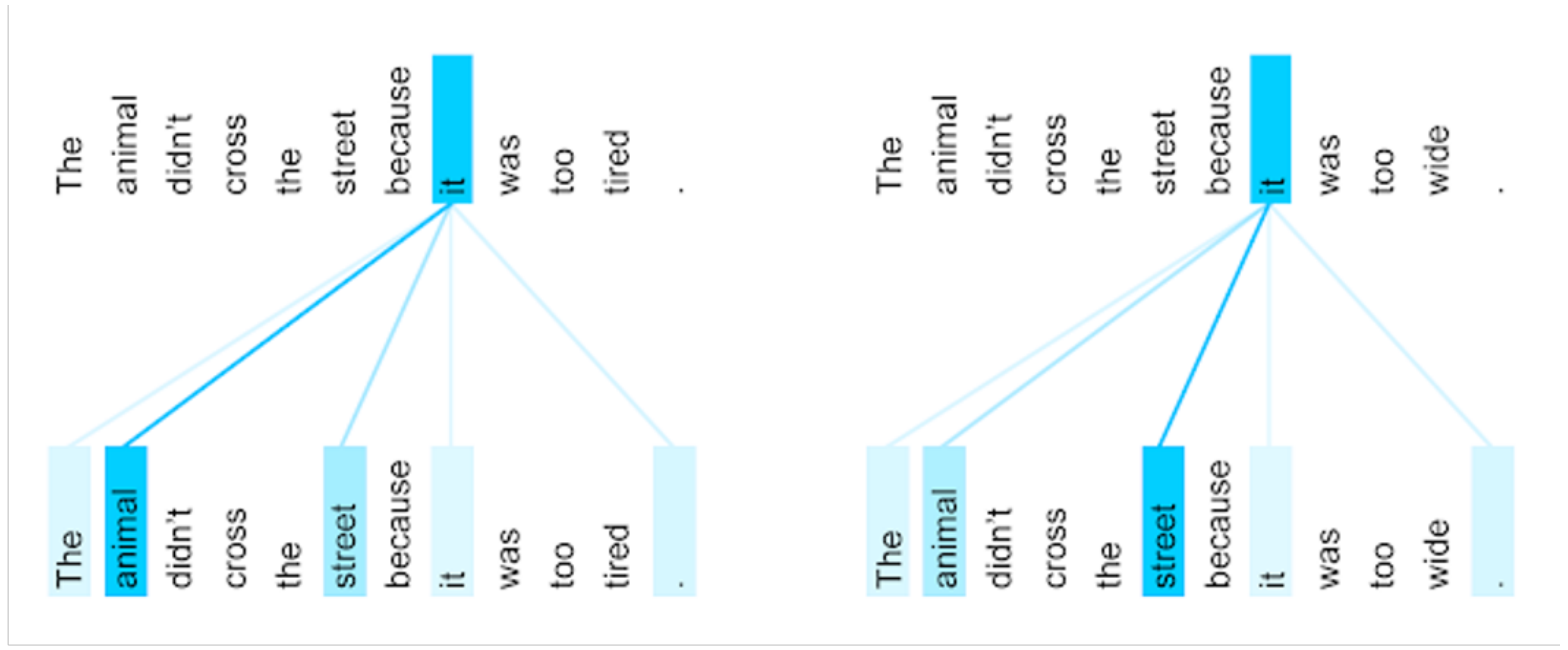
Transformers






The animal didn't cross the street because it was too tired.
L'animal n'a pas traversé la rue parce qu'il était trop fatigué.

The animal didn't cross the street because it was too wide.
L'animal n'a pas traversé la rue parce qu'elle était trop large.



Generative Agents

Generative Agents: Interactive Simulacra of Human Behavior

Joon Sung Park
Stanford University
Stanford, USA
joonspk@stanford.edu

Joseph C. O'Brien
Stanford University
Stanford, USA
jobrien3@stanford.edu

Carrie J. Cai
Google Research
Mountain View, CA, USA
cjcai@google.com

Meredith Ringel Morris
Google DeepMind
Seattle, WA, USA
merrie@google.com

Percy Liang
Stanford University
Stanford, USA
pliang@cs.stanford.edu

Michael S. Bernstein
Stanford University
Stanford, USA
msb@cs.stanford.edu

Figure 1: Generative agents are believable simulacra of human behavior for interactive applications. In this work, we demonstrate generative agents by populating a sandbox environment, reminiscent of The Sims, with twenty-five agents. Users can observe and intervene as agents plan their days, share news, form relationships, and coordinate group activities.

ACM UIST 2023

LLM Agents

- Cognitive Core: The LLM serves as the core decision-making units, performing zero-shot task decomposition and semantic world-modeling.
- Planning Architectures examples:
  - ReAct (Reason + Act)
  - Chain-of-Thought (CoT) to generate intermediate execution steps.
- Memory Systems:
  - Dual-tier integration:
  - Short-term: In-context window management.
  - Long-term: Vector DBs for semantic retrieval.

LLM Agents

- Cognitive Core: The LLM serves as the core decision-making units, performing zero-shot task decomposition and semantic world-modeling.
- Planning Architectures examples:
  - ReAct (Reason + Act)
  - Chain-of-Thought (CoT) to generate intermediate execution steps.
- Memory Systems:
  - Dual-tier integration:
  - Short-term: In-context window management.
  - Long-term: Vector DBs for semantic retrieval.

Reinforcement Learning from Human Feedback

Fine-Tuning Language Models from Human Preferences

Daniel M. Ziegler* **Nisan Stiennon*** **Jeffrey Wu** **Tom B. Brown**
Alec Radford **Dario Amodei** **Paul Christiano** **Geoffrey Irving**
OpenAI
{dmz,nisan,jeffwu,tom,alec,damodei,paul,irving}@openai.com

Abstract

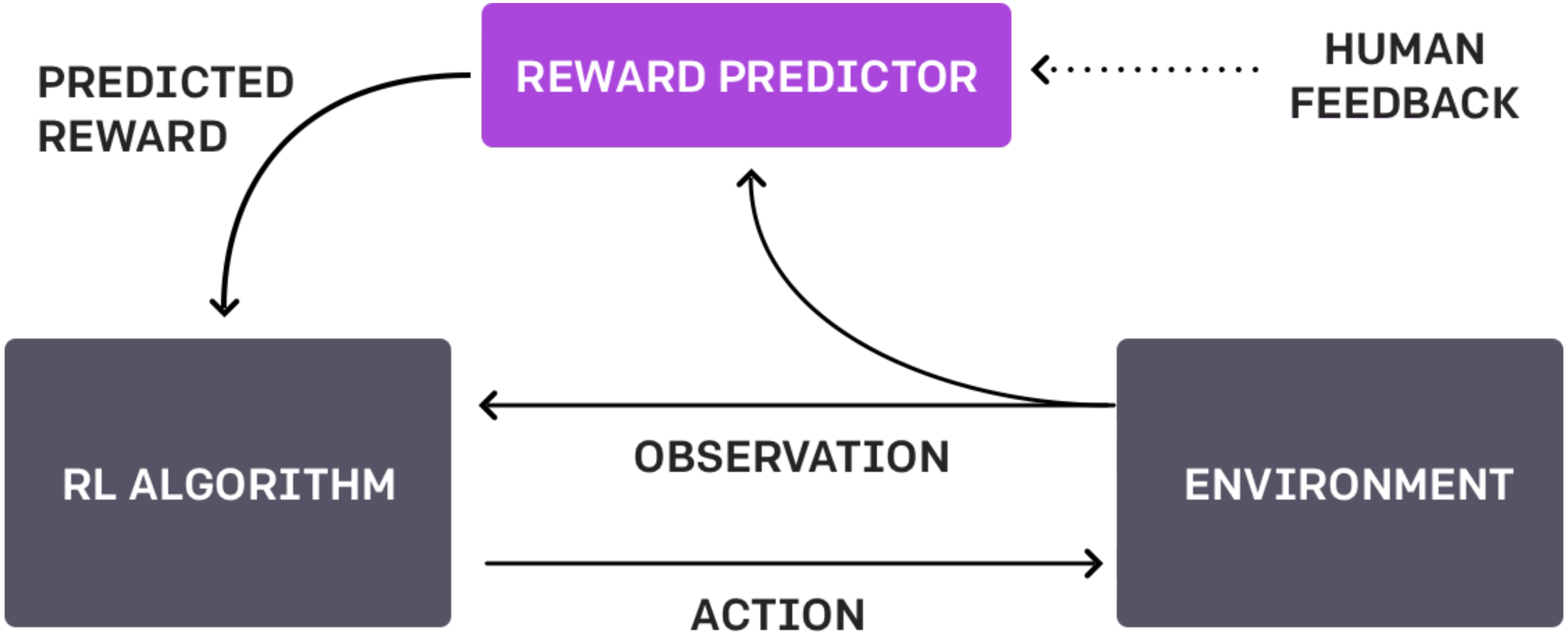
Reward learning enables the application of reinforcement learning (RL) to tasks where reward is defined by human judgment, building a model of reward by asking humans questions. Most work on reward learning has used simulated environments, but complex information about values is often expressed in natural language, and we believe reward learning for language is a key to making RL practical and safe for real-world tasks. In this paper, we build on advances in generative pretraining of language models to apply reward learning to four natural language tasks: continuing text with positive sentiment or physically descriptive language, and summarization tasks on the TL;DR and CNN/Daily Mail datasets. For stylistic continuation we achieve good results with only 5,000 comparisons evaluated by humans. For summarization, models trained with 60,000 comparisons copy whole sentences from the input but skip irrelevant preamble; this leads to reasonable ROUGE scores and very good performance according to our human labelers, but may be exploiting the fact that labelers rely on simple heuristics.

plex goals to AI agents are likely to both involve and require natural language, which is a rich medium for expressing value-laden concepts. Natural language is particularly important when an agent must communicate back to a human to help provide a more accurate supervisory signal (Irving et al., 2018; Christiano et al., 2018; Leike et al., 2018).

Natural language processing has seen substantial recent advances. One successful method has been to pretrain a large generative language model on a corpus of unsupervised data, then fine-tune the model for supervised NLP tasks (Dai and Le, 2015; Peters et al., 2018; Radford et al., 2018; Khandelwal et al., 2019). This method often substantially outperforms training on the supervised datasets from scratch, and a single pretrained language model often can be fine-tuned for state of the art performance on many different supervised datasets (Howard and Ruder, 2018). In some cases, fine-tuning is not required: Radford et al. (2019) find that generatively trained models show reasonable performance on NLP tasks with no additional training (zero-shot).

There is a long literature applying reinforcement learning to natural language tasks. Much of this work uses algorithmically defined reward functions such as BLEU for translation (Ranzato et al., 2015; Wu et al., 2016), ROUGE for summarization (Ranzato et al., 2015; Paulus et al., 2017; Wu and Hu, 2018; Gao et al., 2019b), music theory-based rewards

Reinforcement Learning from Human Feedback (RLHF)



Source: OpenAI

Direct Preference Optimization (DPO)

Fine-Tuning Language Models from Human Preferences

Daniel M. Ziegler* Nisan Stiennon* Jeffrey Wu Tom B. Brown
Alec Radford Dario Amodei Paul Christiano Geoffrey Irving
OpenAI
{dmz,nisan,jeffwu,tom,alec,damodei,paul,irving}@openai.com

Abstract

Reward learning enables the application of reinforcement learning (RL) to tasks where reward is defined by human judgment, building a model of reward by asking humans questions. Most work on reward learning has used simulated environments, but complex information about values is often expressed in natural language, and we believe reward learning for language is a key to making RL practical and safe for real-world tasks. In this paper, we build on advances in generative pretraining of language models to apply reward learning to four natural language tasks: continuing text with positive sentiment or physically descriptive language, and summarization tasks on the TL;DR and CNN/Daily Mail datasets. For stylistic continuation we achieve good results with only 5,000 comparisons evaluated by humans. For summarization, models trained with 60,000 comparisons copy whole sentences from the input but skip irrelevant preamble; this leads to reasonable ROUGE scores and very good performance according to our human labelers, but may be exploiting the fact that labelers rely on simple heuristics.

plex goals to AI agents are likely to both involve and require natural language, which is a rich medium for expressing value-laden concepts. Natural language is particularly important when an agent must communicate back to a human to help provide a more accurate supervisory signal (Irving et al., 2018; Christiano et al., 2018; Leike et al., 2018).

Natural language processing has seen substantial recent advances. One successful method has been to pretrain a large generative language model on a corpus of unsupervised data, then fine-tune the model for supervised NLP tasks (Dai and Le, 2015; Peters et al., 2018; Radford et al., 2018; Khandelwal et al., 2019). This method often substantially outperforms training on the supervised datasets from scratch, and a single pretrained language model often can be fine-tuned for state of the art performance on many different supervised datasets (Howard and Ruder, 2018). In some cases, fine-tuning is not required: Radford et al. (2019) find that generatively trained models show reasonable performance on NLP tasks with no additional training (zero-shot).

There is a long literature applying reinforcement learning to natural language tasks. Much of this work uses algorithmically defined reward functions such as BLEU for translation (Ranzato et al., 2015; Wu et al., 2016), ROUGE for summarization (Ranzato et al., 2015; Paulus et al., 2017; Wu and Hu, 2018; Gao et al., 2019b), music theory-based rewards

What is Creativity?

Source: Wikimedia

"Creativity can be defined as the ability to generate novel, and valuable, ideas. Valuable, here, has many meanings: interesting, useful, beautiful, simple, richly complex, and so on. Ideas covers many meaning too: not only ideas as such (concepts, theories, interpretations, stories), but also artifacts such as graphic images, sculptures, houses and jet engines. Computer models have been designed to generate ideas in all these areas and more."

Margaret A. Boden

Ada Lovelace's Objection

Source: Computer History Museum

"The Analytical Engine has no pretensions whatever to originate anything. It can do whatever we know how to order it to perform; but it has no power of anticipating any analytical relations or truths."

Ada Lovelace

Turing's Response

VOL. LIX. NO. 236.] [October, 1950

MIND
A QUARTERLY REVIEW
OF
PSYCHOLOGY AND PHILOSOPHY

I.—COMPUTING MACHINERY AND INTELLIGENCE

BY A. M. TURING

1. *The Imitation Game.*

I PROPOSE to consider the question, 'Can machines think?' This should begin with definitions of the meaning of the terms 'machine' and 'think'. The definitions might be framed so as to reflect so far as possible the normal use of the words, but this attitude is dangerous. If the meaning of the words 'machine' and 'think' are to be found by examining how they are commonly used it is difficult to escape the conclusion that the meaning and the answer to the question, 'Can machines think?' is to be sought in a statistical survey such as a Gallup poll. But this is absurd. Instead of attempting such a definition I shall replace the question by another, which is closely related to it and is expressed in relatively unambiguous words.

The new form of the problem can be described in terms of a game which we call the 'imitation game'. It is played with three people, a man (A), a woman (B), and an interrogator (C) who may be of either sex. The interrogator stays in a room apart from the other two. The object of the game for the interrogator is to determine which of the other two is the man and which is the woman. He knows them by labels X and Y, and at the end of the game he says either 'X is A and Y is B' or 'X is B and Y is A'. The interrogator is allowed to put questions to A and B thus:

C: Will X please tell me the length of his or her hair?

433

- Ada Lovelace's objection can be seen as the assertion that computers cannot surprise us.

- Alan Turing in his Mind paper argues that actually computers are still able to surprise us. He also underlines the fact that Ada Lovelace lived in a period where neurological phenomena were not known.

Can an (artificial) agent be creative?

References

- Christopher Bishop and Hugh Bishop. Deep Learning. Springer. 2024
- Francois Chollet. Deep Learning with Python. Third Edition. Manning. 2025.
- David Foster. Generative Deep Learning. Second Edition. O'Reilly. 2023.
- Arthur I. Miller. The Artist in the Machine. The World of AI-Powered Creativity. MIT Press. 2019.