

Generative Networks

ACTL3143 & ACTL5111 Deep Learning for Actuaries
Patrick Laub

Lecture Outline

- **Generative Adversarial Networks**
- Conditional GANs
- Image-to-image translation
- Problems with GANs
- Language Models
- Sampling strategy
- Transformers

GAN faces



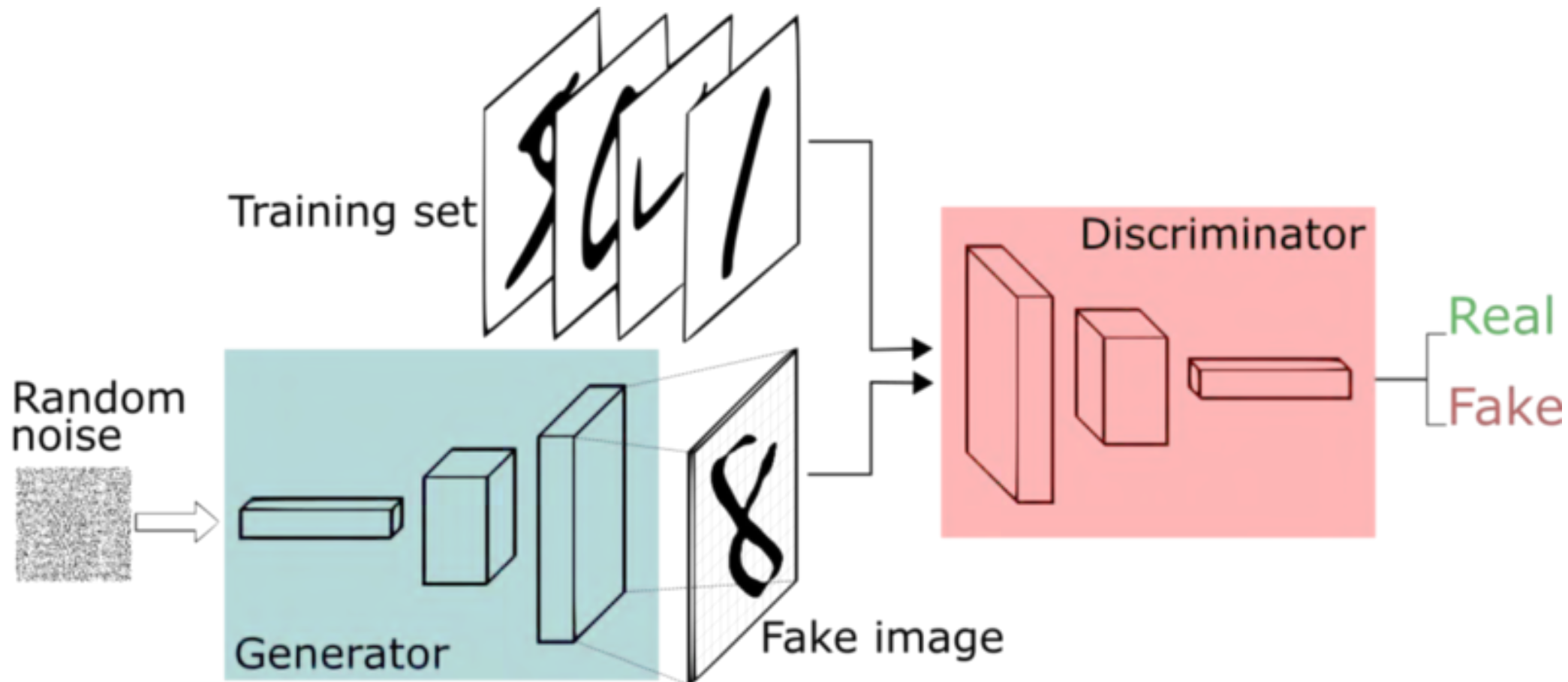
Try out <https://www.whichfaceisreal.com>
Source: <https://thispersondoesnotexist.com>.

Example StyleGAN2-ADA outputs



Source: Jeff Heaton (2021), [Training a GAN from your Own Images: StyleGAN2](#).

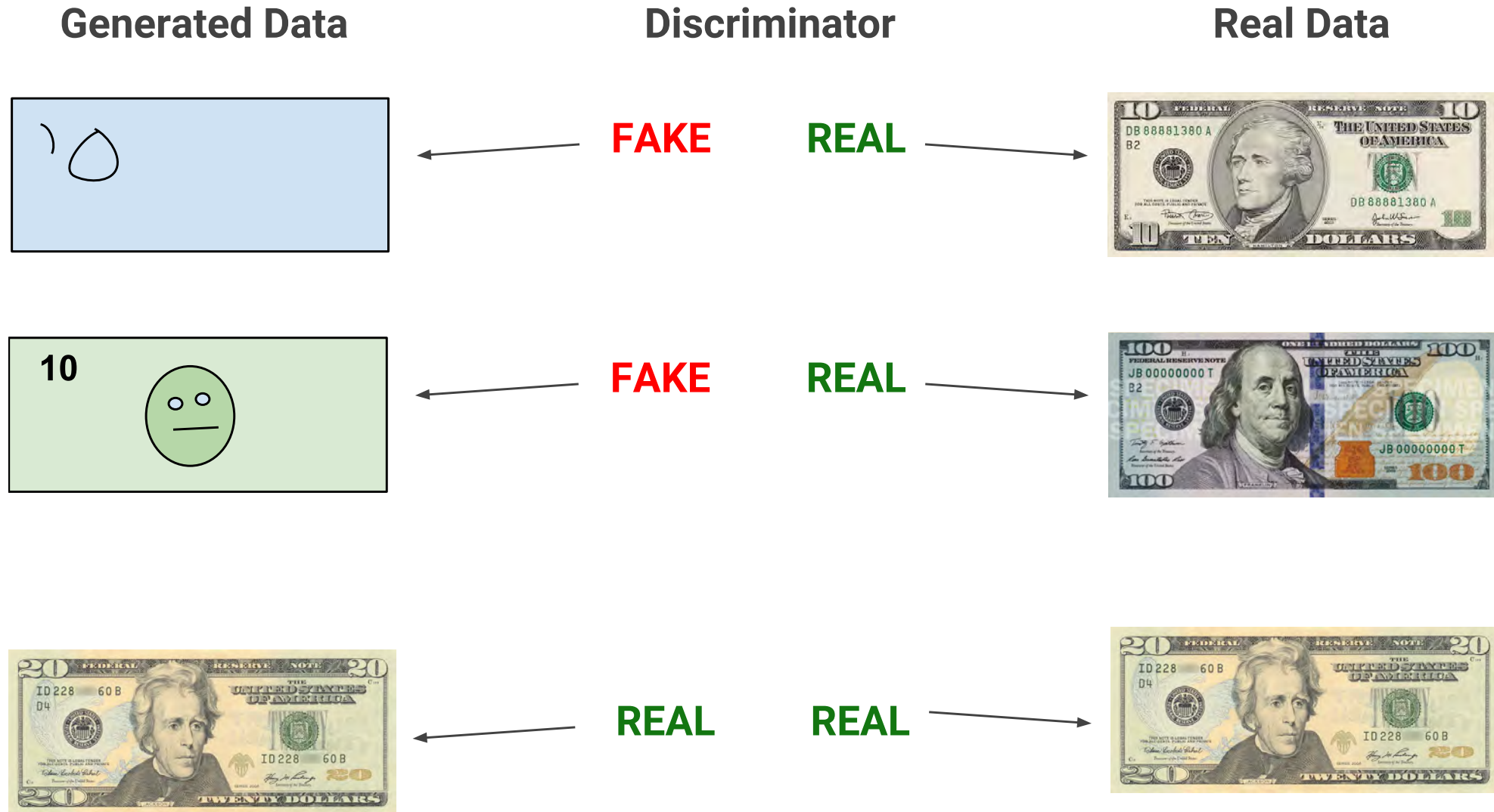
GAN structure



A schematic of a generative adversarial network.

Source: Thales Silva (2018), [An intuitive introduction to Generative Adversarial Networks \(GANs\)](#), freeCodeCamp.

GAN intuition



Source: Google Developers, [Overview of GAN Structure](#), Google Machine Learning Education.

Intuition about GANs

- A forger creates a fake Picasso painting to sell to an art dealer.
- The art dealer assesses the painting.

How they best each other:

- The art dealer is given both authentic paintings and fake paintings to look at. Later on, the validity of his assessment is evaluated and he trains to become better at detecting fakes. Over time, he becomes increasingly expert at authenticating Picasso's artwork.
- The forger receives an assessment from the art dealer every time he gives him a fake. He knows he has to perfect his craft if the art dealer can detect his fake. He becomes increasingly adept at imitating Picasso's style.

Generative adversarial networks

- A GAN is made up of two parts:
 - Generator network: the forger. Takes a random point in the latent space, and decodes it into a synthetic data/image.
 - Discriminator network (or adversary): the expert. Takes a data/image and decides whether it exists in the original data set (the training set) or was created by the generator network.

GAN – Schematic process

First step: Training *discriminator*:

- Draw random points in the latent space (random noise).
- Use *generator* to generate data from this random noise.
- Mix generated data with real data and input them into the *discriminator*. The training targets are the correct labels of *real data* or *fake data*. Use *discriminator* to give feedback on the mixed data whether they are real or synthetic. Train *discriminator* to minimize the loss function which is the difference between the *discriminator's* feedback and the correct labels.

GAN – Schematic process II

Second step: Training *generator*:

- Draw random points in the latent space and generate data with *generator*.
- Use *discriminator* to give feedback on the generated data. What the generator tries to achieve is to fool the *discriminator* into thinking all generated data are real data. Train *generator* to minimize the loss function which is the difference between the discriminator's feedback and the desired feedback: "All data are real data" (which is not true).

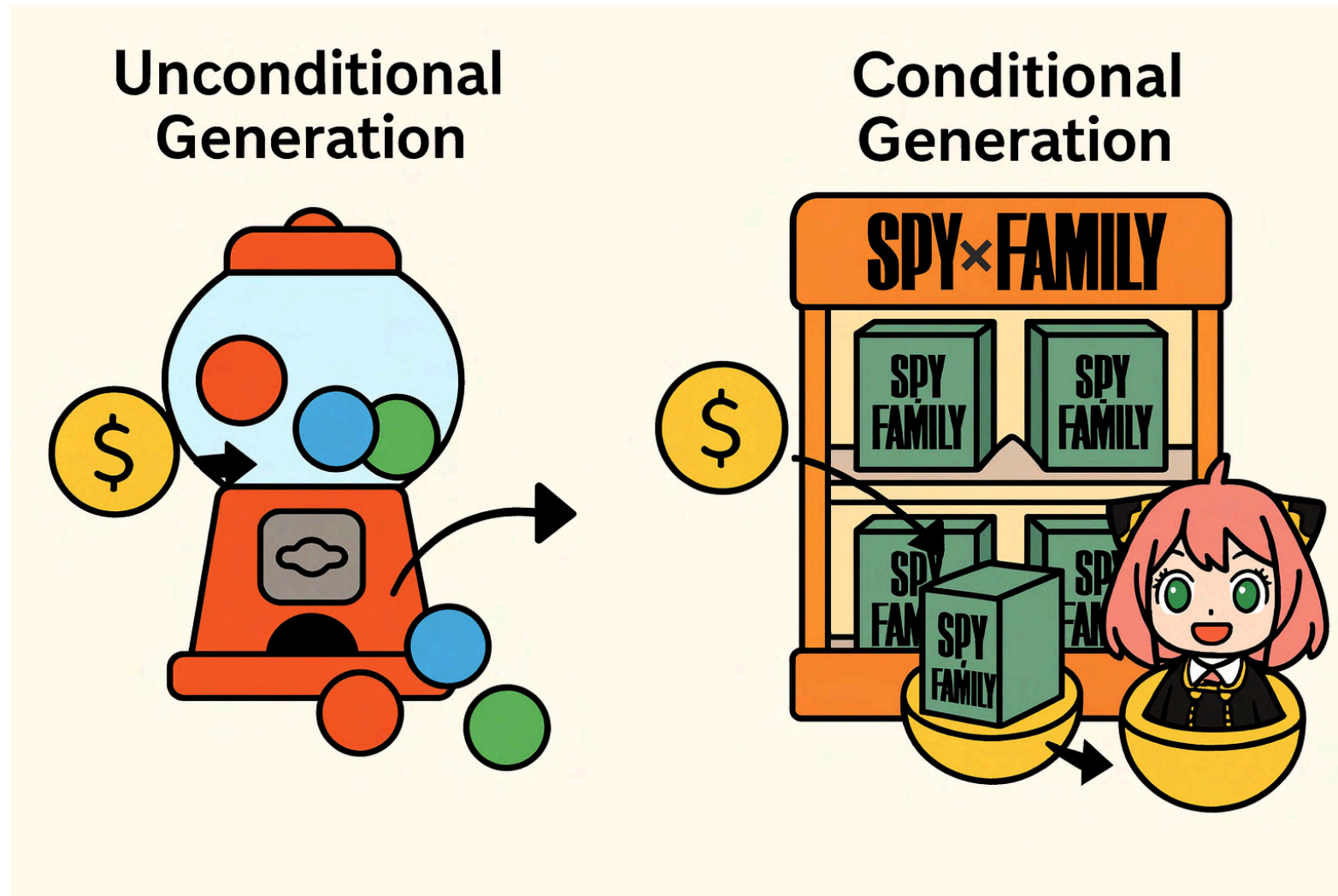
GAN - Schematic process III

- When training, the discriminator may end up dominating the generator because the loss function for training the discriminator tends to zero faster. In that case, try reducing the learning rate and increasing the dropout rate of the discriminator.
- There are a few tricks for implementing GANs such as introducing stochasticity by adding random noise to the labels for the discriminator, using stride instead of pooling in the discriminator, using kernel size that is divisible by stride size, etc.

Lecture Outline

- Generative Adversarial Networks
- **Conditional GANs**
- Image-to-image translation
- Problems with GANs
- Language Models
- Sampling strategy
- Transformers

Unconditional vs conditional generation



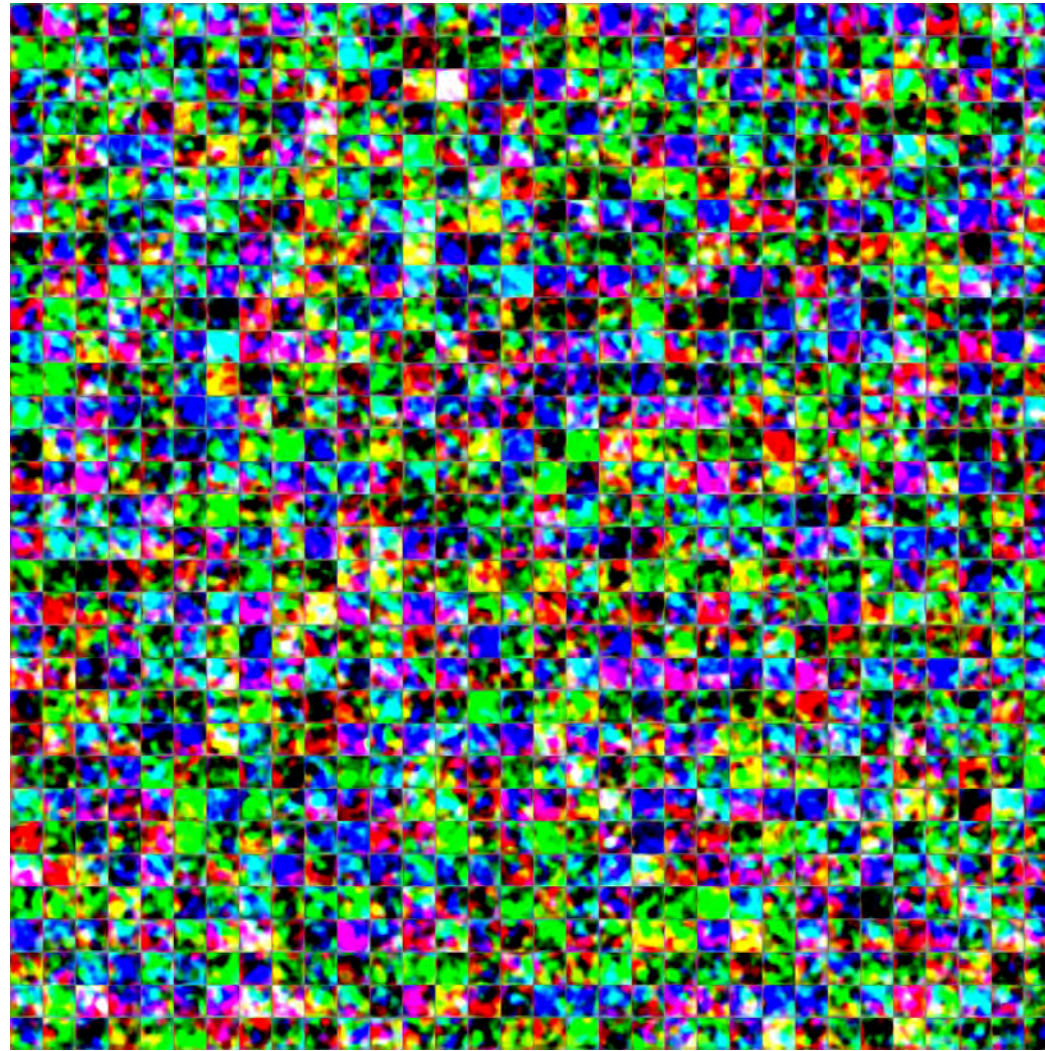
An analogy for unconditional vs conditional GANs

Hurricane example data



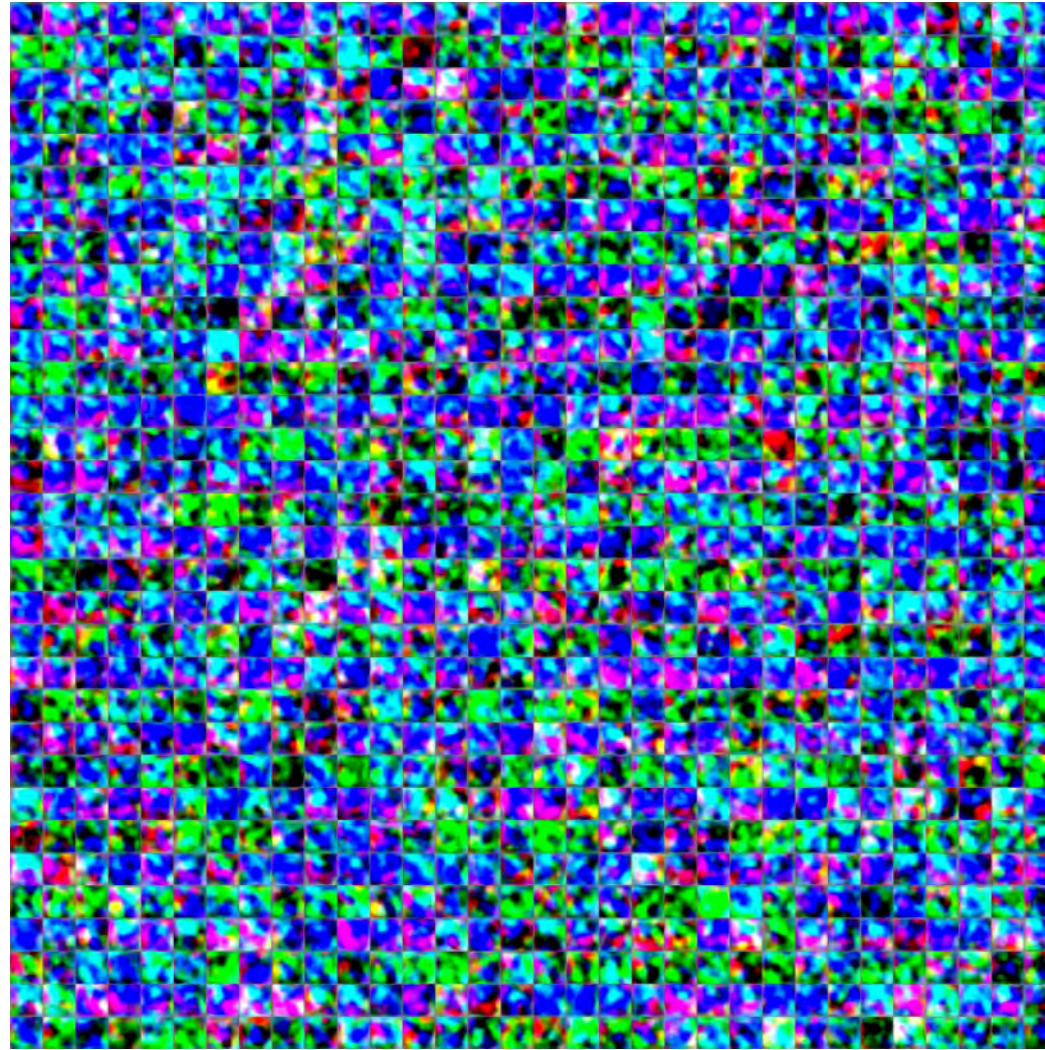
Original data

Hurricane example



Initial fakes

Hurricane example (after 54s)



Fakes after 1 iteration

Hurricane example (after 21m)



Fakes after 100 km

Hurricane example (after 47m)



Fakes after 200 king

Hurricane example (after 4h10m)



Fakes after 1000 kmg

Hurricane example (after 14h41m)



Fakes after 3700 km

Lecture Outline

- Generative Adversarial Networks
- Conditional GANs
- **Image-to-image translation**
- Problems with GANs
- Language Models
- Sampling strategy
- Transformers

Example: Deoldify images #1



A deoldified version of the famous “Migrant Mother” photograph.

Source: [Deoldify package](#).

Example: Deoldify images #2



A deoldified Golden Gate Bridge under construction.

Source: [Deoldify package](#).

Example: Deoldify images #3



Explore the latent space



Generator can't generate everything



Target



Projection

Lecture Outline

- Generative Adversarial Networks
- Conditional GANs
- Image-to-image translation
- **Problems with GANs**
- Language Models
- Sampling strategy
- Transformers

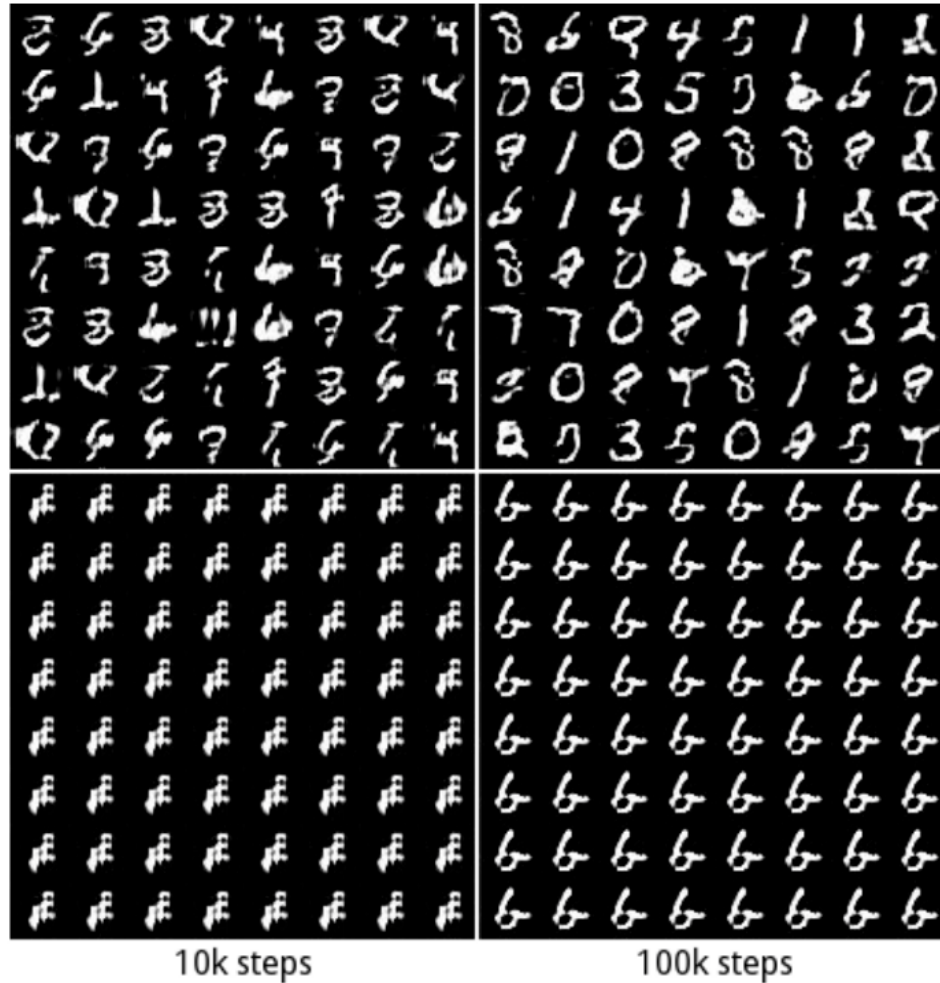
They are slow to train

StyleGAN2-ADA training times on V100s (1024x1024):

GPUs	1000 kimg	25000 kimg	sec / kimg	GPU mem	CPU mem
1	1d 20h	46d 03h	158	8.1 GB	5.3 GB
2	23h 09m	24d 02h	83	8.6 GB	11.9 GB
4	11h 36m	12d 02h	40	8.4 GB	21.9 GB
8	5h 54m	6d 03h	20	8.3 GB	44.7 GB

Source: NVIDIA's Github, [StyleGAN2-ADA — Official PyTorch implementation](#).

Mode collapse

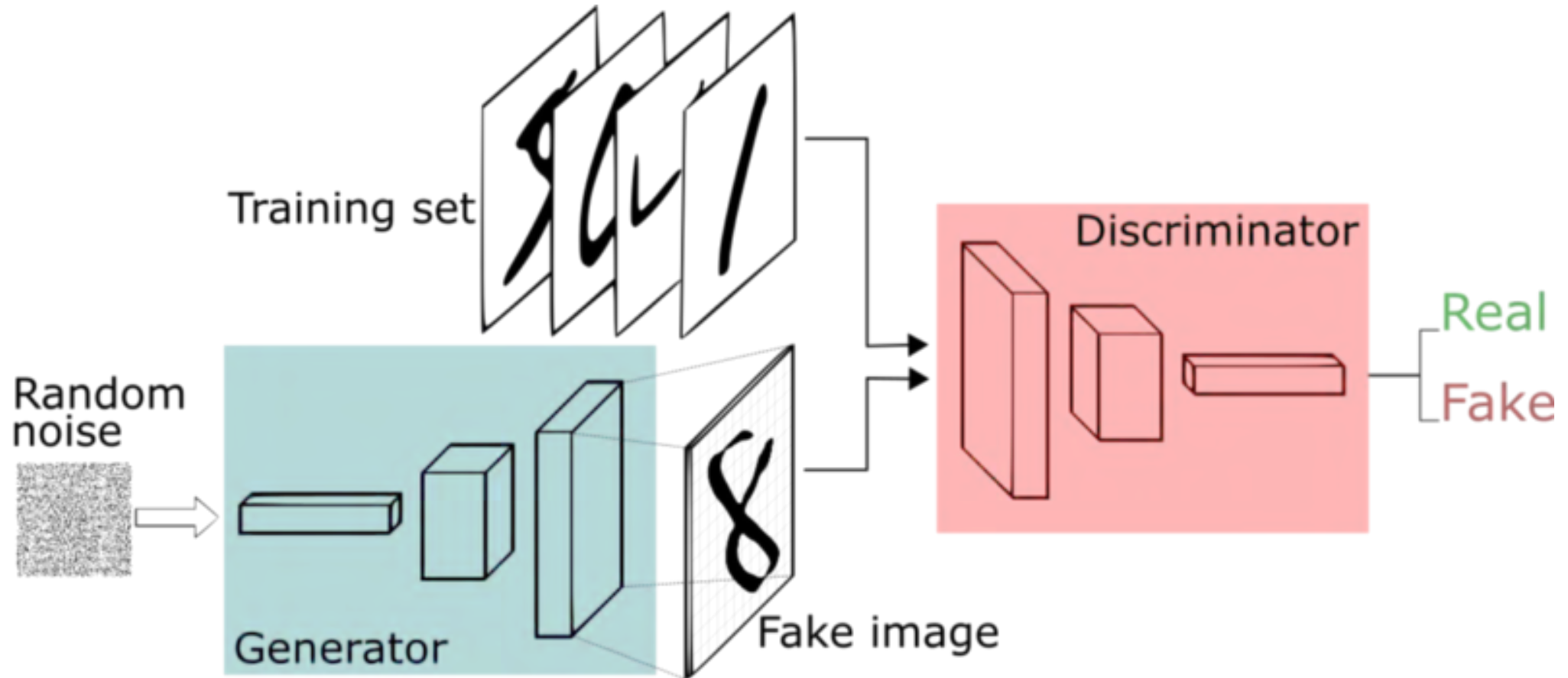


```
int getRandomNumber()
{
    return 4; // chosen by fair dice roll.
              // guaranteed to be random.
}
```

Example of mode collapse

Source: Metz et al. (2017), [Unrolled Generative Adversarial Networks](#) and Randall Munroe (2007), [xkcd #221: Random Number](#).

Generation is harder



A schematic of a generative adversarial network.

```

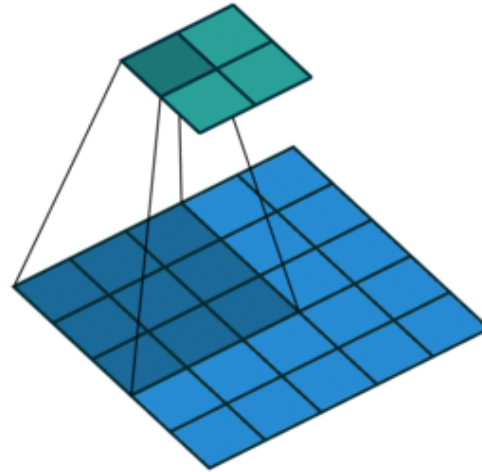
1 # Separate optimisers for discriminator and generator.
2 d_optimizer = keras.optimizers.Adam(learning_rate=0.0003)
3 g_optimizer = keras.optimizers.Adam(learning_rate=0.0004)

```

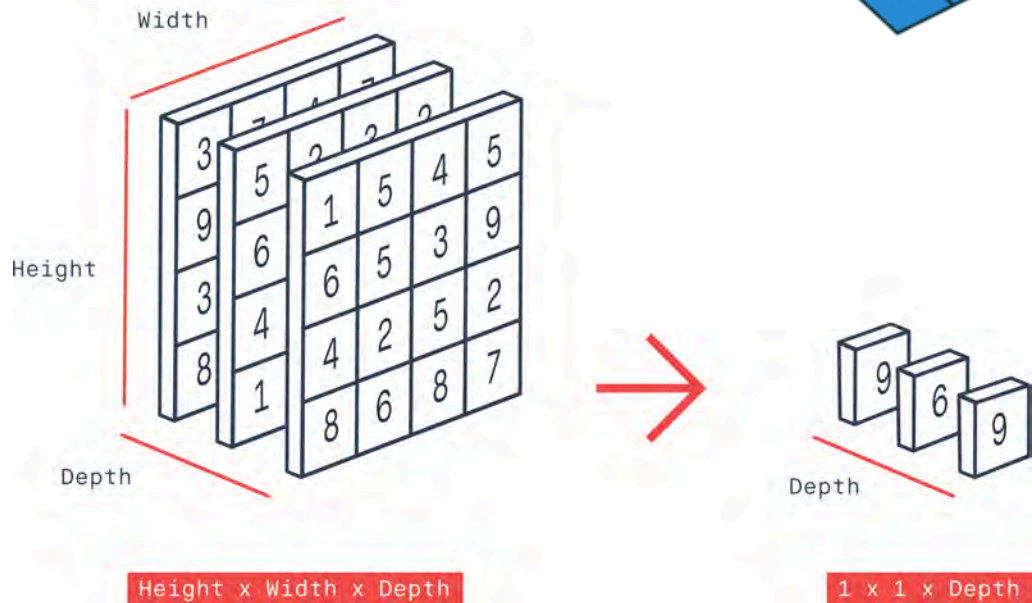
Source: Thales Silva (2018), [An intuitive introduction to Generative Adversarial Networks \(GANs\)](#), freeCodeCamp.

Advanced image layers

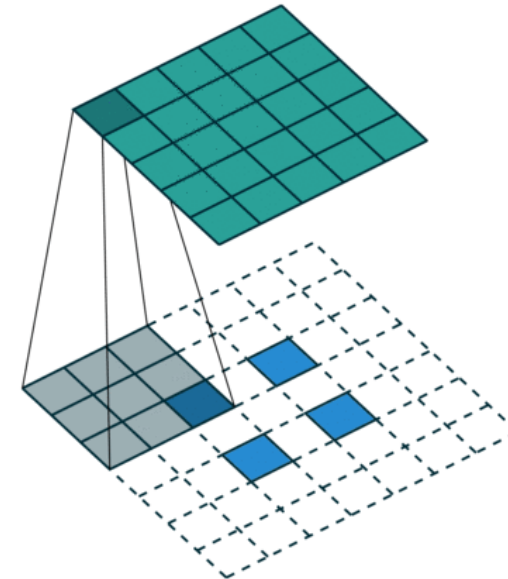
Conv2D



GlobalMaxPool2D



Conv2DTranspose



GANs with differential privacy

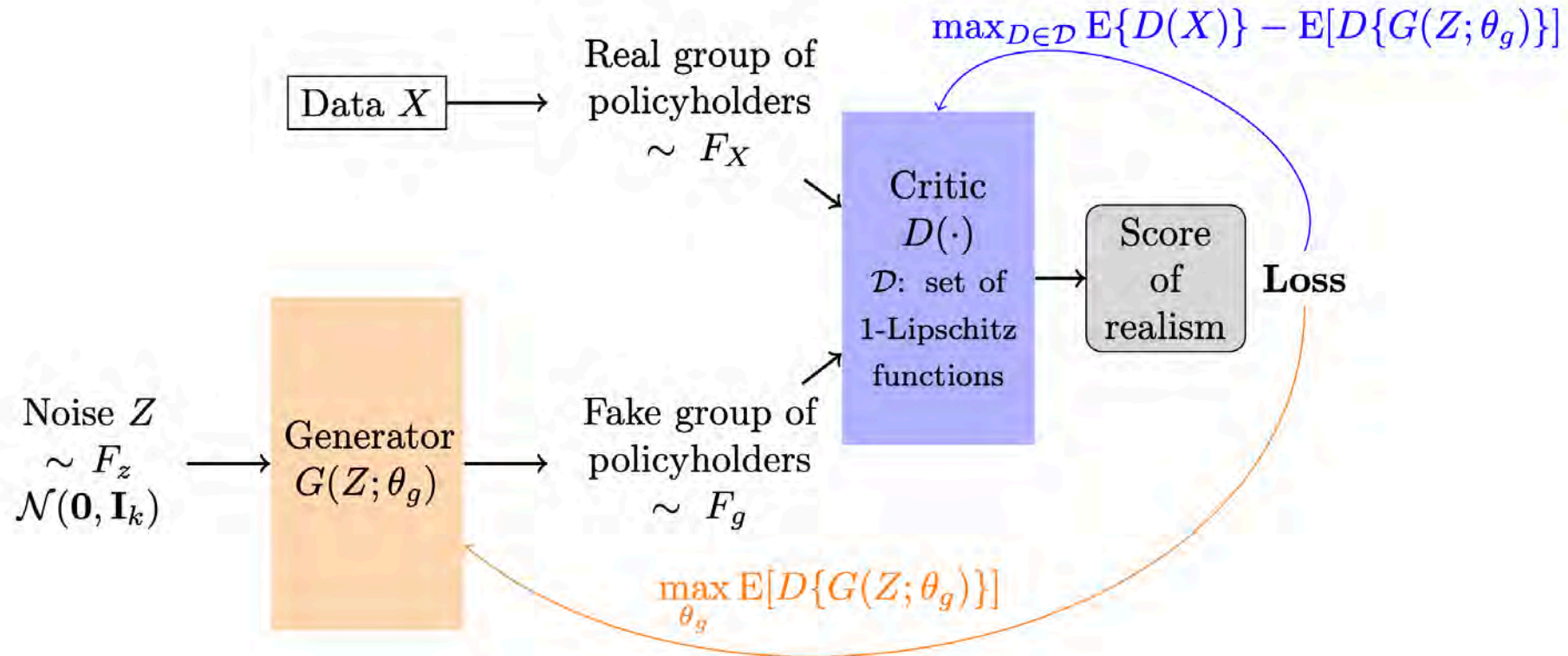


Figure 1: WGAN Schema. The arrows represent the flow of the training process.

Generating synthetic user information with *differential privacy* and *Wasserstein GANs*.

Source: Côté et al. (2020), *Synthesizing Property & Casualty Ratemaking Datasets using Generative Adversarial Networks*, arXiv.

Links

- Dongyu Liu (2021), [TadGAN: Time Series Anomaly Detection Using Generative Adversarial Networks](#)
- Jeff Heaton (2022), [GANs for Tabular Synthetic Data Generation \(7.5\)](#)
- Jeff Heaton (2022), [GANs to Enhance Old Photographs Deoldify \(7.4\)](#)

Lecture Outline

- Generative Adversarial Networks
- Conditional GANs
- Image-to-image translation
- Problems with GANs
- **Language Models**
- Sampling strategy
- Transformers

Generative deep learning

- Using AI as augmented intelligence rather than artificial intelligence.
- Use of deep learning to augment creative activities such as writing, music and art, to *generate* new things.
- Some applications: text generation, deep dreaming, neural style transfer, variational autoencoders and generative adversarial networks.

Text generation

Generating sequential data is the closest computers get to dreaming.

- Generate sequence data: Train a model to predict the next token or next few tokens in a sentence, using previous tokens as input.
- A network that models the probability of the next tokens given the previous ones is called a *language model*.

Word-level language model

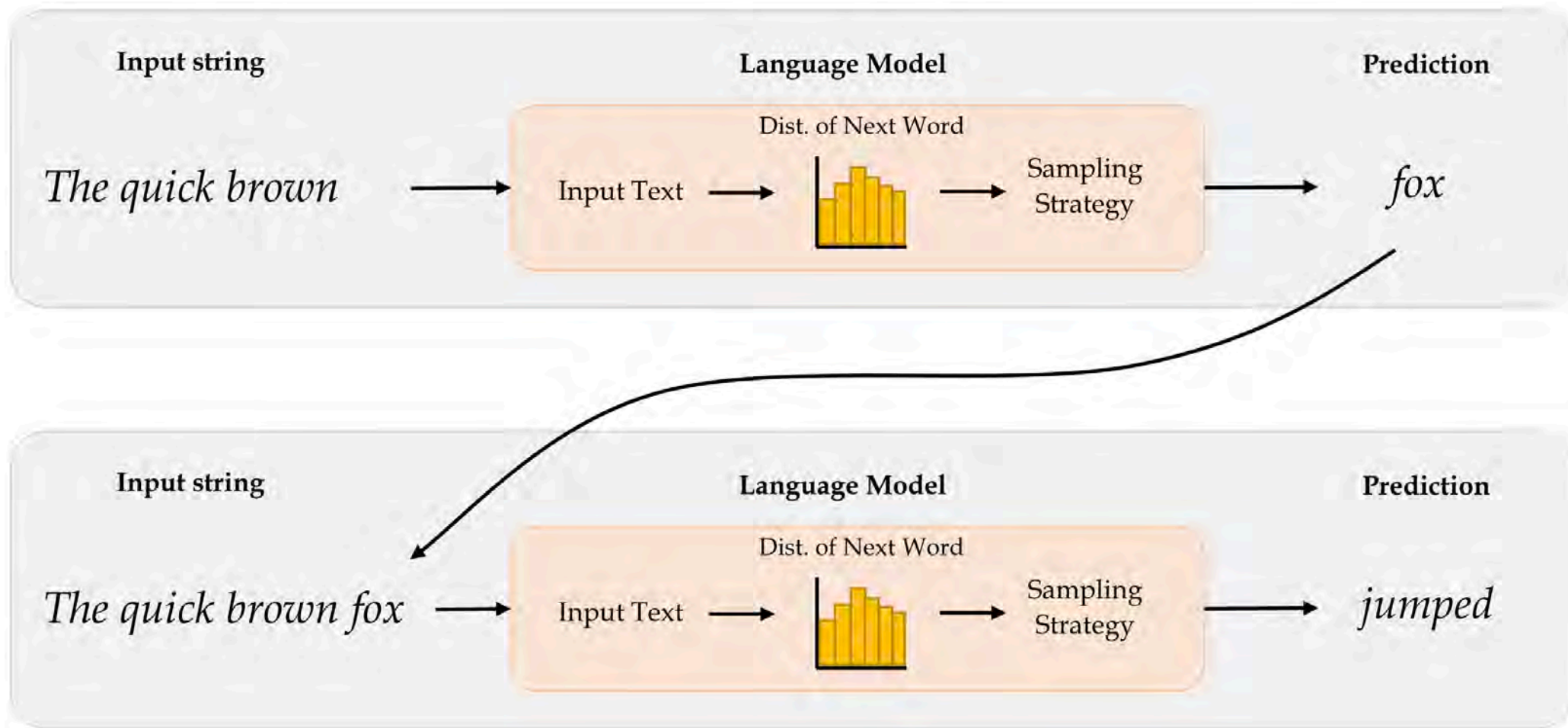


Diagram of a word-level language model.

Source: Marcus Lautier (2022).

Character-level language model

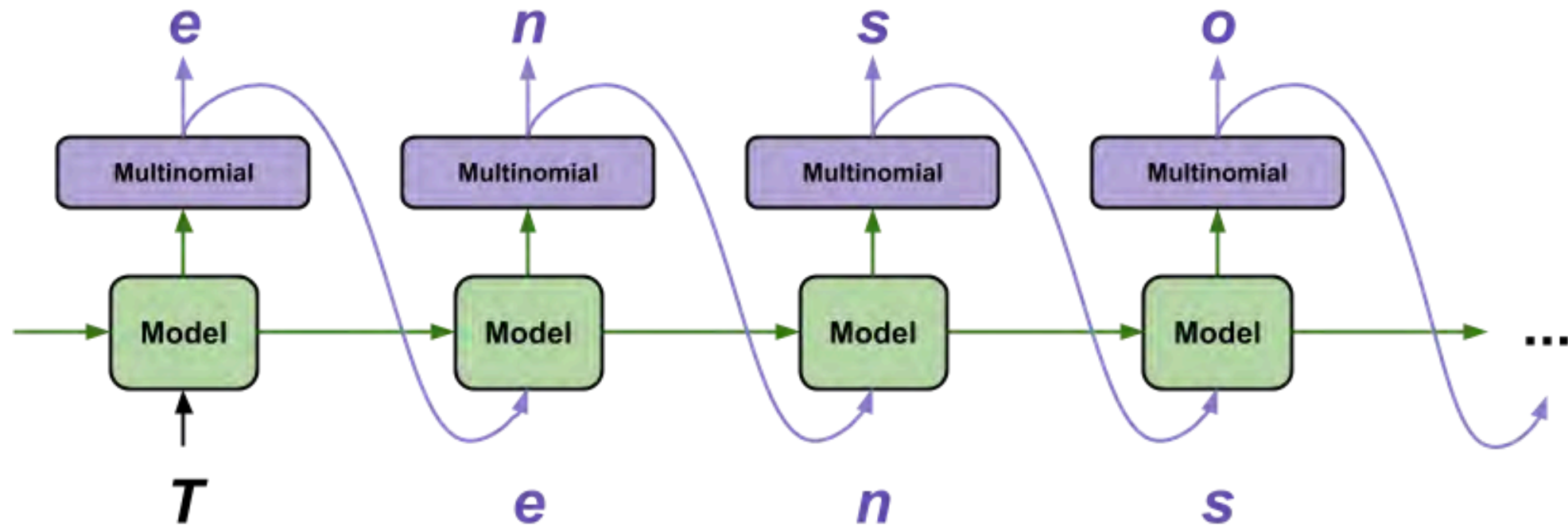


Diagram of a character-level language model (Char-RNN)

Useful for speech recognition

RNN output	Decoded Transcription
what is the weather like in bostin right now	what is the weather like in boston right now
prime miniter nerenr modi	prime minister narendra modi
arther n tickets for the game	are there any tickets for the game

Figure 1: Examples of transcriptions directly from the RNN with errors that are fixed by addition of a language model.

Generating Shakespeare I

ROMEO:

Why, sir, what think you, sir?

AUTOLYCUS:

A dozen; shall I be deceased.

The enemy is parting with your general,

As bias should still combit them offend

That Montague is as devotions that did satisfied;

But not they are put your pleasure.

Generating Shakespeare II

DUKE OF YORK:

Peace, sing! do you must be all the law;
And overmuting Mercutio slain;
And stand betide that blows which wretched shame;
Which, I, that have been complaints me older hours.

LUCENTIO:

What, marry, may shame, the forish priest-lay estimest you, sir,
Whom I will purchase with green limits o' the commons' ears!

Generating Shakespeare III

ANTIGONUS:

To be by oath enjoin'd to this. Farewell!

The day frowns more and more: thou'rt like to have

A lullaby too rough: I never saw

The heavens so dim by day. A savage clamour!

[Exit, pursued by a bear]



Lecture Outline

- Generative Adversarial Networks
- Conditional GANs
- Image-to-image translation
- Problems with GANs
- Language Models
- **Sampling strategy**
- Transformers

Sampling strategy

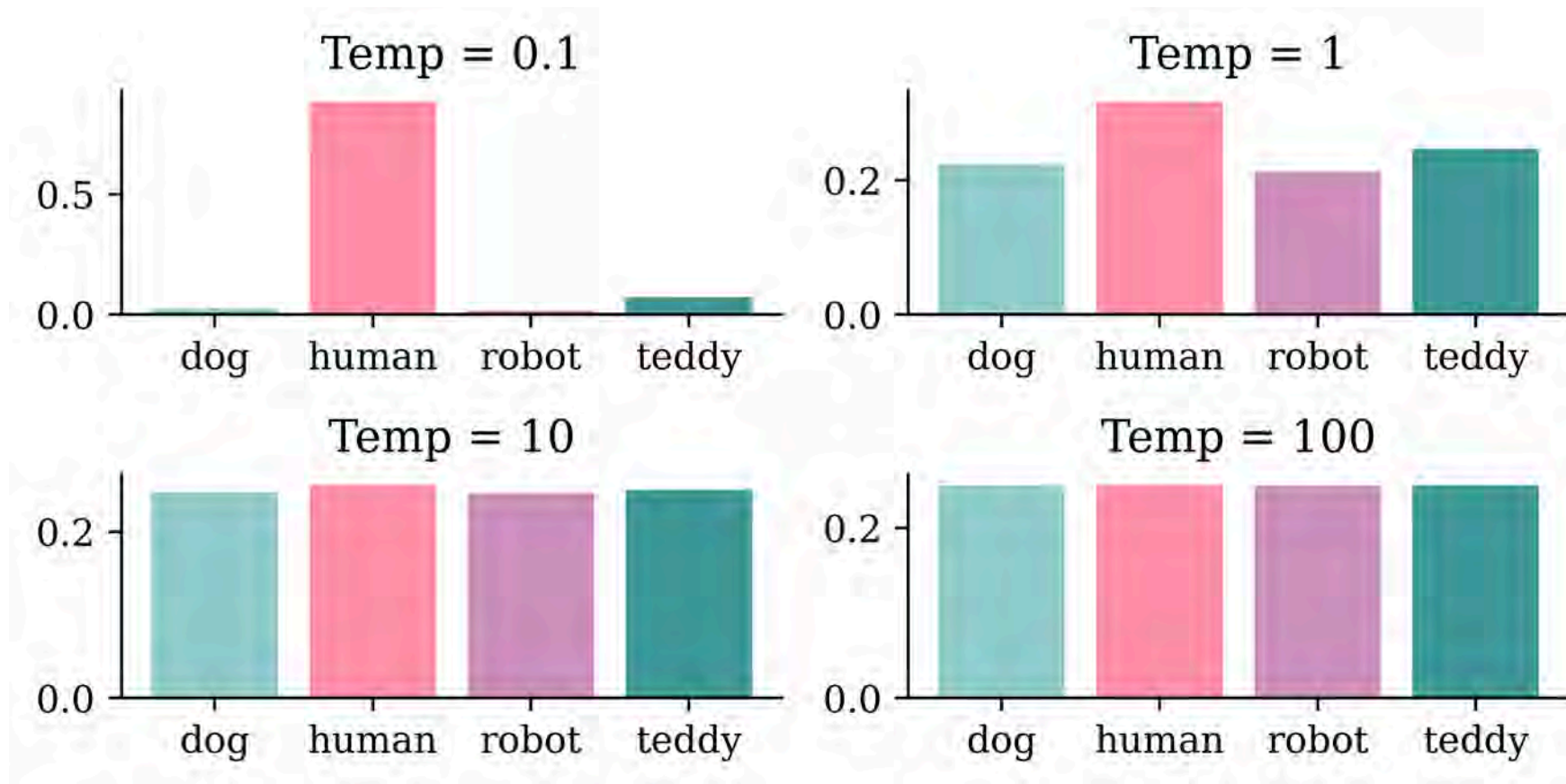
- *Greedy sampling* will choose the token with the highest probability. It makes the resulting sentence repetitive and predictable.
- *Stochastic sampling*: if a word has probability 0.3 of being next in the sentence according to the model, we'll choose it 30% of the time. But the result is still not interesting enough and still quite predictable.
- Use a *softmax temperature* to control the randomness. More randomness results in more surprising and creative sentences.

Softmax temperature

- The softmax temperature is a parameter that controls the randomness of the next token.
- The formula is:

$$\text{softmax}_{\text{temperature}}(x) = \frac{\exp(x/\text{temperature})}{\sum_i \exp(x_i/\text{temperature})}$$

“I am a” ...



Generating Laub (temp = 0.01)

In today's lecture we will be different situation. So, next one is what they rective that each commit to be able to learn some relationships from the course, and that is part of the image that it's very clesse and black problems that you're trying to fit the neural network to do there instead of like a specific though shaf series of layers mean about full of the chosen the baseline of car was in the right, but that's an important facts and it's a very small summary with very scrotr by the beginning of the sentence.

Generating Laub (temp = 0.25)

In today's lecture we will decrease before model that we have to think about it, this might be better, for chat the same project, because you might use the test set because it's to be picked up the things that I wanted to hear of things that I like that even real you and you're using the same thing again now because we need to understand what it's doing the same thing but instead of putting it in particular week, and we can say that's a thing I mainly link it's three columns.

Generating Laub (temp = 0.5)

In today's lecture we will probably the adw n wait lots of ngobs teulagedation to calculate the gradient and then I'll be less than one layer the next slide will br input over and over the threshow you ampaigey the one that we want to apply them quickly. So, here this is the screen here the main top kecw onct three thing to told them, and the output is a vertical variables and Marceparase of things that you're moving the blurring and that just data set is to maybe kind of categorical variants here but there's more efficiently not basically replace that with respect to the best and be the same thing.

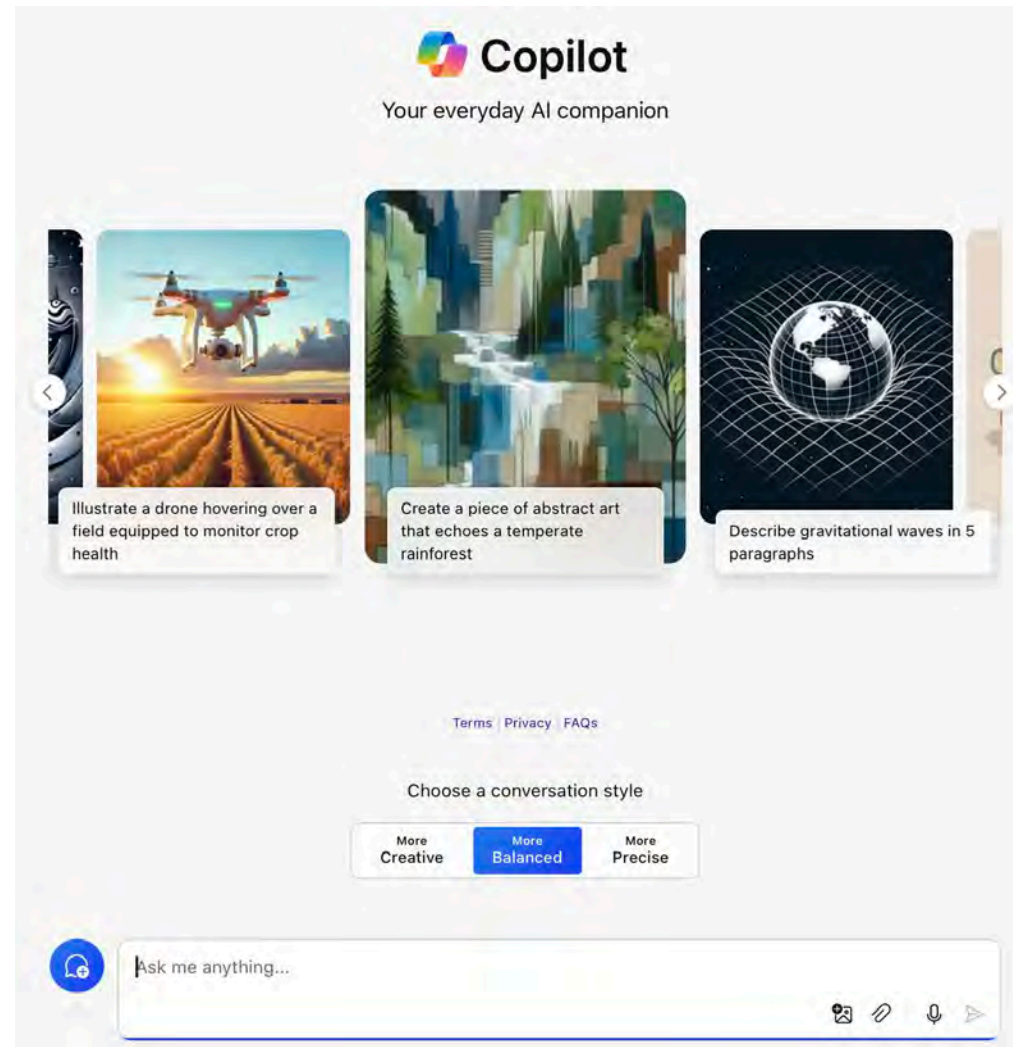
Generating Laub (temp = 1)

In today's lecture we will put it different shates to touch on last week, so I want to ask what are you object frod current. They don't have any zero into it, things like that which mistakes. 10 claims that the average version was relden distever ditgs and Python for the whole term wo long right to really. The name of these two options. There are in that seems to be modified version. If you look at when you're putting numbers into your, that that's over. And I went backwards, up, if they'rina functional pricing working with.

Generating Laub (temp = 1.5)

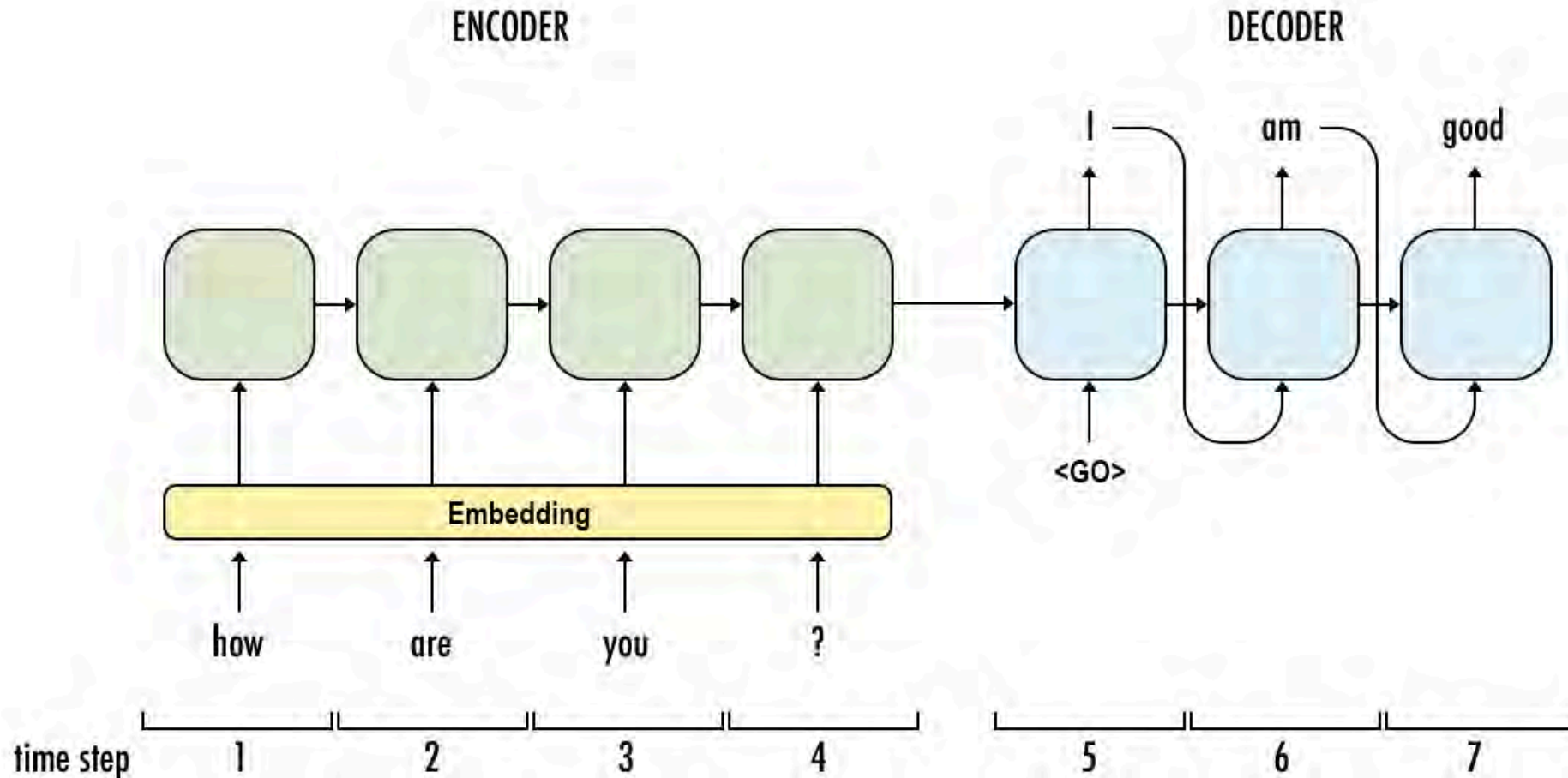
In today's lecture we will put it could be bedinnth. Lowerstoriage nruron. So rochain the everything that I just sGiming. If there was a large. It's gonua draltionation. Tow many, up, would that black and 53% that's girter thankAty will get you just typically stickK thing. But maybe. Anyway, I'm going to work on this libry two, past, at shit citcs just pleming to memorize overcamples like pre pysing, why wareed to smart a one in this reportbryeccuriay.

Copilot's "Conversation Style"



This is (probably) just the 'temperature' knob under the hood.

Generate the most likely sequence



An example sequence-to-sequence chatbot model.

Source: Payne (2021), [What is beam search](#), Width.ai blog.

Beam search

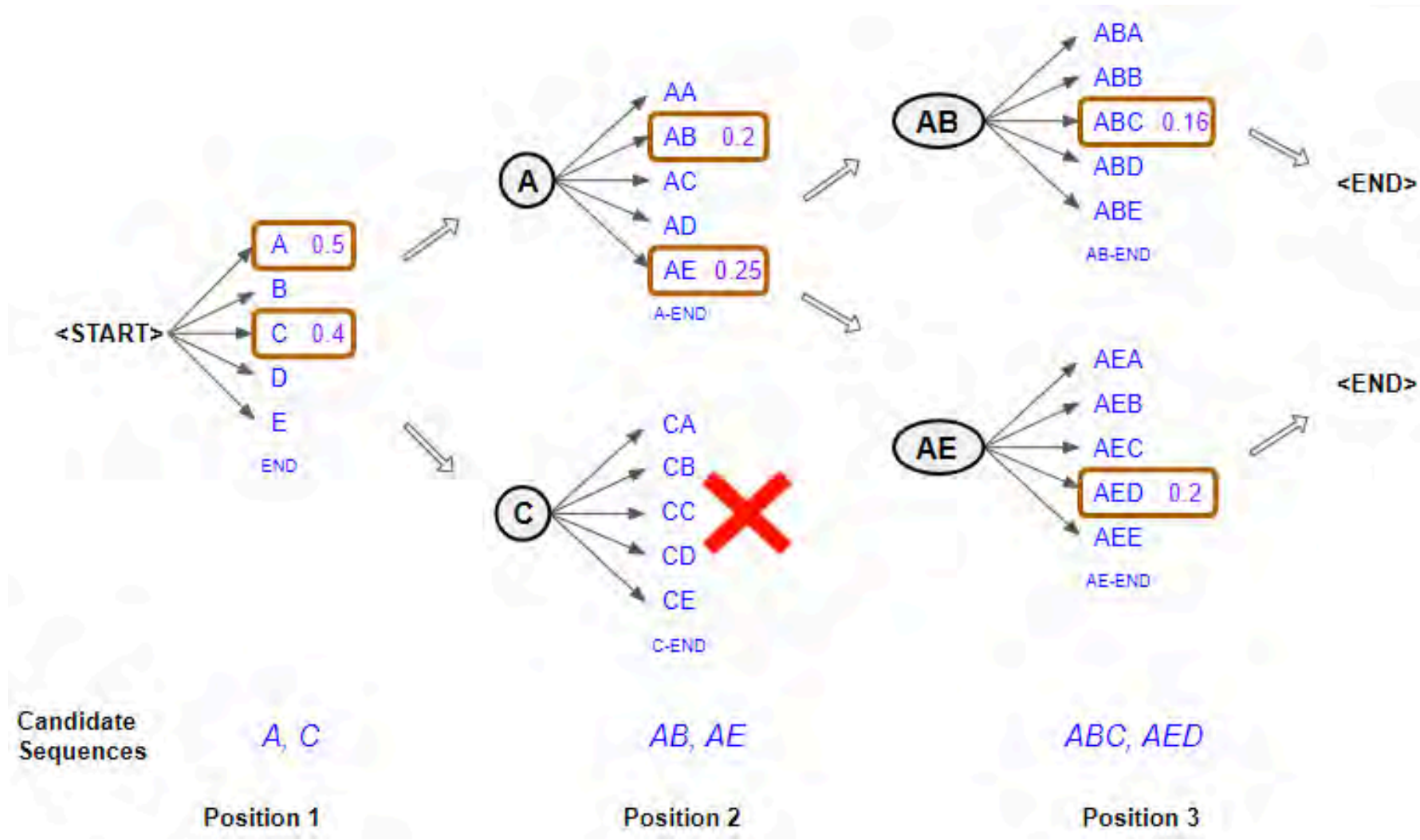


Illustration of a beam search.

Source: Doshi (2021), [Foundations of NLP Explained Visually: Beam Search, How It Works](https://towardsdatascience.com/foundations-of-nlp-explained-visually-beam-search-how-it-works/), towardsdatascience.com.

Lecture Outline

- Generative Adversarial Networks
- Conditional GANs
- Image-to-image translation
- Problems with GANs
- Language Models
- Sampling strategy
- **Transformers**

Transformer architecture

GPT makes use of a mechanism known as attention, which removes the need for recurrent layers (e.g., LSTMs). It works like an information retrieval system, utilizing queries, keys, and values to decide how much information it wants to extract from each input token.

Attention heads can be grouped together to form what is known as a multihead attention layer. These are then wrapped up inside a Transformer block, which includes layer normalization and skip connections around the attention layer. Transformer blocks can be stacked to create very deep neural networks.

Highly recommended viewing: Iulia Turk (2021), *Transfer learning and Transformer models*, ML Tech Talks.

😊 Transformers package

```
1 import transformers
2 from transformers import pipeline
3 generator = pipeline(task="text-generation", model="gpt2", revision="6c0e608")
```

Loading weights: 100%

148/148 [00:00<00:00, 4507.65it/s, Materializing param=transformer.wte.weight]

```
1 transformers.set_seed(123)
2 print(generator("It's the holidays so I'm going to enjoy")[0]["generated_text"])
```

It's the holidays so I'm going to enjoy it," she said. "I don't have to worry about going to Disneyland anymore. I want to go back to my original life and go to Disneyland so that I can enjoy my vacation."

```
1 transformers.set_seed(234)
2 print(generator("It's the holidays so I'm going to enjoy")[0]["generated_text"])
```

It's the holidays so I'm going to enjoy it a lot.

I've been thinking about this for the last 12 hours. It's just a matter of time before I have a chance to go and see how the rest of the team does.

But I've got a lot of friends who live in the same area and they all agree that Christmas is always a big time of year, so I think I'll enjoy it a lot.

You have a lot of work to do, so I'm going to have to do it all the time.

Reading the course profile

```
1 context = ""
2 StoryWall Formative Discussions: An initial StoryWall, worth 2%, is due by noon on June 3
3 The project will be submitted in stages: draft due at noon on July 1 (10%), recorded pres
4
5 As a student at UNSW you are expected to display academic integrity in your work and inte
6 To assist you in understanding what academic integrity means, and how to ensure that you
7
8 StoryWall (30%)
9
10 The StoryWall format will be used for small weekly questions. Each week of questions will
11
12 Project (40%)
13
14 Over the term, students will complete an individual project. There will be a selection of
15
16 The deliverables for the project will include: a draft/progress report mid-way through th
17
18 Exam (30%)
19
20 The exam will test the concepts presented in the lectures. For example, students will be
21 ""
```

Question answering

```
1 qa = pipeline("question-answering", model="distilbert-base-cased-distilled-squad")
```

Loading weights: 100%

102/102 [00:00<00:00, 2951.07it/s, Materializing param=qa_outputs.weight]

```
1 qa(question="What weight is the exam?", context=context)
```

```
{'score': 0.5020921107206959, 'start': 2092, 'end': 2095, 'answer': '30%'}
```

```
1 qa(question="What topics are in the exam?", context=context)
```

```
{'score': 0.2127583920955658,
 'start': 1778,
 'end': 1791,
 'answer': 'deep learning'}
```

```
1 qa(question="When is the presentation due?", context=context)
```

```
{'score': 0.5296497344970703,
 'start': 1319,
 'end': 1335,
 'answer': 'Monday at midday'}
```

```
1 qa(question="How many StoryWall tasks are there?", context=context)
```

```
{'score': 0.2139095962047577, 'start': 1155, 'end': 1158, 'answer': '30%'}
```

ChatGPT is Transformer + RLHF

“... there is no official paper that describes how ChatGPT works in detail, but ... we know that it uses a technique called reinforcement learning from human feedback (RLHF) to fine-tune the GPT-3.5 model. While ChatGPT still has many limitations (such as sometimes “hallucinating” factually incorrect information), it is a powerful example of how Transformers can be used to build generative models that can produce complex, long-ranging, and novel output that is often indistinguishable from human-generated text. The progress made thus far by models like ChatGPT serves as a testament to the potential of AI and its transformative impact on the world.”

Next Steps

Two new courses starting in 2026:

ACTL4306 “Quantitative Ethical AI for Risk & Actuarial Applications”

ACTL4307 “Generative AI for Actuaries”

Glossary

- beam search
- bias
- ChatGPT (& RLHF)
- generative adversarial networks
- greedy sampling
- Hugging Face
- language model
- latent space
- softmax temperature
- stochastic sampling