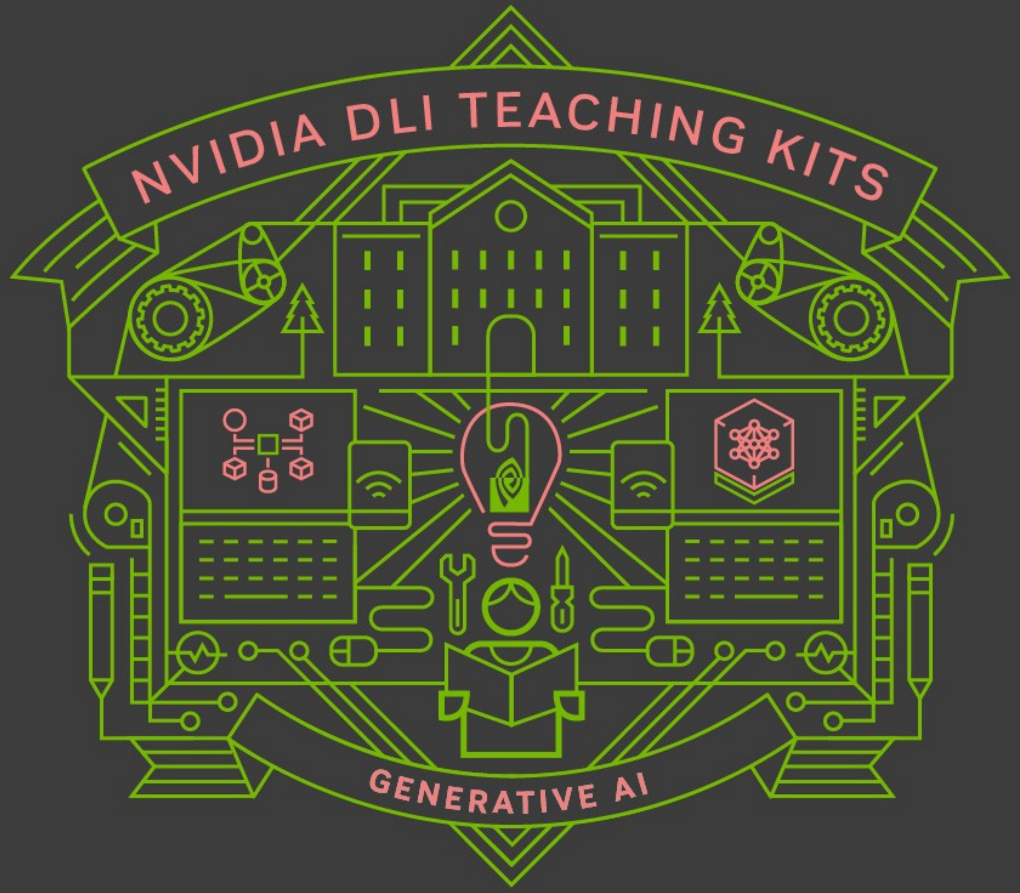
NVIDIA

DARTMOUTH ENGINEERING

Lecture 1.1 – Introduction to the Generative AI Course

Generative AI Teaching Kit

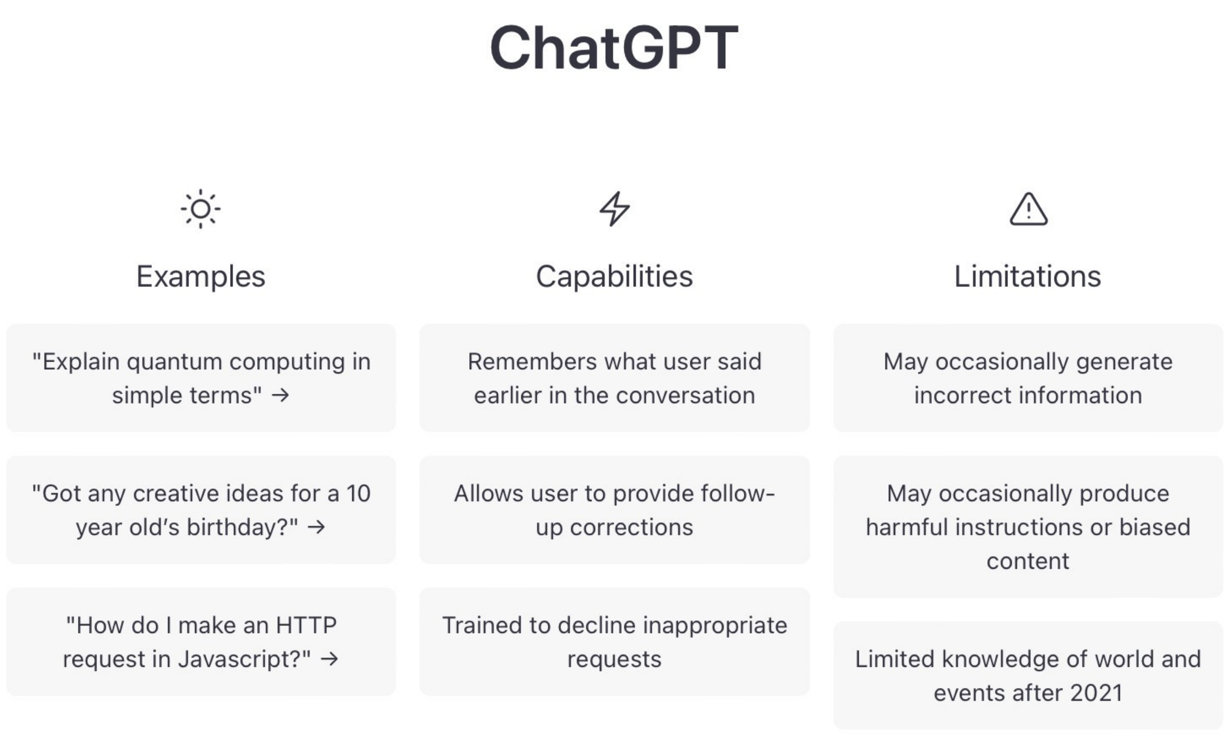


Course Structure

Motivation for this course

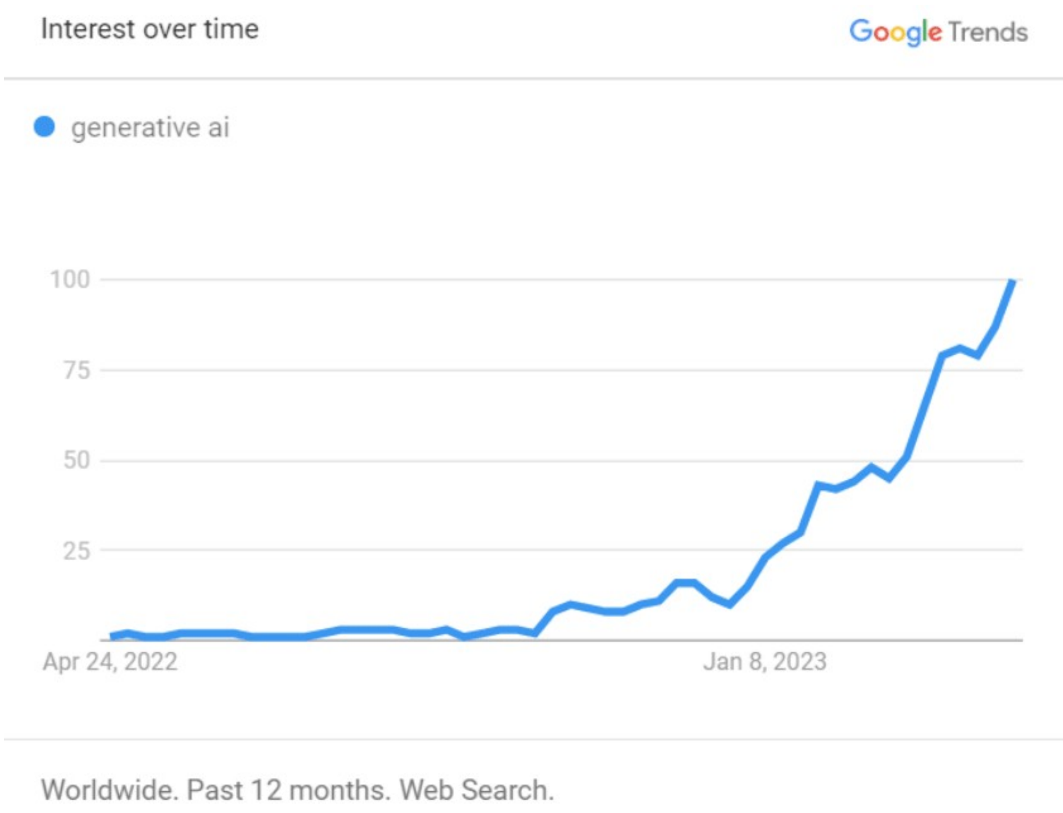
Over the last two years, you've probably heard all of these GenAI buzzwords:

- LLMs
- Chatbots
- RAG
- Agents
- Image/Video generation
- Deep Fakes

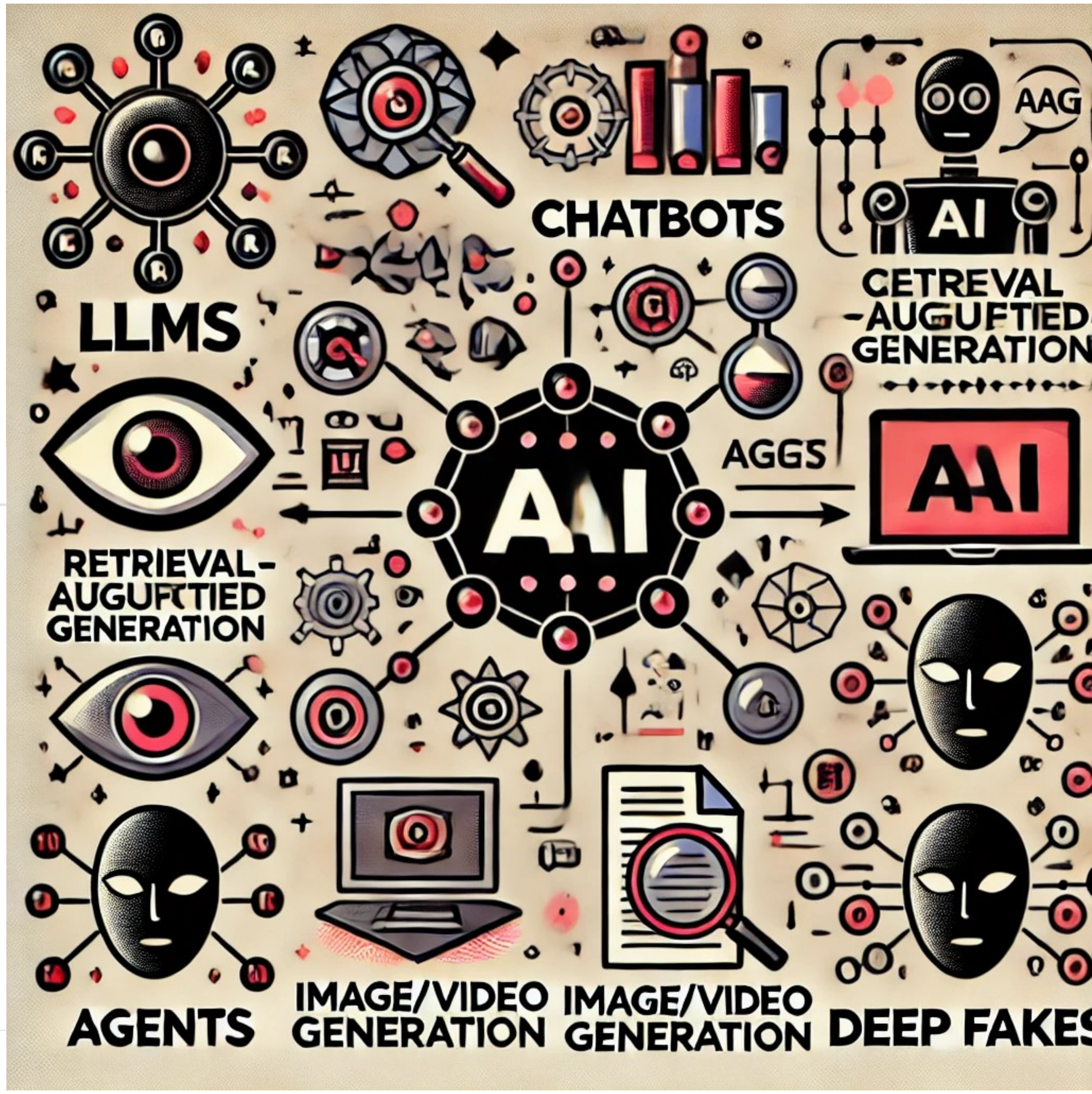


ChatGPT

Examples	Capabilities	Limitations
"Explain quantum computing in simple terms" →	Remembers what user said earlier in the conversation	May occasionally generate incorrect information
"Got any creative ideas for a 10 year old's birthday?" →	Allows user to provide follow-up corrections	May occasionally produce harmful instructions or biased content
"How do I make an HTTP request in Javascript?" →	Trained to decline inappropriate requests	Limited knowledge of world and events after 2021



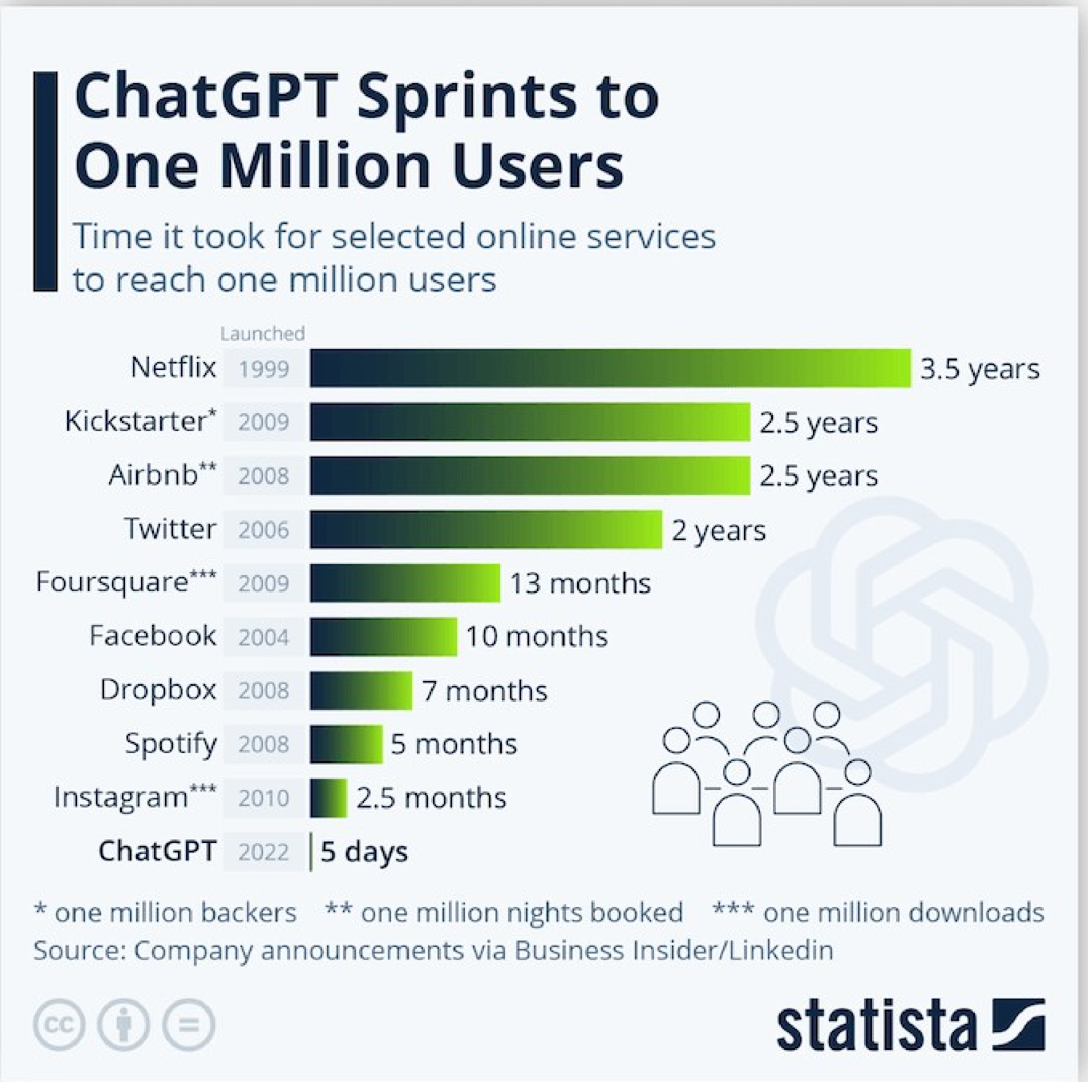
Worldwide. Past 12 months. Web Search.



Motivation for this course

ChatGPT Changed the world of knowledge forever

- Released Nov 2022 Took the world by storm
- How does it know so much?
- Is it “thinking” ?
- What is it made of?



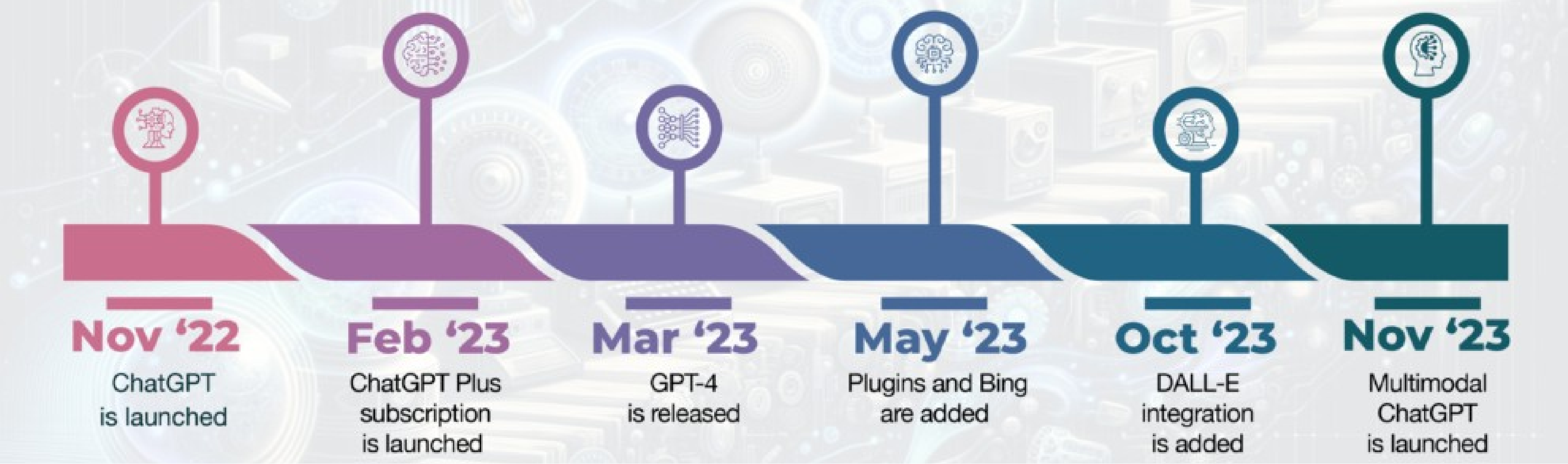
ChatGPT Sprints to One Million Users

Time it took for selected online services to reach one million users

* one million backers ** one million nights booked *** one million downloads

Source: Company announcements via Business Insider/Linkedin

statista



Nov ‘22 ChatGPT is launched

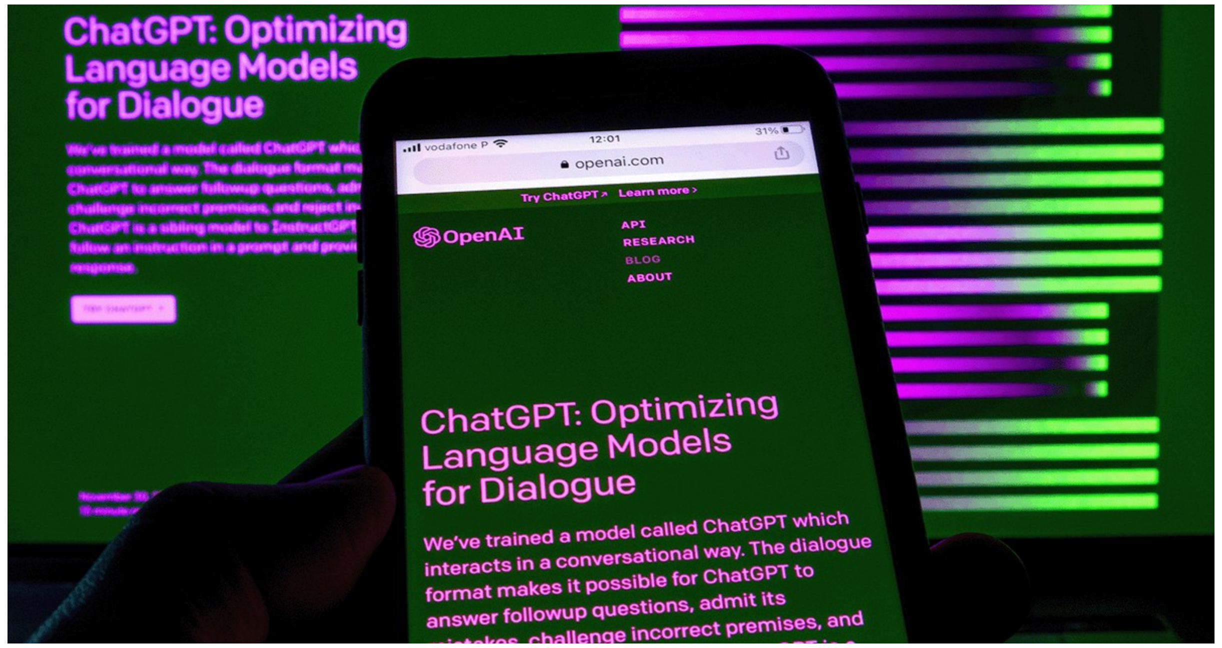
Feb ‘23 ChatGPT Plus subscription is launched

Mar ‘23 GPT-4 is released

May ‘23 Plugins and Bing are added

Oct ‘23 DALL-E integration is added

Nov ‘23 Multimodal ChatGPT is launched

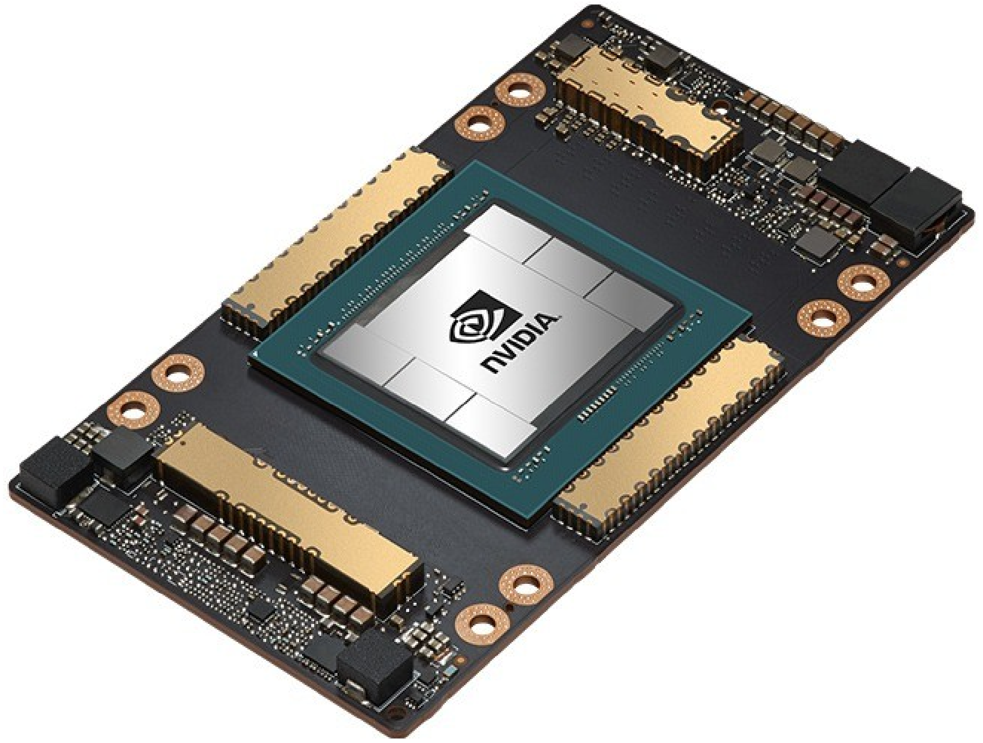


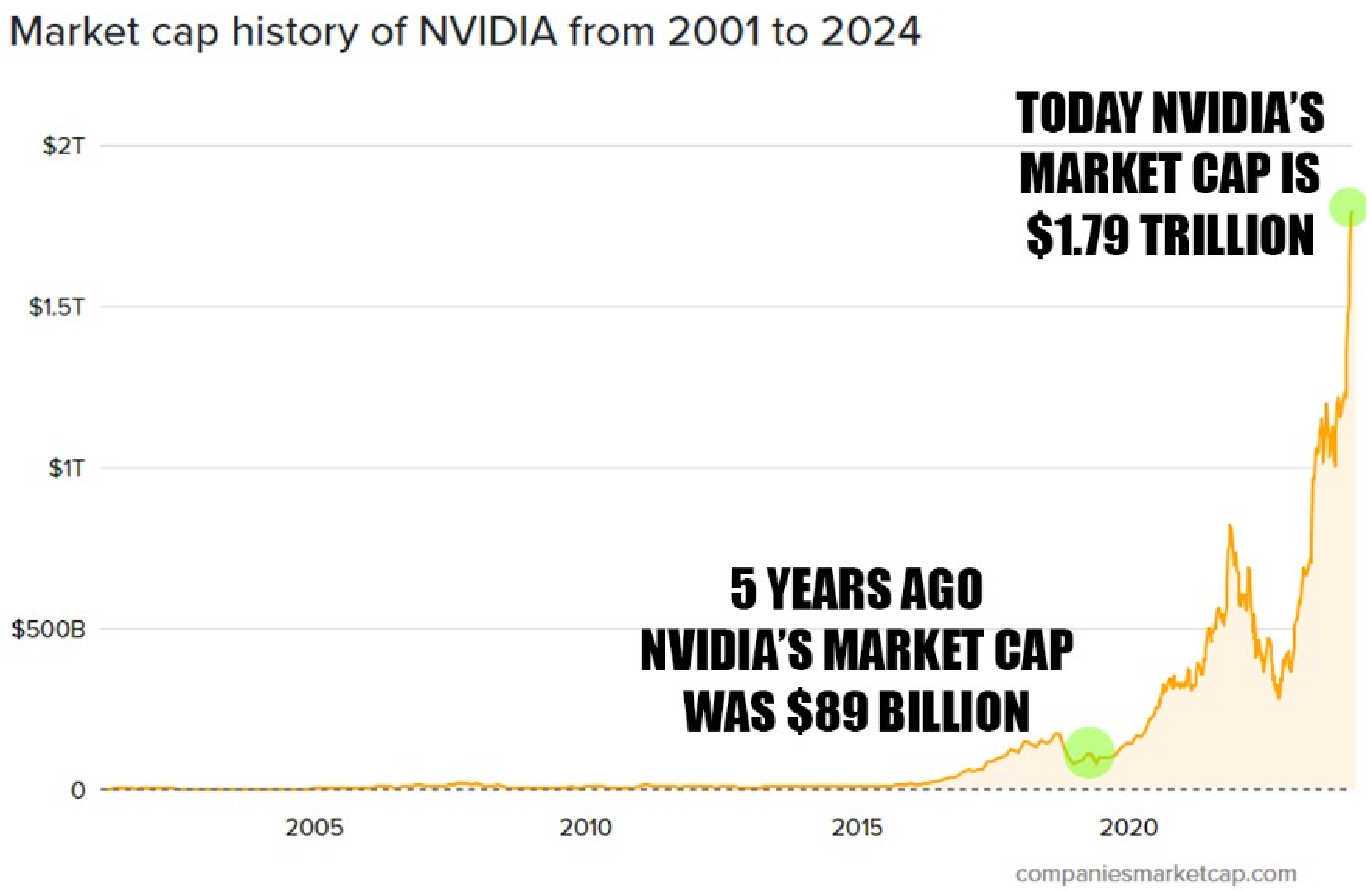
ChatGPT: Optimizing Language Models for Dialogue
openai.com
OpenAI
API
RESEARCH
BLOG
ABOUT
ChatGPT: Optimizing Language Models for Dialogue
We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and

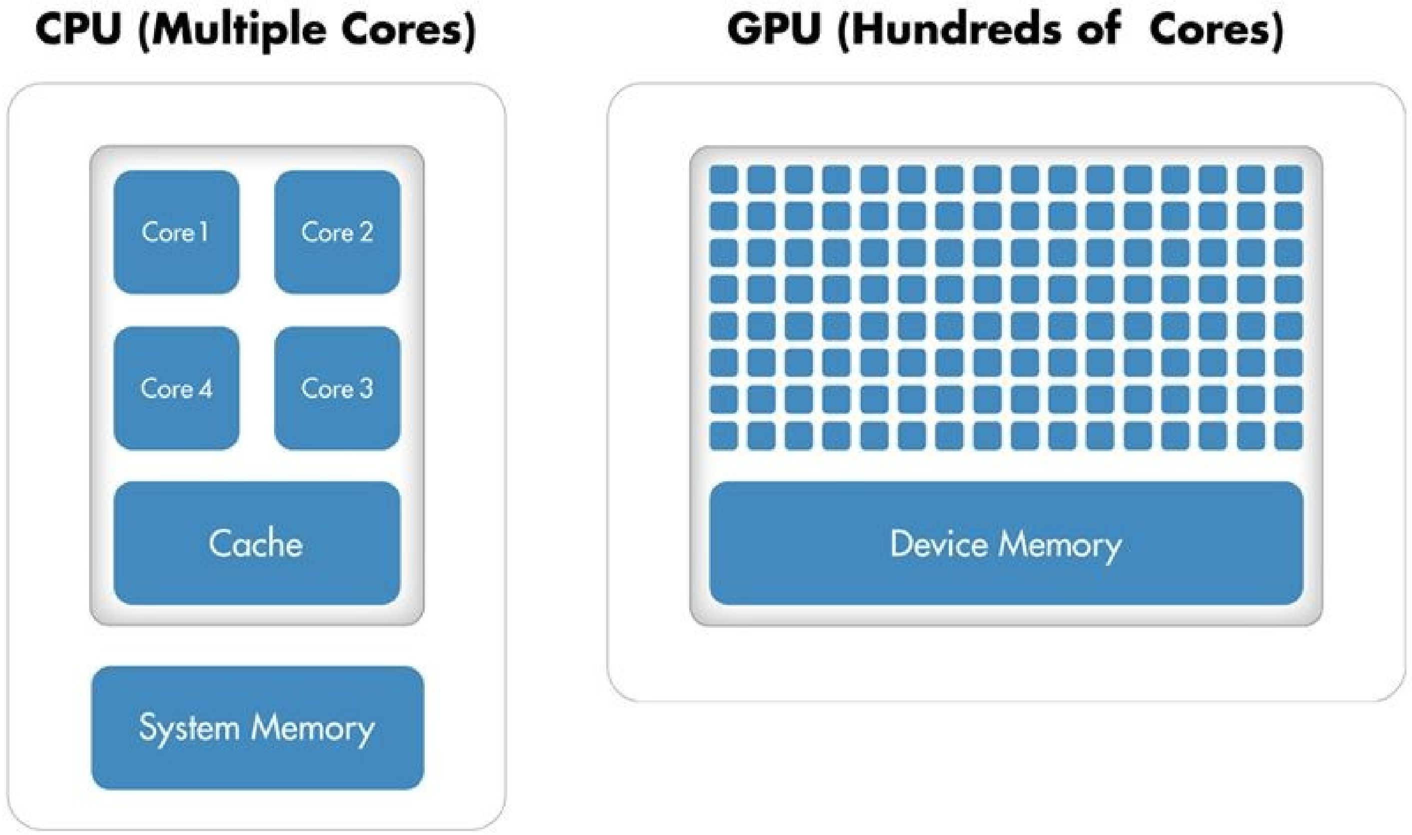
Motivation for this course

In 2024, NVIDIA's market cap reached 3T! Why?

Why are GPUs now the hottest commodity for GenAI?

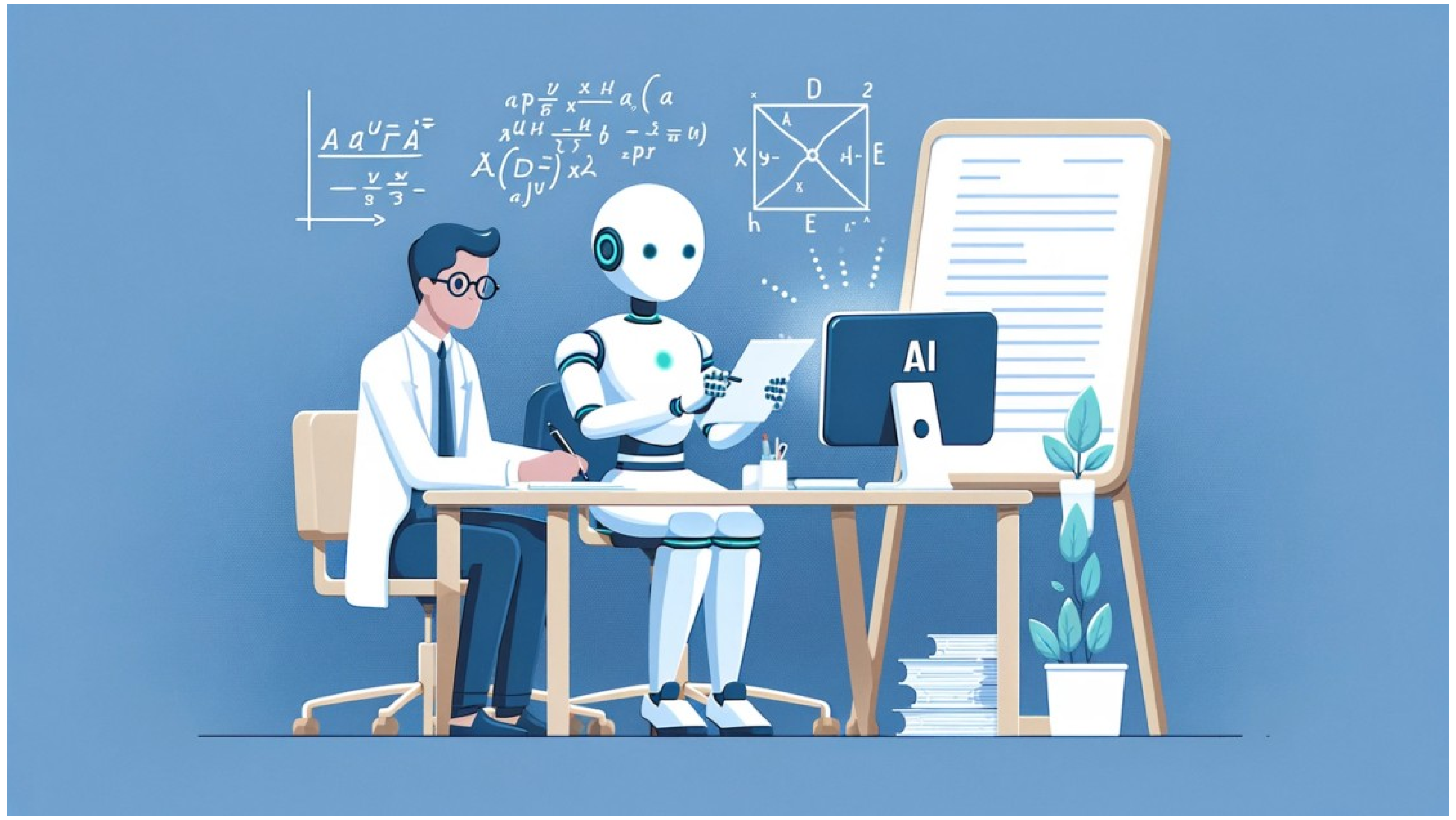



Market cap history of NVIDIA from 2001 to 2024
TODAY NVIDIA'S MARKET CAP IS $1.79 TRILLION
5 YEARS AGO NVIDIA'S MARKET CAP WAS $89 BILLION
$2T
$1.5T
$1T
$500B
0
2005
2010
2015
2020
companiesmarketcap.com



Motivation for this course

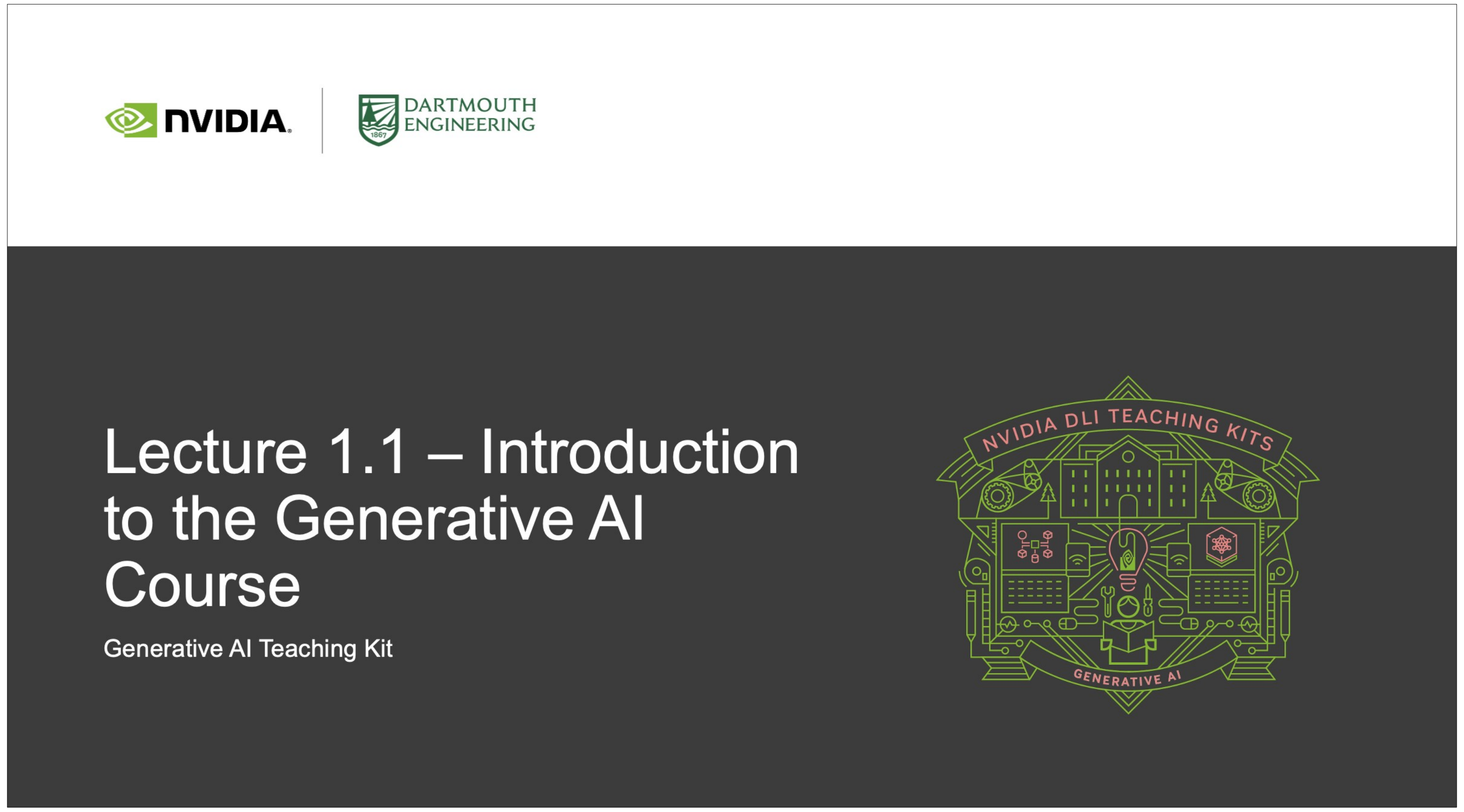
- GenAI poses a challenge for Academia
- Pace of development has far outpaced adoption in schooling
- How do we do teaching given these extremely knowledgeable, highly accessible tools?
- Familiarity with the abilities and limits of these tools is vital to maintain competitiveness for both teachers and students



Structure of the course

How each module of the course is designed

- Online Syllabus
  - Lays out entire course including planned future content
- Lecture Slides
  - Introduce content
- Demo Notebooks
  - Show the tools in practice
- Knowledge Checks
  - Keeping track of the content
- Lab Notebooks
  - Your turn to build with the tools
- DLI Online Courses
  - Online, self-paced courses offering certificate



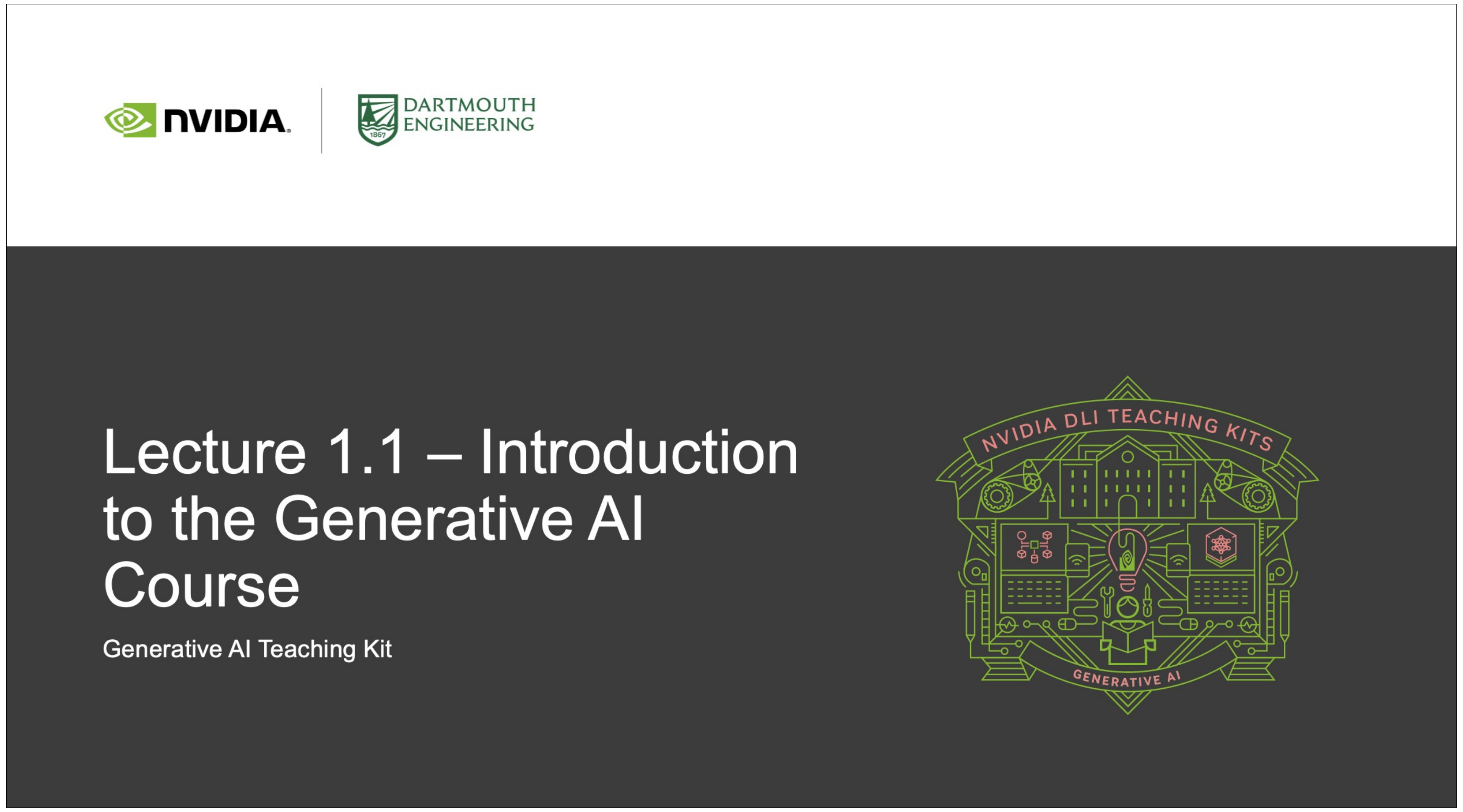
Lecture 1.1 – Introduction to the Generative AI Course

Generative AI Teaching Kit

Structure of the course

Lecture Slides

- 2 or three lectures per module
- Designed to introduce the necessary information needed to understand what/how things work
- Not deep mathematical proofs, more applied science/engineering focus



NVIDIA | DARTMOUTH ENGINEERING

Lecture 1.1 – Introduction to the Generative AI Course

Generative AI Teaching Kit

Structure of the course

Demo Notebooks

- Designed to be used during class time for the instructor to show how these tools/topics can work
- Built with an interactive Python notebook, namely Google Colab

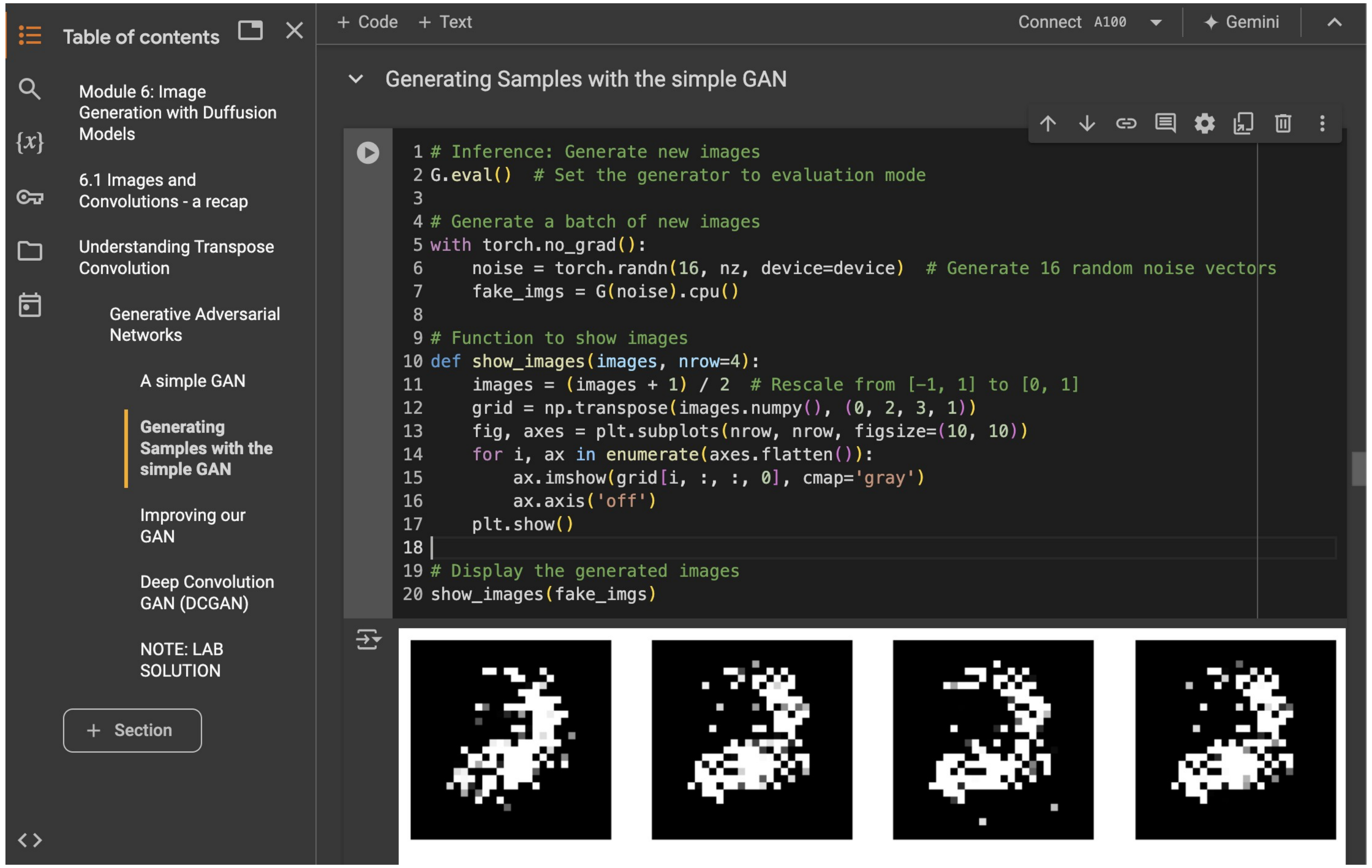


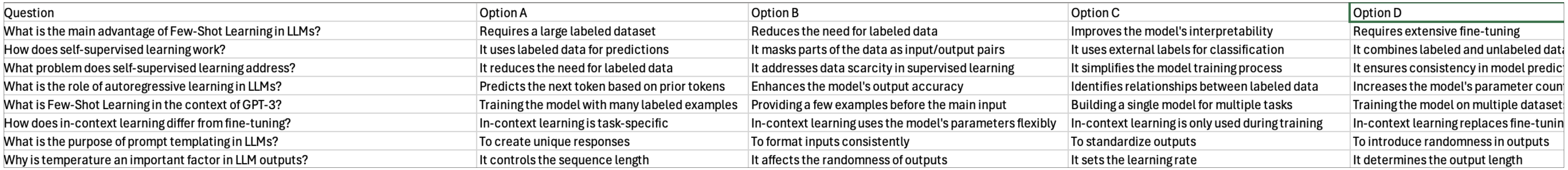
Table of contents
Module 6: Image Generation with Duffusion Models
6.1 Images and Convolutions - a recap
Understanding Transpose Convolution
Generative Adversarial Networks
A simple GAN
Generating Samples with the simple GAN
Improving our GAN
Deep Convolution GAN (DCGAN)
NOTE: LAB SOLUTION
+ Section
+ Code
+ Text
Connect A100
Gemini
Generating Samples with the simple GAN

```python
# Inference: Generate new images
G.eval()  # Set the generator to evaluation mode

# Generate a batch of new images
with torch.no_grad():
    noise = torch.randn(16, nz, device=device)  # Generate 16 random noise vectors
    fake_imgs = G(noise).cpu()

# Function to show images
def show_images(images, nrow=4):
    images = (images + 1) / 2  # Rescale from [-1, 1] to [0, 1]
    grid = np.transpose(images.numpy(), (0, 2, 3, 1))
    fig, axes = plt.subplots(nrow, nrow, figsize=(10, 10))
    for i, ax in enumerate(axes.flatten()):
        ax.imshow(grid[i, :, :, 0], cmap='gray')
        ax.axis('off')
    plt.show()

# Display the generated images
show_images(fake_imgs)
```

Structure of the course

Knowledge Checks

- Confirmation that you're developing a working knowledge of GenAI topics



Question	Option A	Option B	Option C	Option D
What is the main advantage of Few-Shot Learning in LLMs?	Requires a large labeled dataset	Reduces the need for labeled data	Improves the model's interpretability	Requires extensive fine-tuning
How does self-supervised learning work?	It uses labeled data for predictions	It masks parts of the data as input/output pairs	It uses external labels for classification	It combines labeled and unlabeled dat
What problem does self-supervised learning address?	It reduces the need for labeled data	It addresses data scarcity in supervised learning	It simplifies the model training process	It ensures consistency in model predic
What is the role of autoregressive learning in LLMs?	Predicts the next token based on prior tokens	Enhances the model's output accuracy	Identifies relationships between labeled data	Increases the model's parameter coun
What is Few-Shot Learning in the context of GPT-3?	Training the model with many labeled examples	Providing a few examples before the main input	Building a single model for multiple tasks	Training the model on multiple dataset
How does in-context learning differ from fine-tuning?	In-context learning is task-specific	In-context learning uses the model's parameters flexibly	In-context learning is only used during training	In-context learning replaces fine-tunin
What is the purpose of prompt templating in LLMs?	To create unique responses	To format inputs consistently	To standardize outputs	To introduce randomness in outputs
Why is temperature an important factor in LLM outputs?	It controls the sequence length	It affects the randomness of outputs	It sets the learning rate	It determines the output length

Structure of the course

Lab Notebooks

- Based on the tutorial notebooks
- These will give you the chance to implement the tools and explore the topics covered in each module

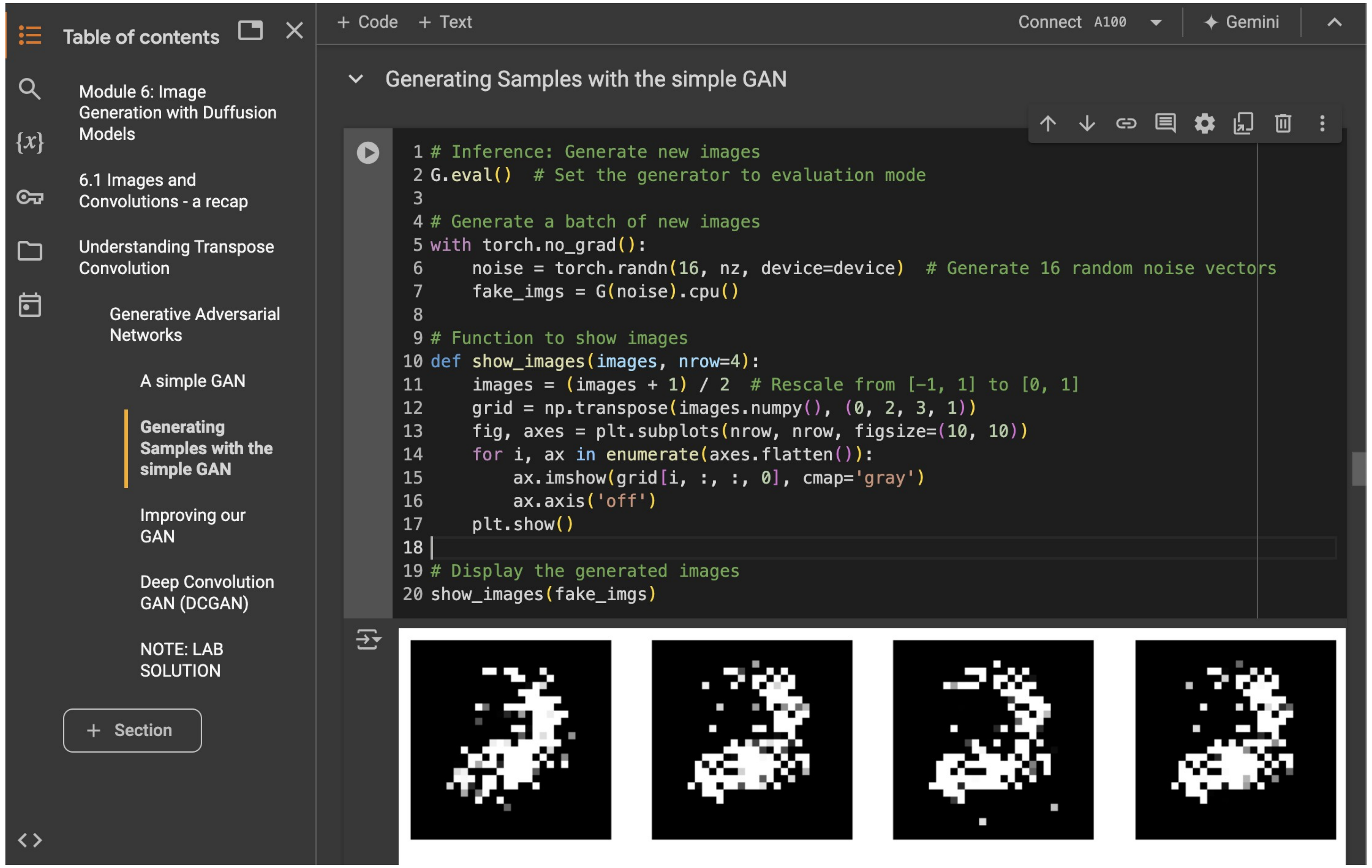


Table of contents
Module 6: Image Generation with Duffusion Models
6.1 Images and Convolutions - a recap
Understanding Transpose Convolution
Generative Adversarial Networks
A simple GAN
Generating Samples with the simple GAN
Improving our GAN
Deep Convolution GAN (DCGAN)
NOTE: LAB SOLUTION
+ Section
+ Code
+ Text
Connect A100
✦ Gemini

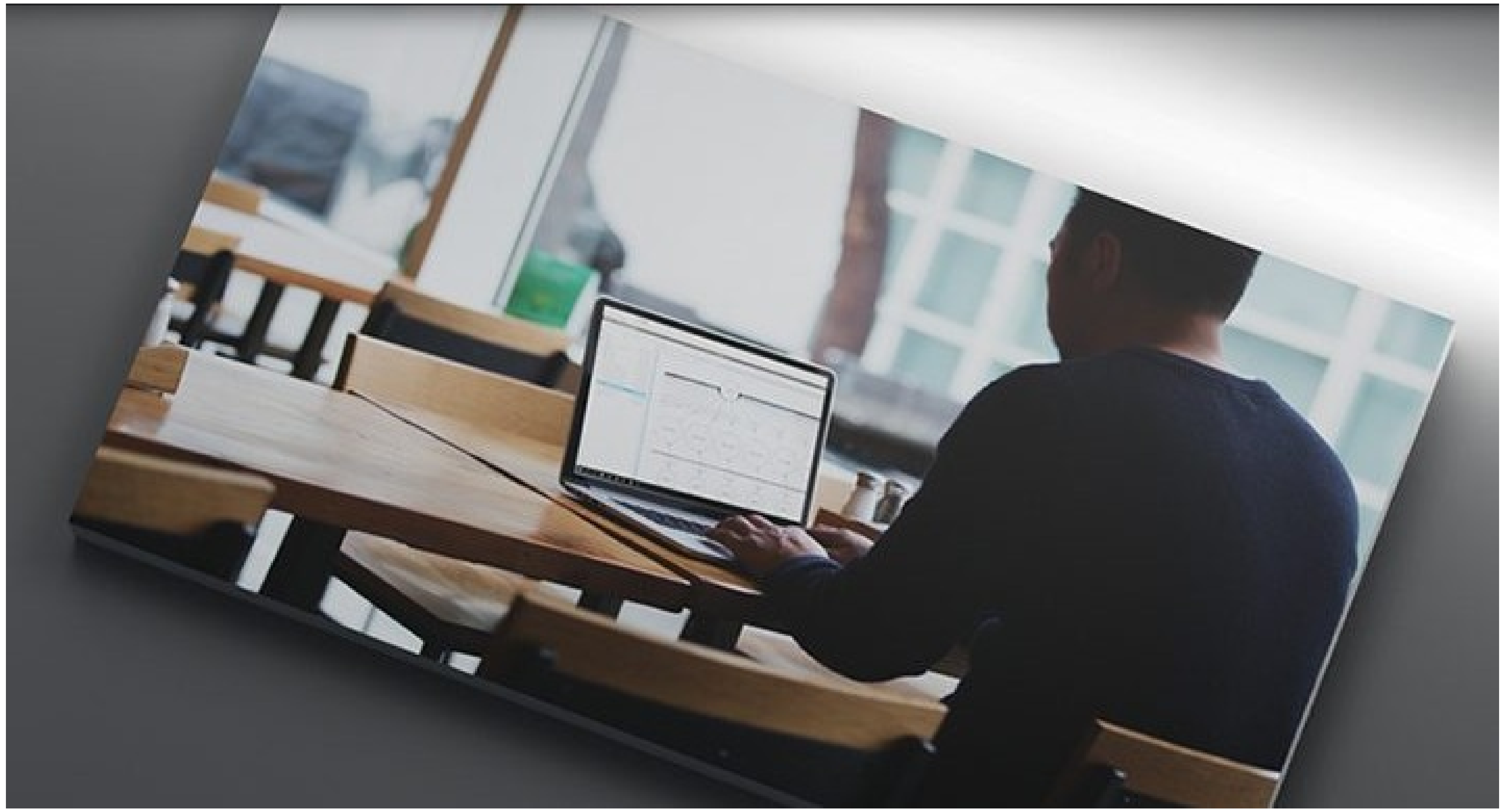
Generating Samples with the simple GAN

```python
# Inference: Generate new images
G.eval()  # Set the generator to evaluation mode

# Generate a batch of new images
with torch.no_grad():
    noise = torch.randn(16, nz, device=device)  # Generate 16 random noise vectors
    fake_imgs = G(noise).cpu()

# Function to show images
def show_images(images, nrow=4):
    images = (images + 1) / 2  # Rescale from [-1, 1] to [0, 1]
    grid = np.transpose(images.numpy(), (0, 2, 3, 1))
    fig, axes = plt.subplots(nrow, nrow, figsize=(10, 10))
    for i, ax in enumerate(axes.flatten()):
        ax.imshow(grid[i, :, :, 0], cmap='gray')
        ax.axis('off')
    plt.show()

# Display the generated images
show_images(fake_imgs)
```

Structure of the course

DLI Online Courses

- Related self-contained, self-paced DLI courses
- Free for students using promo codes provided
- Runs on DLI Platform instead of Colab
- Offers student certificates of competency based on built-in assessments



Tools used in GenAI

GenAI Tools in this Course

Python is the most widely used language in the AI/ML community, making it the go-to language for developing Generative AI models.
Python's simple syntax and readability allow developers and researchers to quickly prototype and implement complex algorithms.

- TensorFlow & PyTorch: Essential for building and training deep learning models, which are at the core of GenAI.
- Hugging Face Transformers: A popular library for working with large language models (LLMs), crucial for tasks like text generation and RAG.
- OpenCV & PIL: Important for image and video manipulation in GenAI applications like deep fakes and image generation

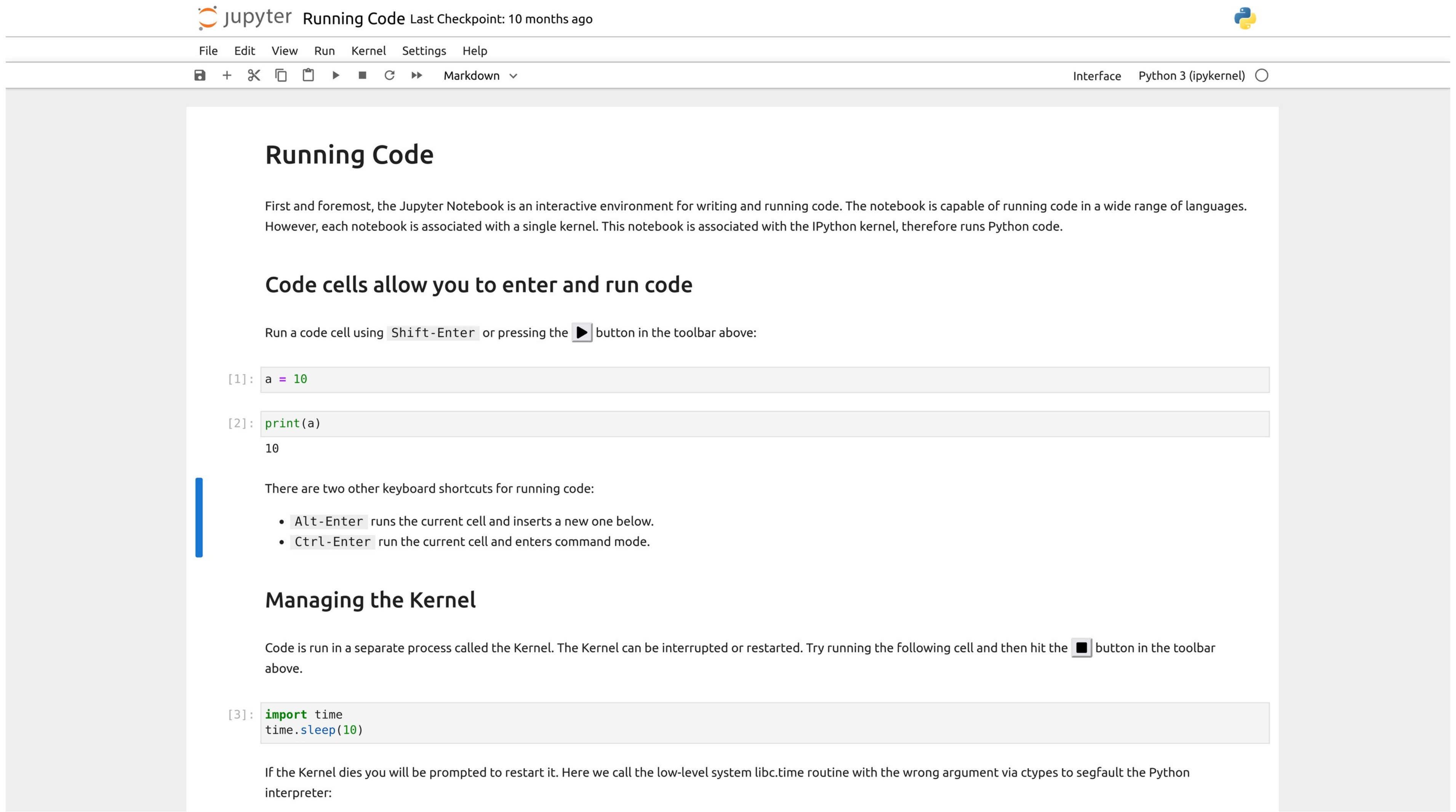
GenAI Tools in this Course

Coding Notebooks

The primary platform to work with these AI tools and code libraries will be through the Google Colab notebooks

These notebooks allow for easy interaction with the data and models in a visual manner

Platforms like Google Colab and other cloud notebook providers also supply access to cloud-based GPUs and compute infrastructure.



Running Code

First and foremost, the Jupyter Notebook is an interactive environment for writing and running code. The notebook is capable of running code in a wide range of languages. However, each notebook is associated with a single kernel. This notebook is associated with the IPython kernel, therefore runs Python code.

Code cells allow you to enter and run code

Run a code cell using `Shift-Enter` or pressing the ▶ button in the toolbar above:

```
[1]: a = 10
```

```
[2]: print(a)
     10
```

There are two other keyboard shortcuts for running code:

- `Alt-Enter` runs the current cell and inserts a new one below.
- `Ctrl-Enter` run the current cell and enters command mode.

Managing the Kernel

Code is run in a separate process called the Kernel. The Kernel can be interrupted or restarted. Try running the following cell and then hit the ■ button in the toolbar above.

```
[3]: import time
     time.sleep(10)
```

If the Kernel dies you will be prompted to restart it. Here we call the low-level system libc.time routine with the wrong argument via ctypes to segfault the Python interpreter:

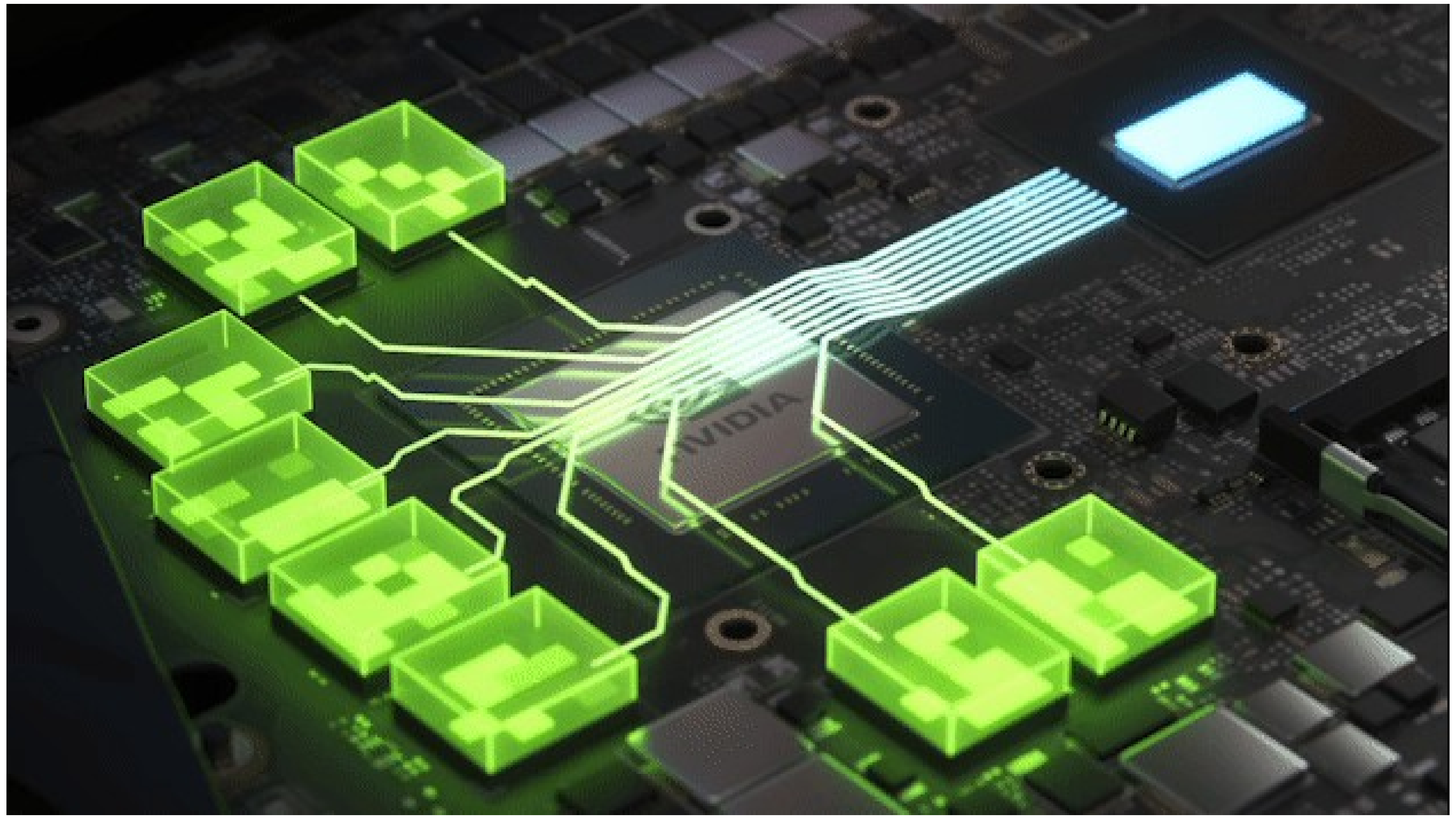
GenAI Tools in this Course

The Importance of GPUs in GenAI

Parallelism: GPUs have thousands of cores optimized for parallel processing, allowing them to handle many operations at once. This is essential for the matrix multiplications and other computations involved in AI inference.

Throughput: GPUs can process large batches of data simultaneously, which is critical for applications like image generation or video synthesis where high throughput is needed.

Latency: GPUs reduce latency by executing multiple operations in parallel, which speeds up the time it takes to get from input to output in GenAI models.



Course Content

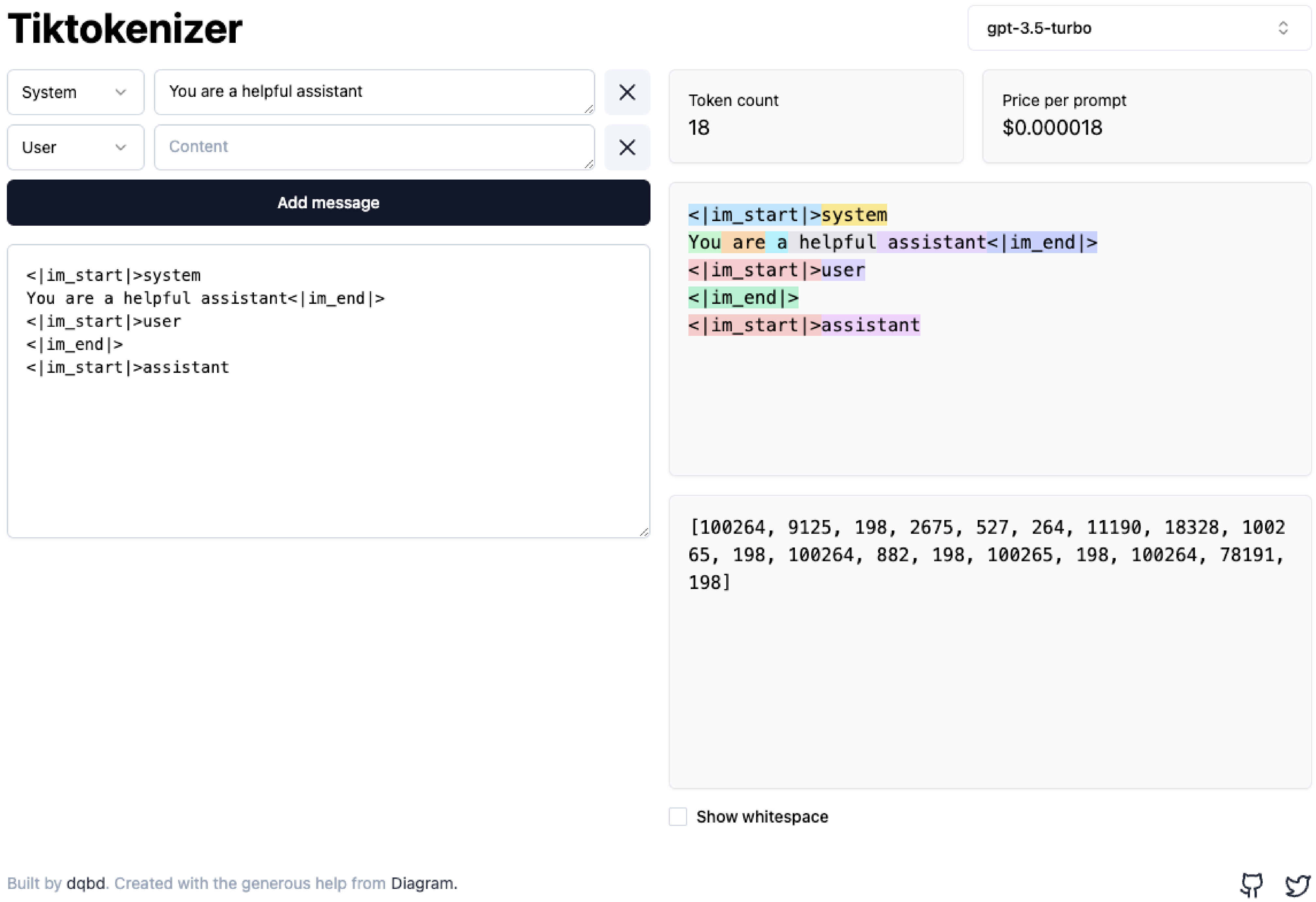
What this course will cover

Modules

1. Introduction to Generative AI
2. Word Embeddings, Tokens, and NLP
3. Large Language Models and the Transformer
4. LLM Scaling Laws and LLM Families
5. Multimodal Learning and its Applications
6. Diffusion Models in Generative AI
7. Model Training (Pre-Training, Instruction Following, and PEFT)
8. LLM Orchestration
9. Scaling Model Training to Distributed Workloads

Full online syllabus

2. Word Embeddings, Tokens, and NLP



Tiktokenizer

gpt-3.5-turbo

System

You are a helpful assistant

User

Content

Add message

```
<|im_start|>system
You are a helpful assistant<|im_end|>
<|im_start|>user
<|im_end|>
<|im_start|>assistant
```

Token count
18

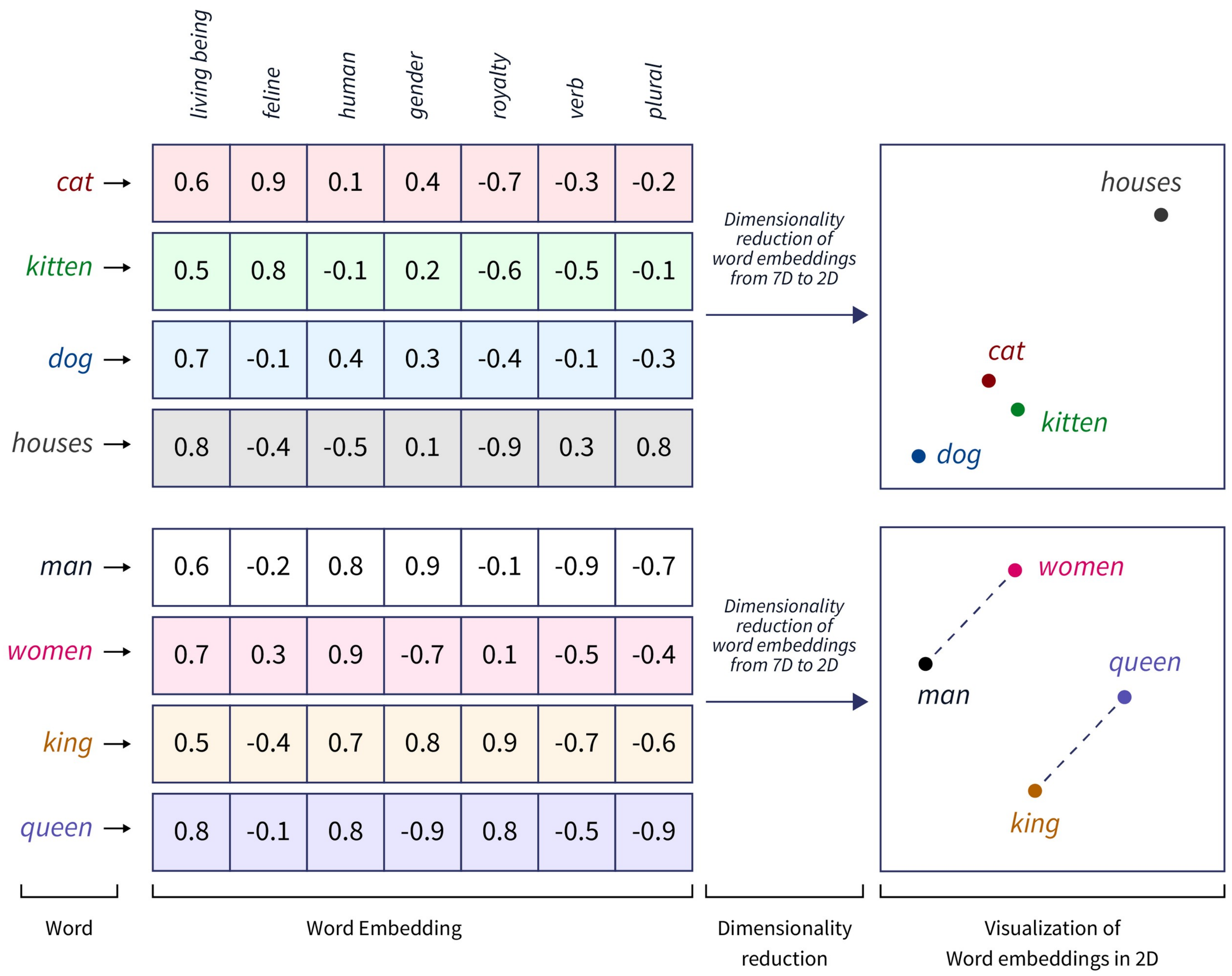
Price per prompt
$0.000018

```
<|im_start|>system
You are a helpful assistant<|im_end|>
<|im_start|>user
<|im_end|>
<|im_start|>assistant
```

```
[100264, 9125, 198, 2675, 527, 264, 11190, 18328, 100265, 198, 100264, 882, 198, 100265, 198, 100264, 78191, 198]
```

□ Show whitespace

Built by dqbd. Created with the generous help from Diagram.

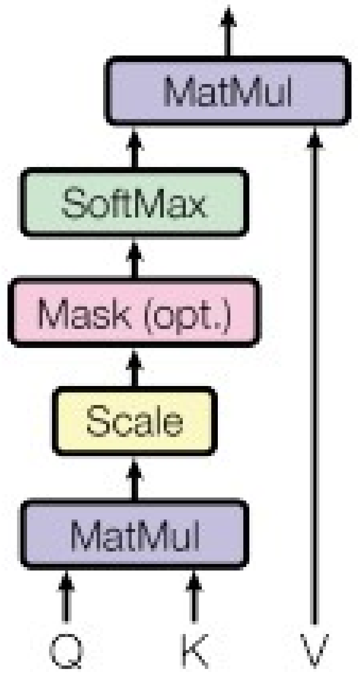


Word	living being	feline	human	gender	royalty	verb	plural
cat	0.6	0.9	0.1	0.4	-0.7	-0.3	-0.2
kitten	0.5	0.8	-0.1	0.2	-0.6	-0.5	-0.1
dog	0.7	-0.1	0.4	0.3	-0.4	-0.1	-0.3
houses	0.8	-0.4	-0.5	0.1	-0.9	0.3	0.8
man	0.6	-0.2	0.8	0.9	-0.1	-0.9	-0.7
women	0.7	0.3	0.9	-0.7	0.1	-0.5	-0.4
king	0.5	-0.4	0.7	0.8	0.9	-0.7	-0.6
queen	0.8	-0.1	0.8	-0.9	0.8	-0.5	-0.9

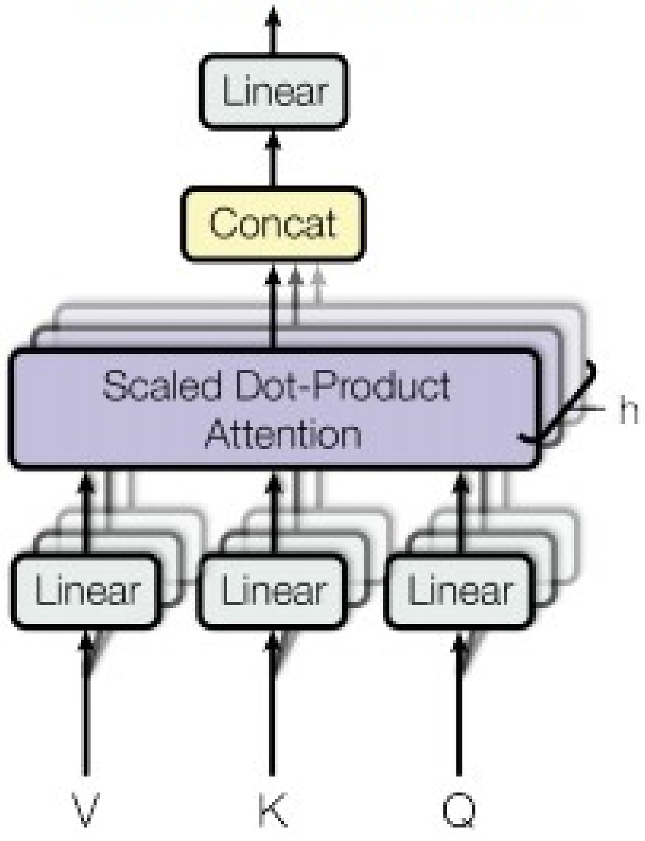
Dimensionality reduction of word embeddings from 7D to 2D
houses
cat
kitten
dog
Dimensionality reduction of word embeddings from 7D to 2D
women
man
queen
king
Word
Word Embedding
Dimensionality reduction
Visualization of Word embeddings in 2D

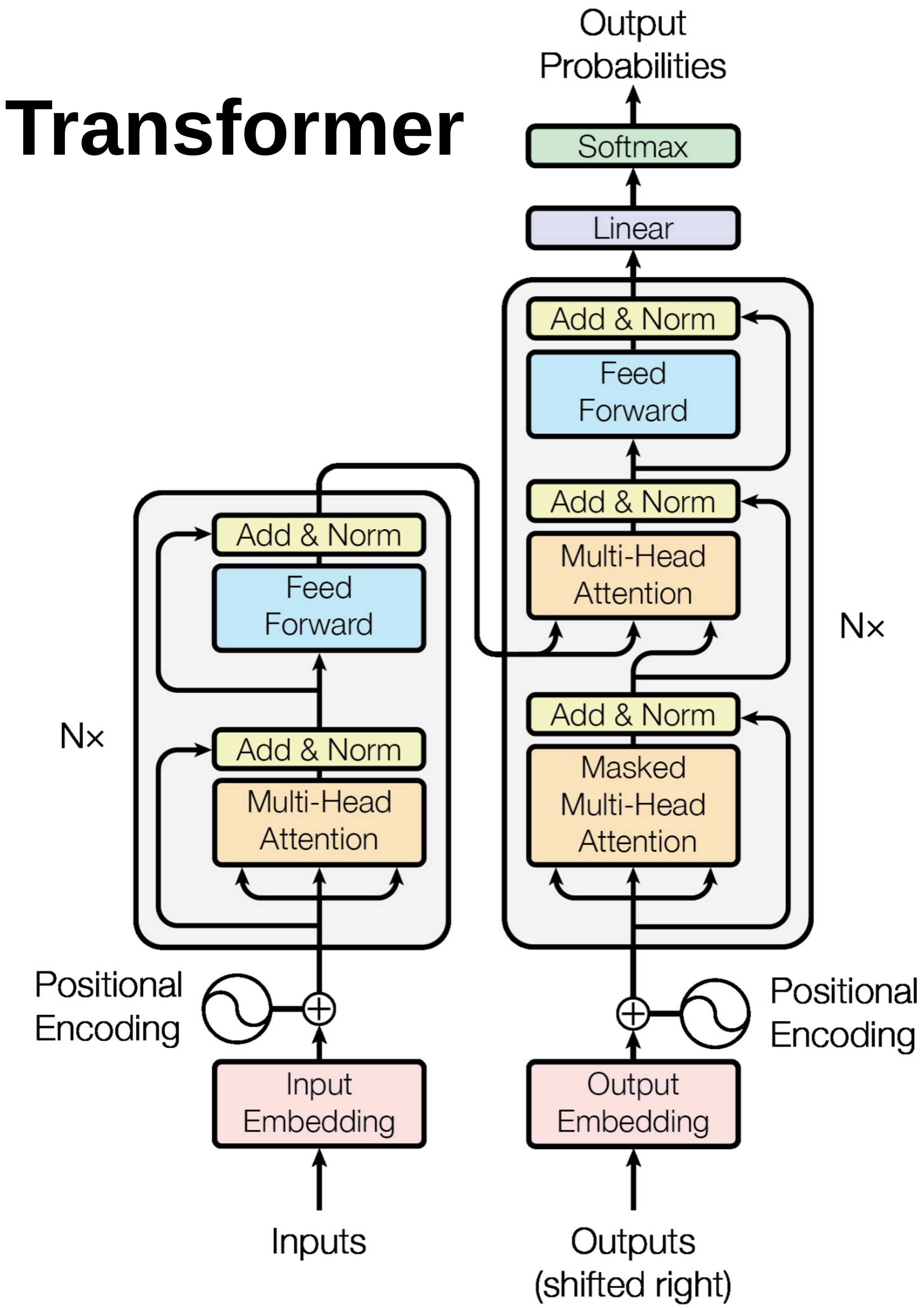
3. Large Language Models and the Transformer

Scaled Dot-Product Attention

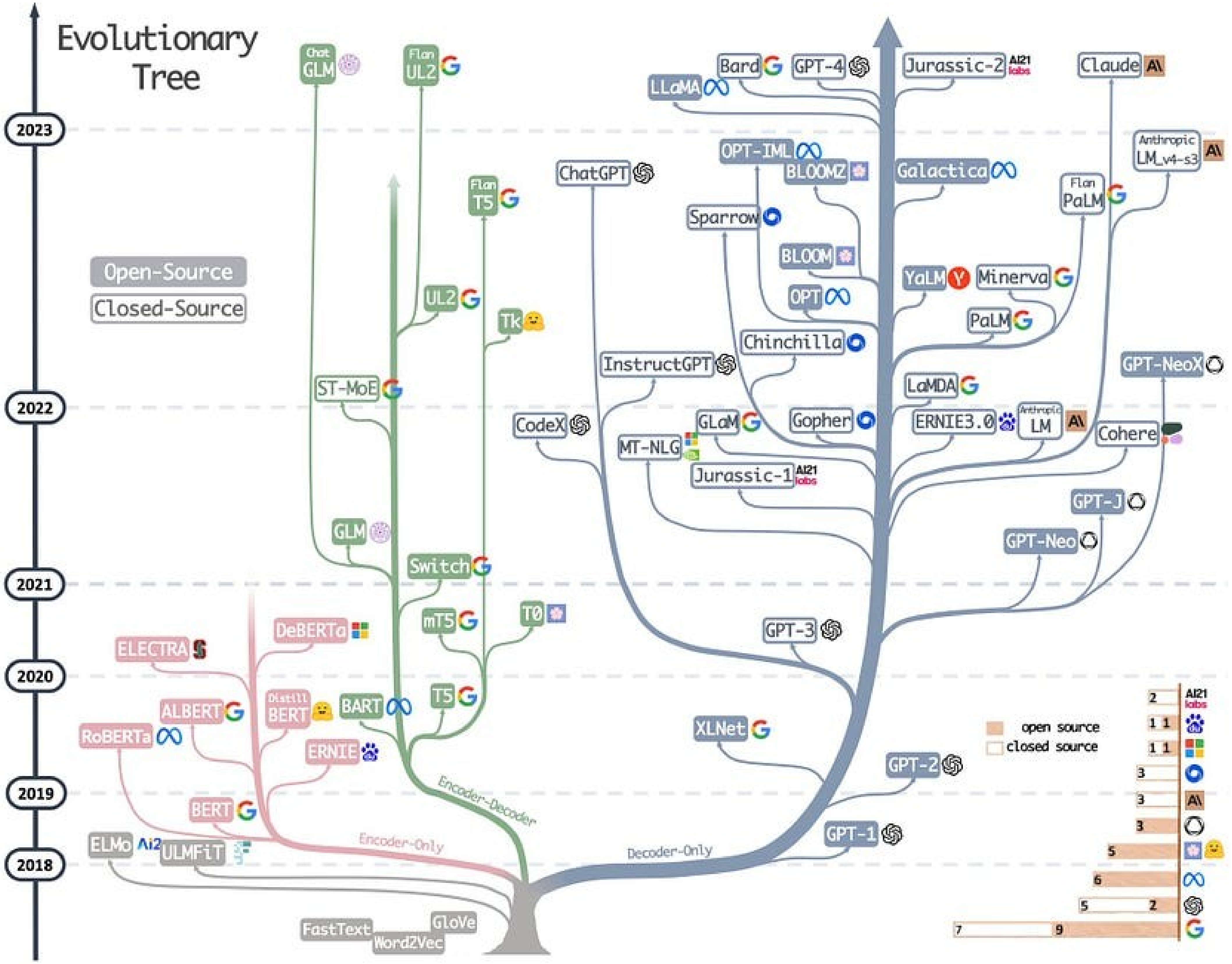


Multi-Head Attention

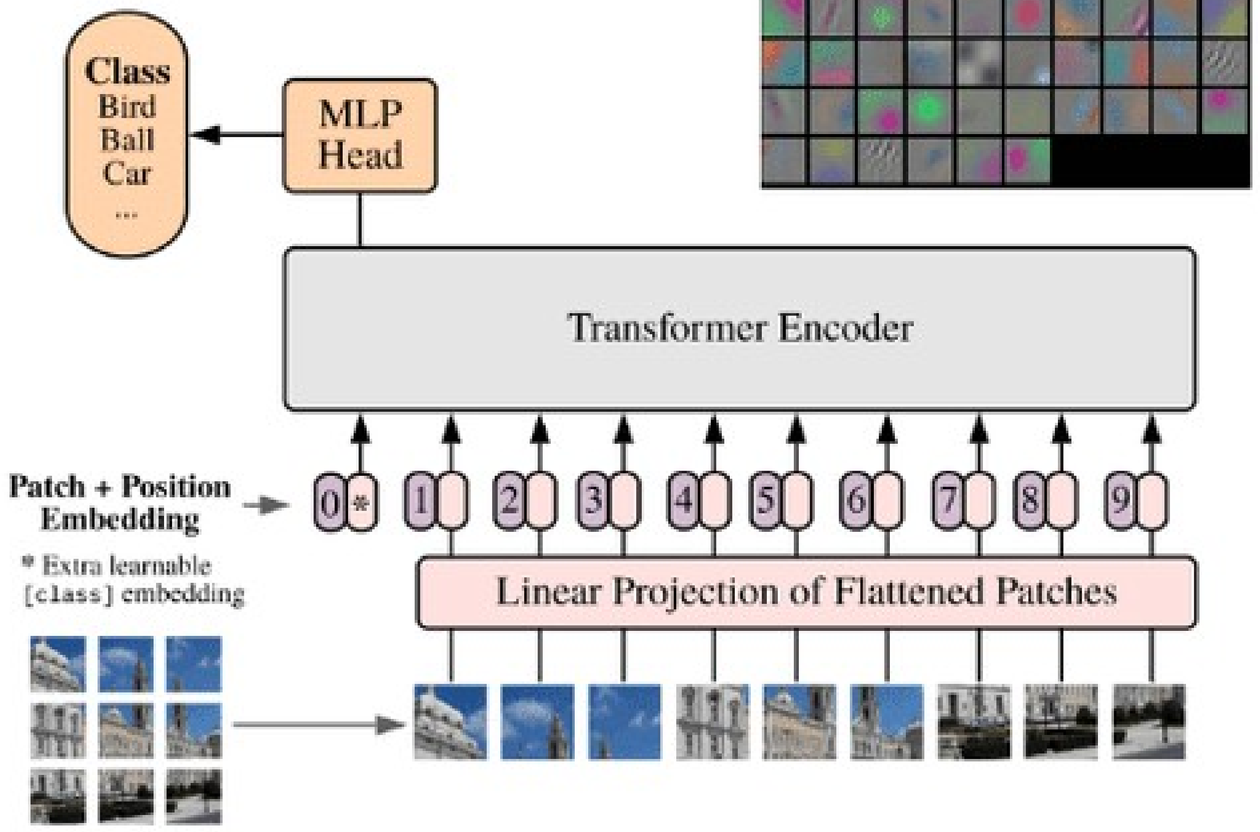




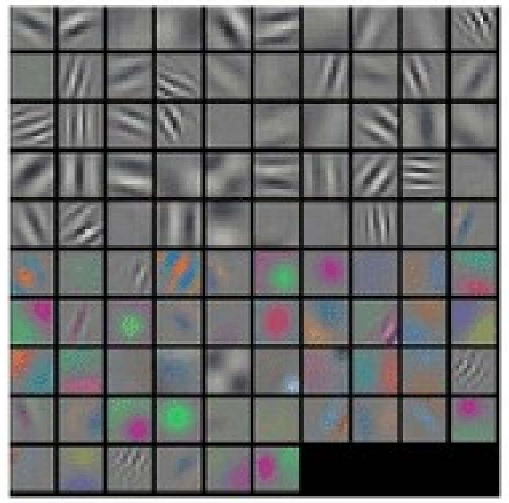
4. LLM Scaling Laws and LLM Families



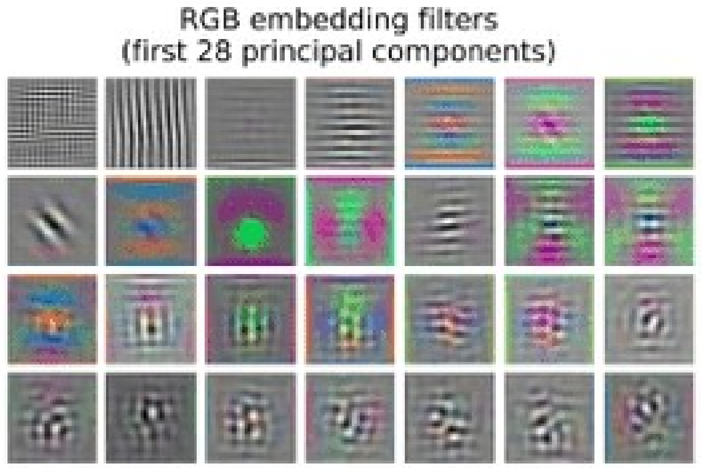
5. Multimodal Learning and its Applications

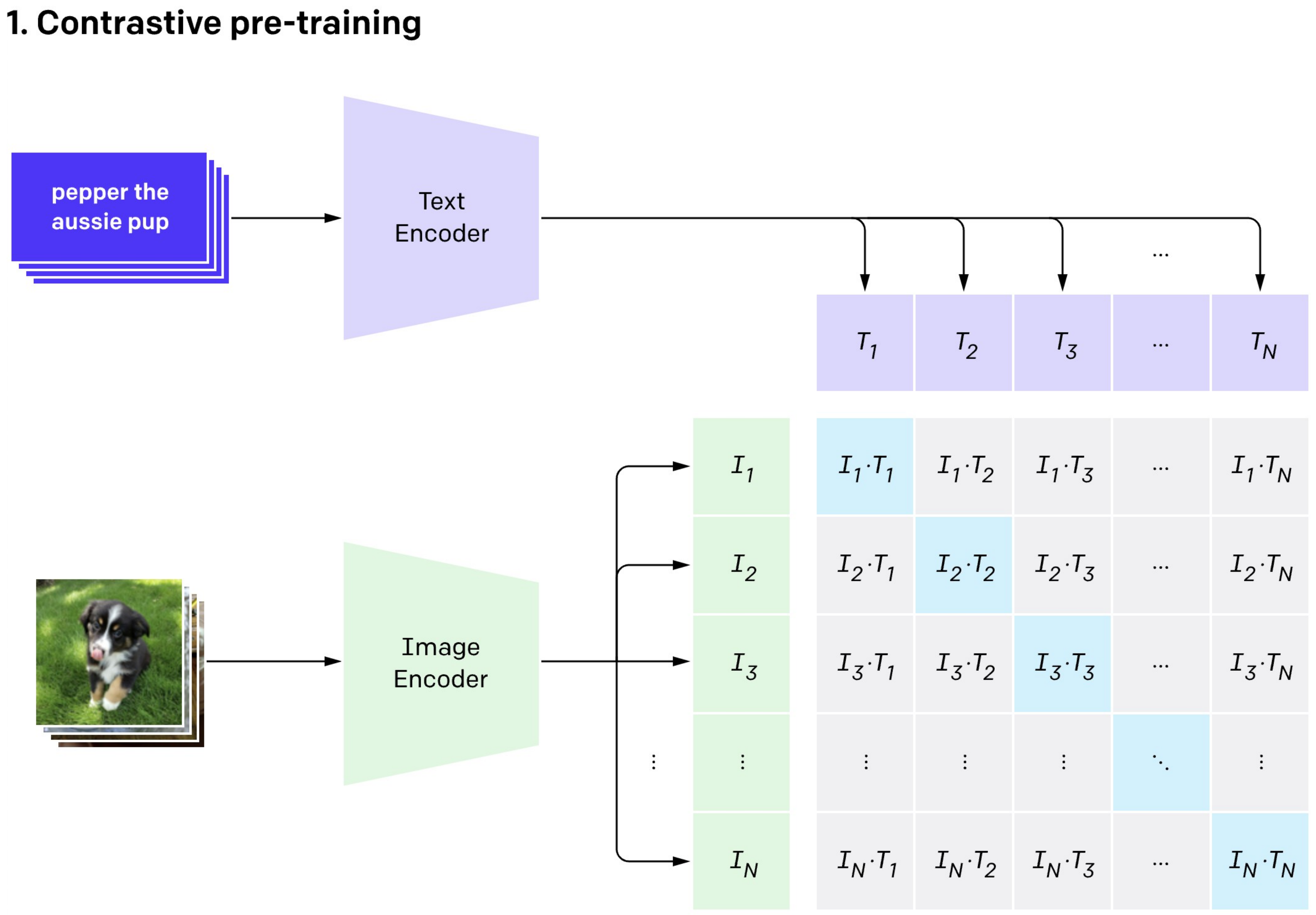

Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0 *
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches

Alexnet 1st conv filters



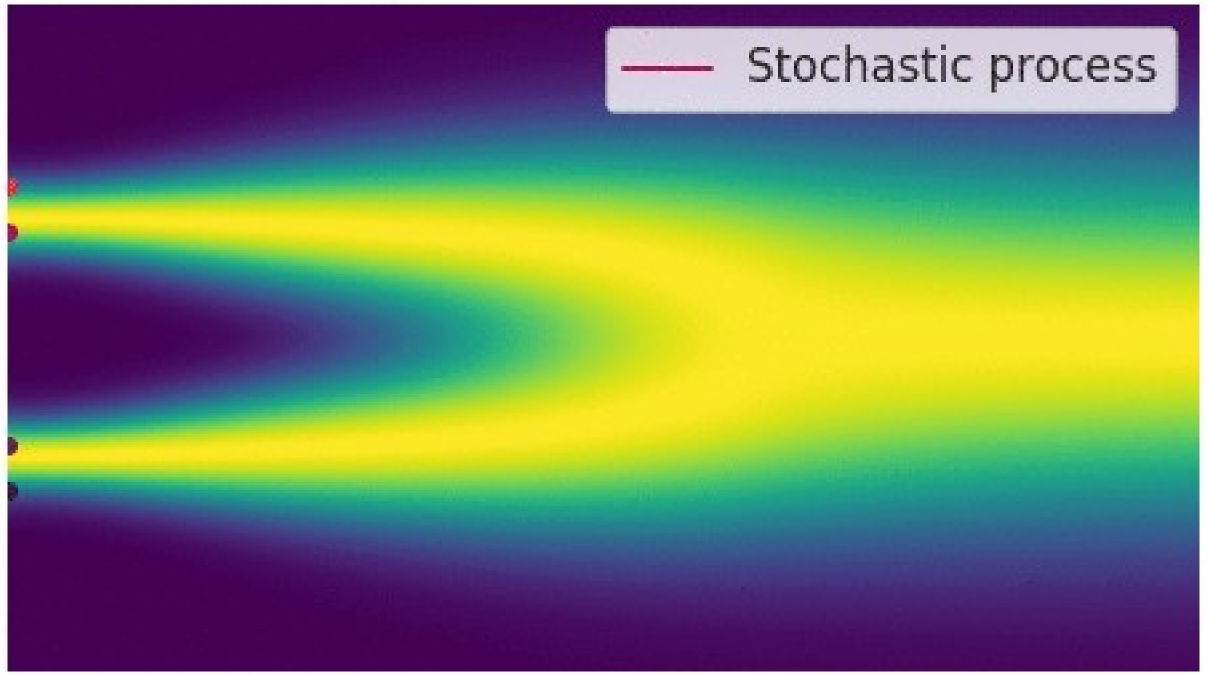
ViT 1st linear embedding filters

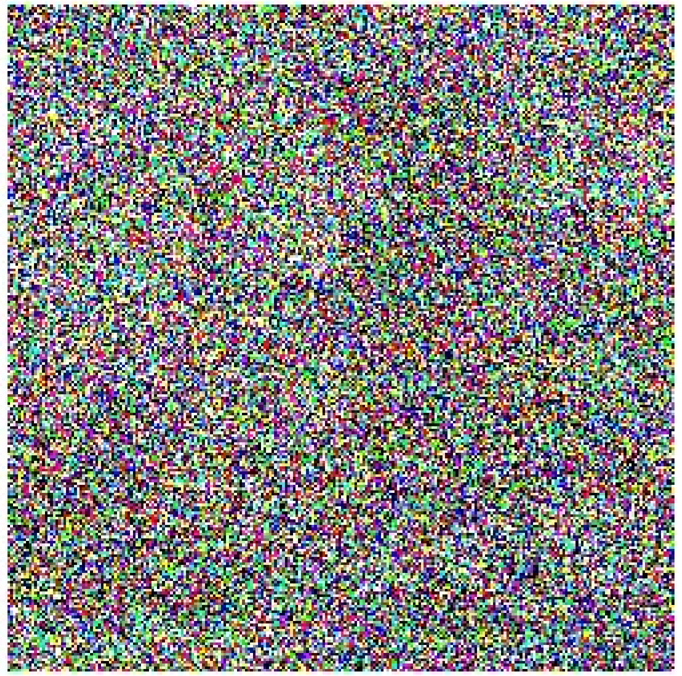


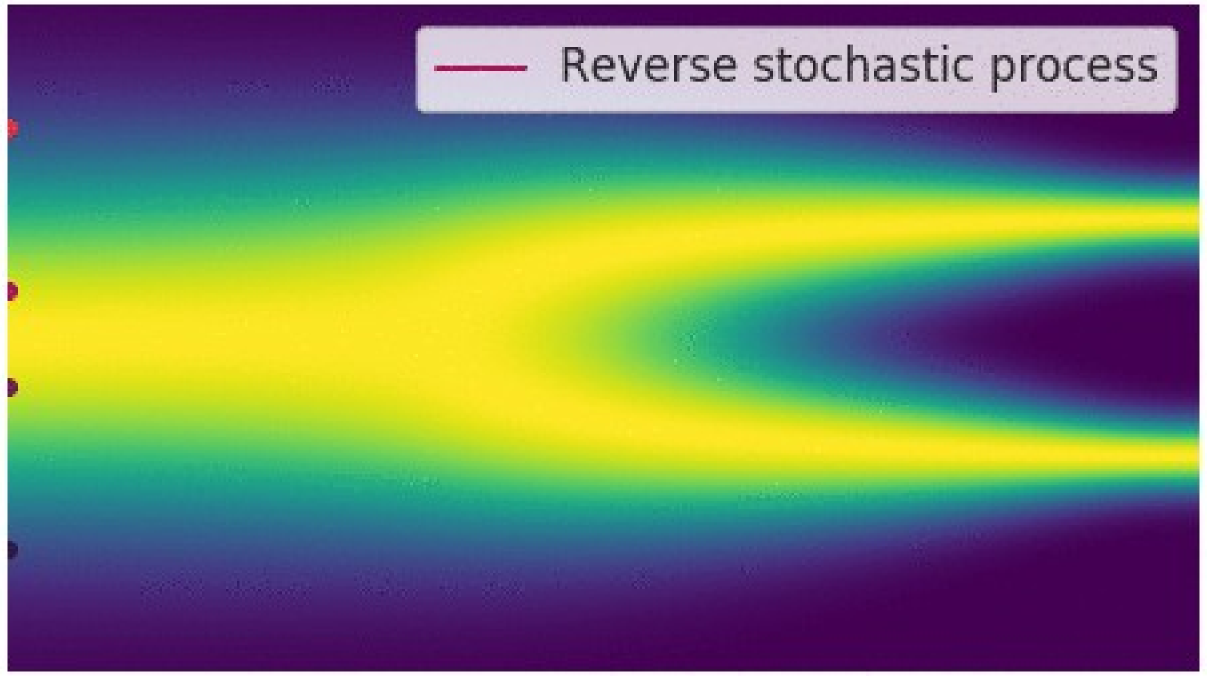


	T_1	T_2	T_3	...	T_N
I_1	$I_1 \cdot T_1$	$I_1 \cdot T_2$	$I_1 \cdot T_3$	...	$I_1 \cdot T_N$
I_2	$I_2 \cdot T_1$	$I_2 \cdot T_2$	$I_2 \cdot T_3$	...	$I_2 \cdot T_N$
I_3	$I_3 \cdot T_1$	$I_3 \cdot T_2$	$I_3 \cdot T_3$	...	$I_3 \cdot T_N$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
I_N	$I_N \cdot T_1$	$I_N \cdot T_2$	$I_N \cdot T_3$	...	$I_N \cdot T_N$

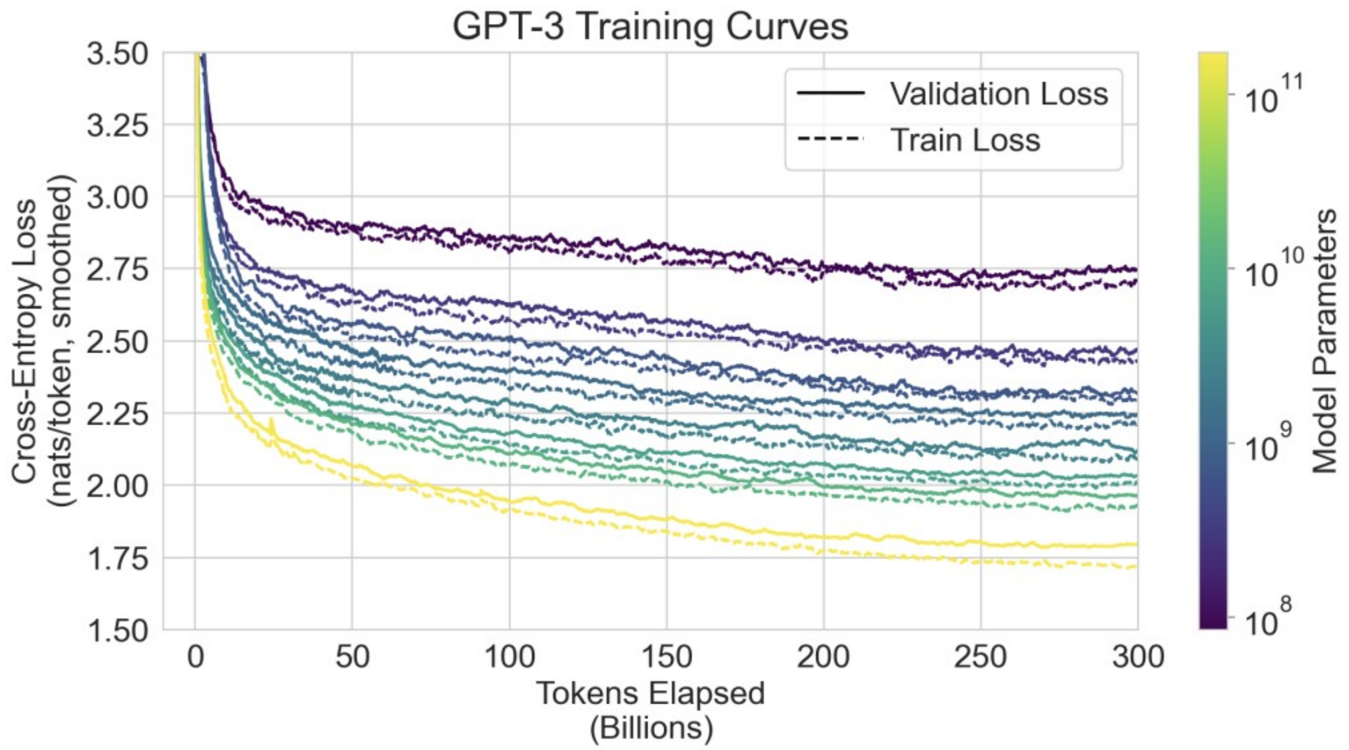
6. Diffusion Models in Generative AI

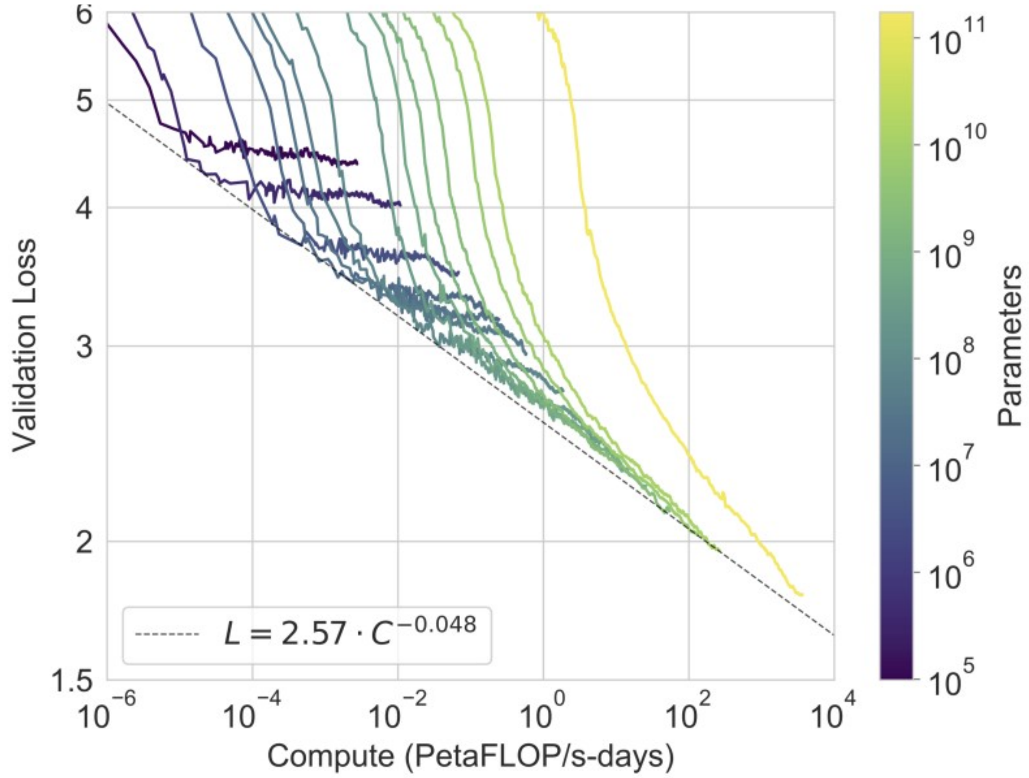


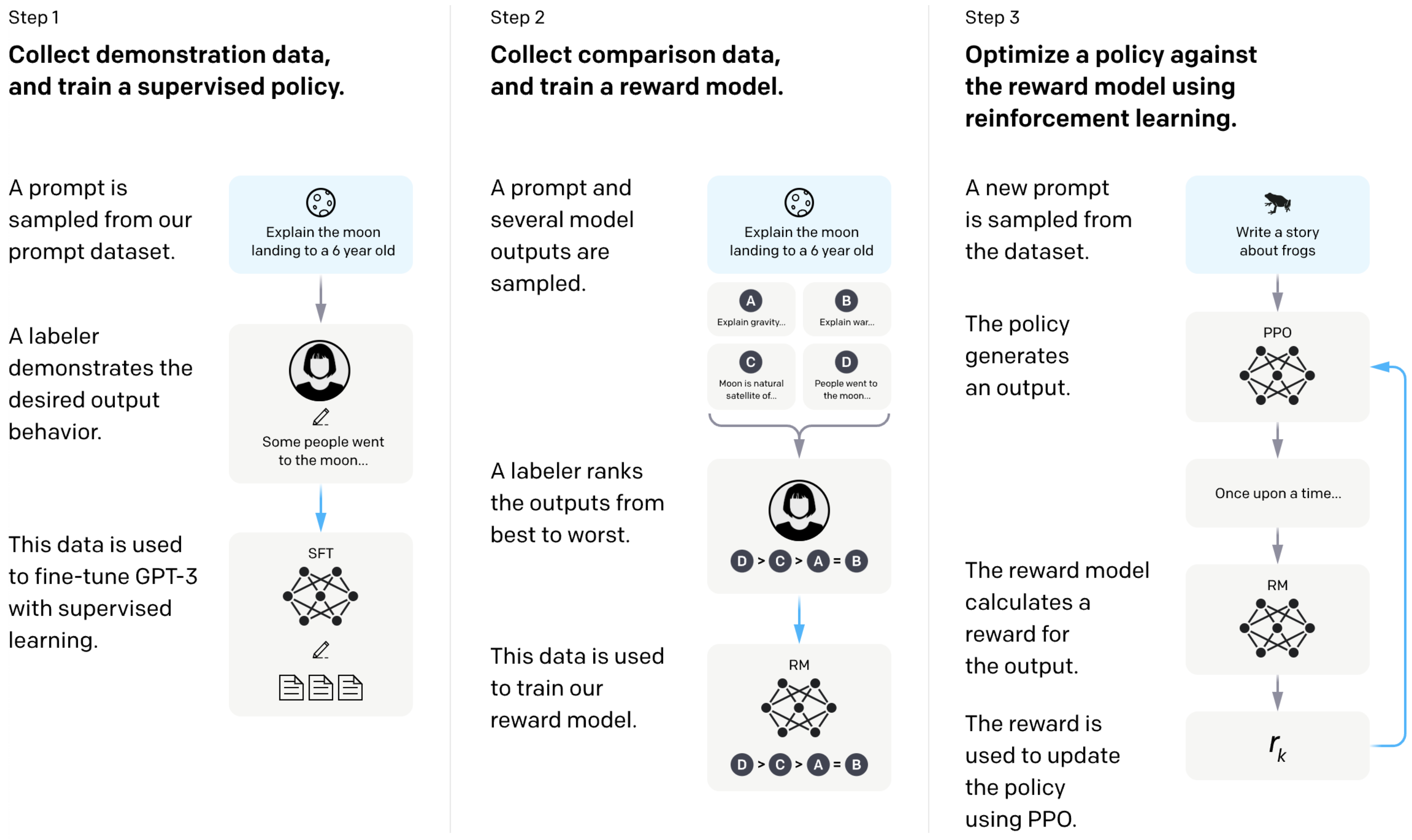




7. Model Training (Pre-Training, Instruction Following, and PEFT)







Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.

Explain the moon landing to a 6 year old
Some people went to the moon...
SFT

Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.

Explain the moon landing to a 6 year old
A
Explain gravity...
B
Explain war...
C
Moon is natural satellite of...
D
People went to the moon...
D > C > A = B
RM
D > C > A = B

Step 3

Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

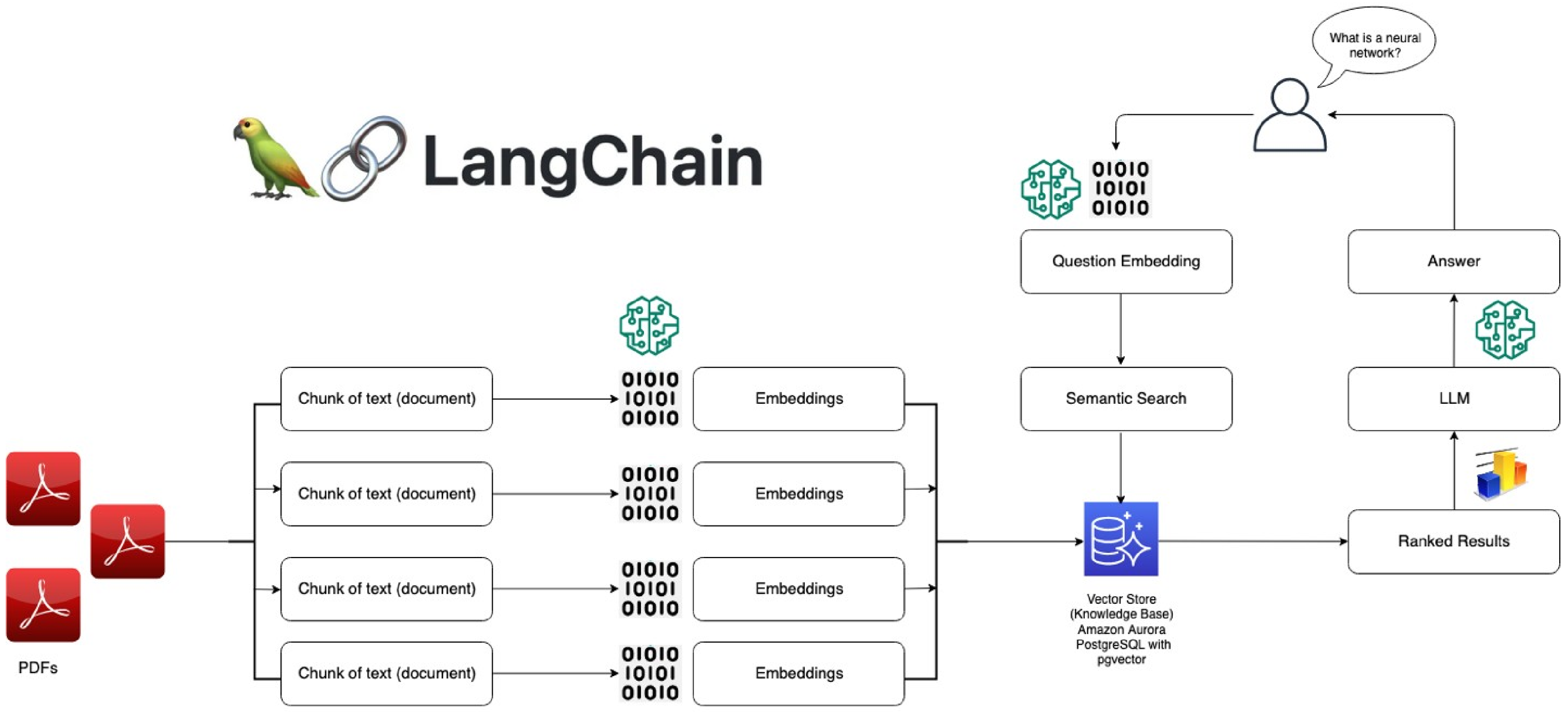
The policy generates an output.

The reward model calculates a reward for the output.

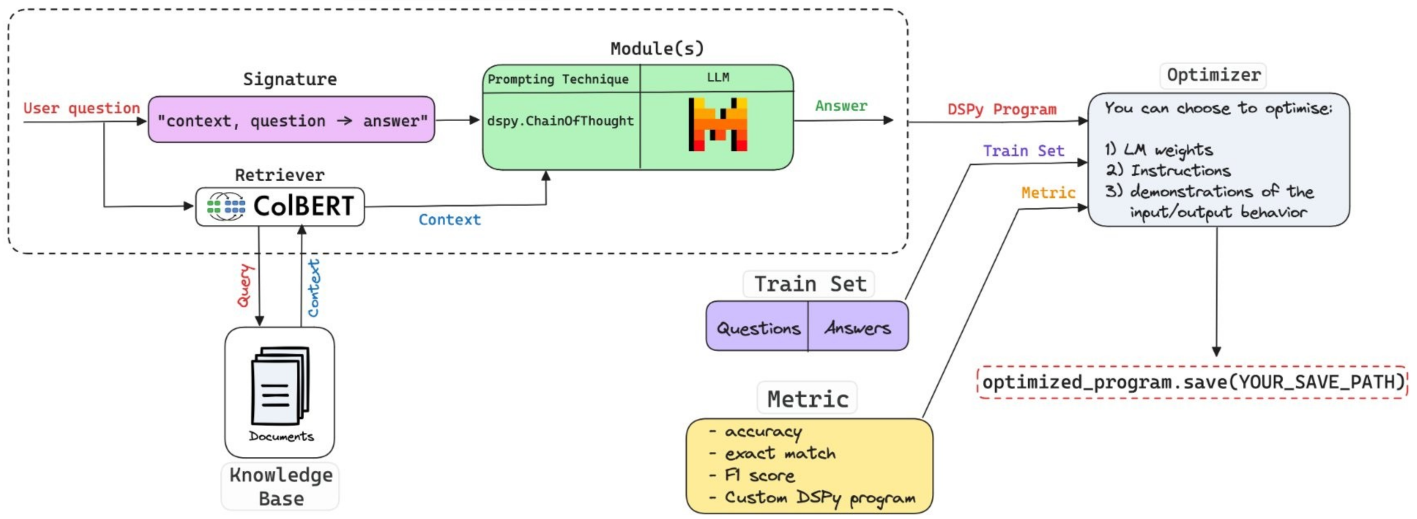
The reward is used to update the policy using PPO.

8. LLM Orchestration

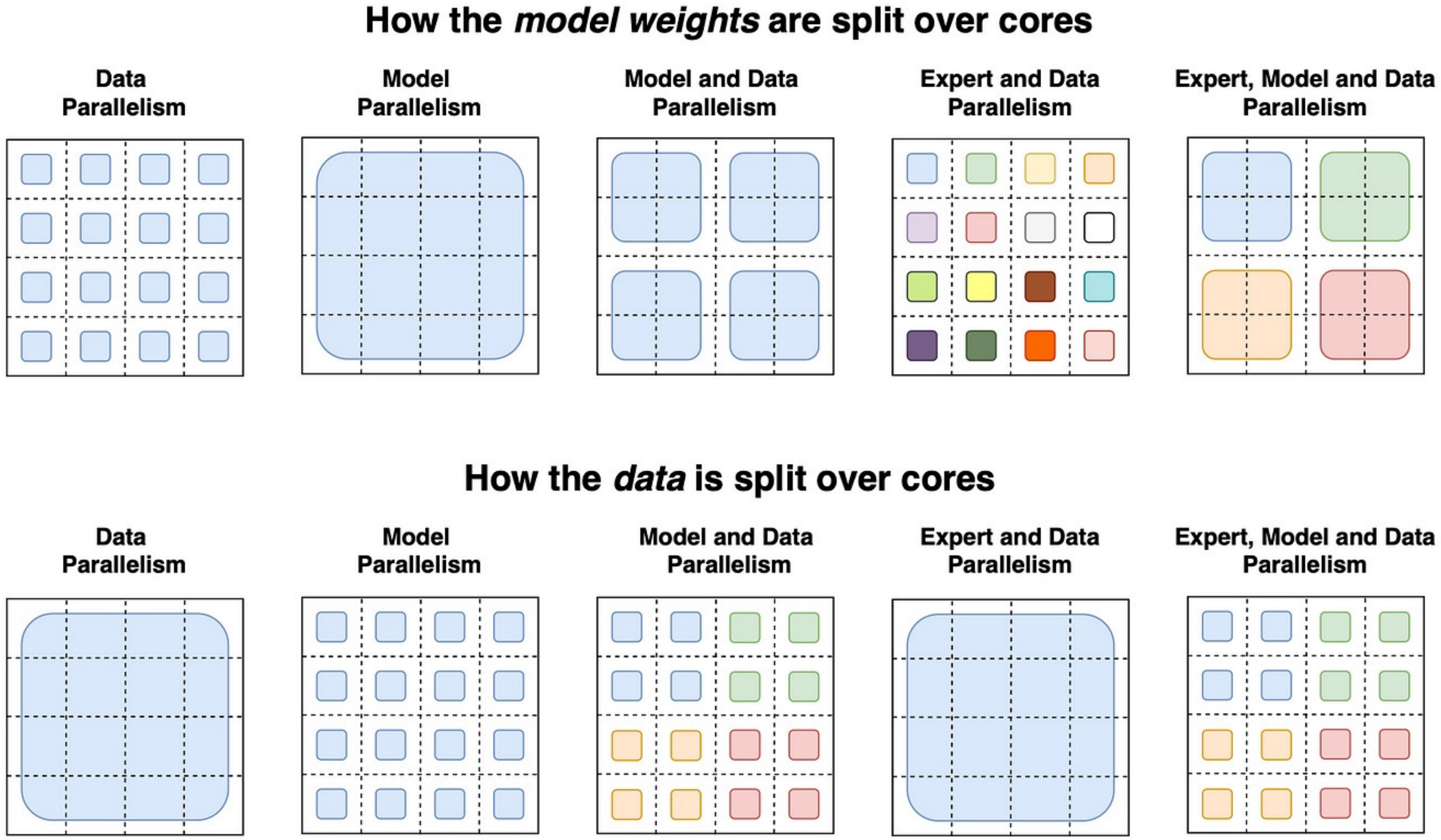
LangChain



DSPy



9. Scaling Model Training to Distributed Workloads


How the *model weights* are split over cores
Data Parallelism
Model Parallelism
Model and Data Parallelism
Expert and Data Parallelism
Expert, Model and Data Parallelism
How the *data* is split over cores
Data Parallelism
Model Parallelism
Model and Data Parallelism
Expert and Data Parallelism
Expert, Model and Data Parallelism

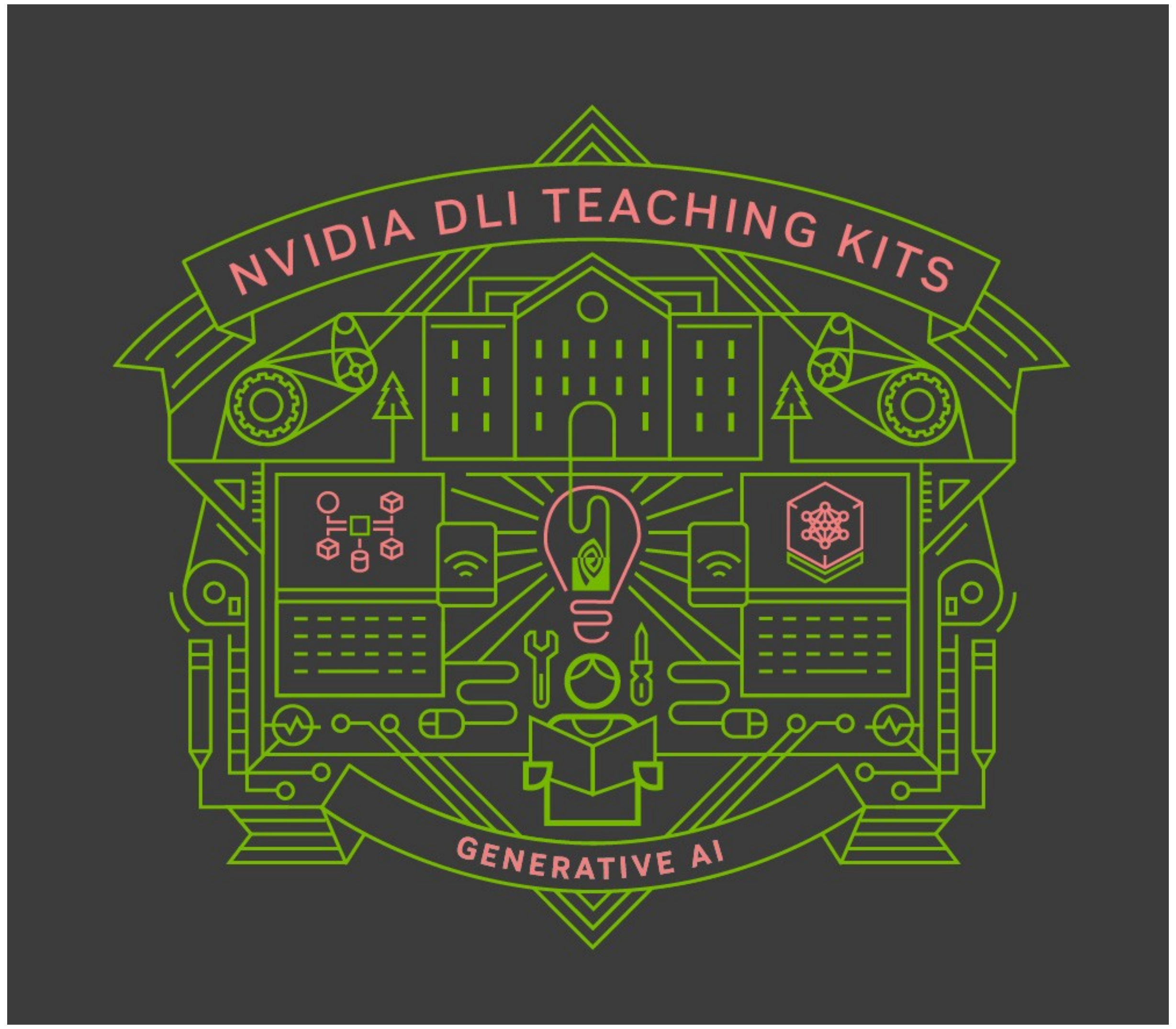
Wrap Up

Introduction to GenAI

- Today we introduced the GenAI Course
- We covered the different materials used in the class
- Provided an overview of the modules to come
- Highlighted how GenAI has exploded onto the scene of science, engineering, and the economy at large

----

In the next lesson we will start our discussion into how we got to this point and what advances led us from the digital computer, to ChatGPT



NVIDIA

DARTMOUTH
ENGINEERING
1867

Thank you!