



DEEP  
LEARNING  
INSTITUTE

# SELF-SUPERVISION, BERT, AND BEYOND

Building Transformer-Based Natural Language Processing Applications  
(Part 2)





# FULL COURSE AGENDA

## Part 1: Machine Learning in NLP

Lecture: NLP background and the role of DNNs leading to the Transformer architecture

Lab: Tutorial-style exploration of a *translation task* using the Transformer architecture

---

## Part 2: Self-Supervision, BERT, and Beyond

Lecture: Discussion of how language models with self-supervision have moved beyond the basic Transformer to BERT and ever larger models

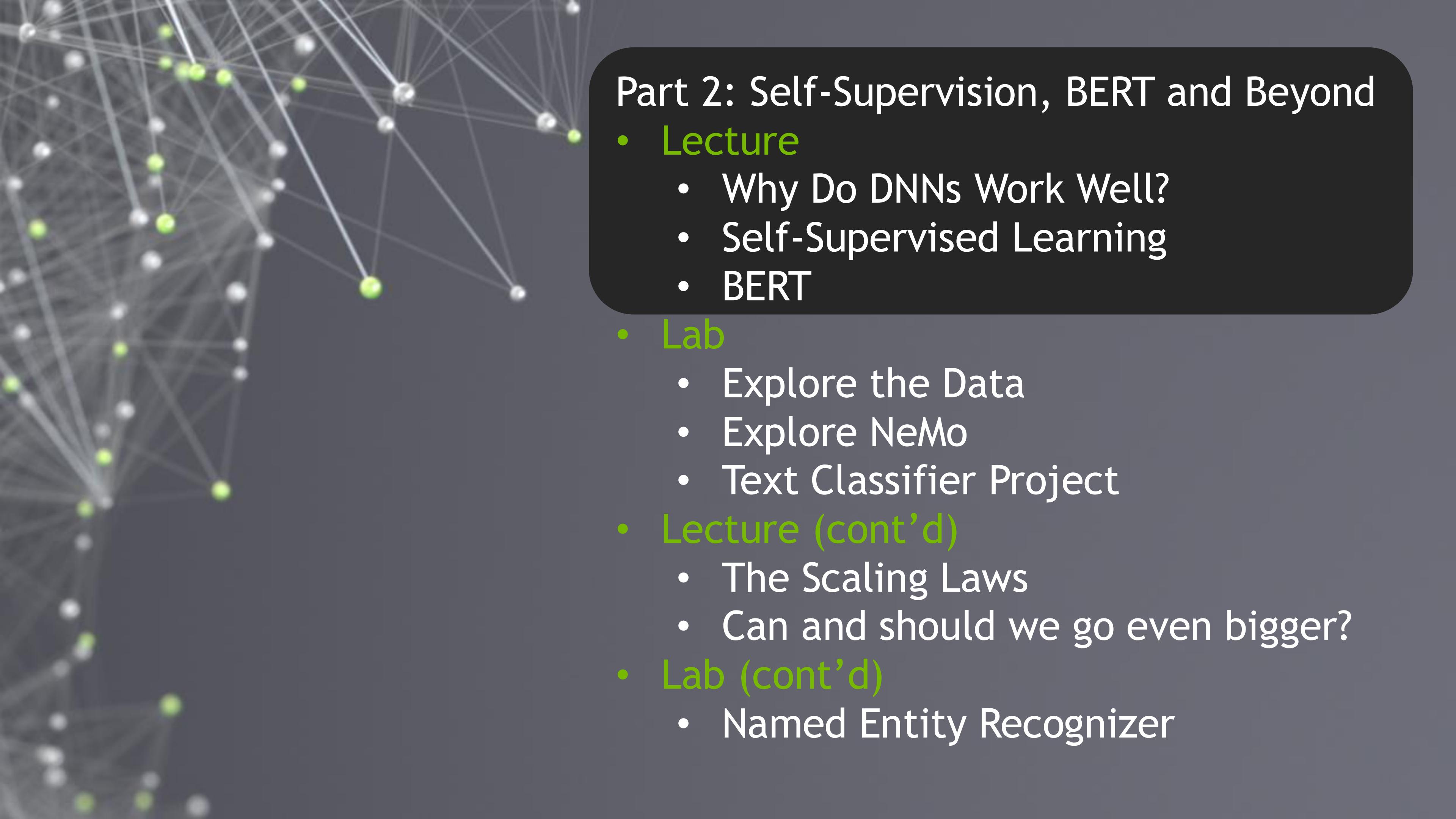
Lab: Practical hands-on guide to the NVIDIA NeMo API and exercises to build a *text classification task* and a *named entity recognition task* using BERT-based language models

---

## Part 3: Production Deployment

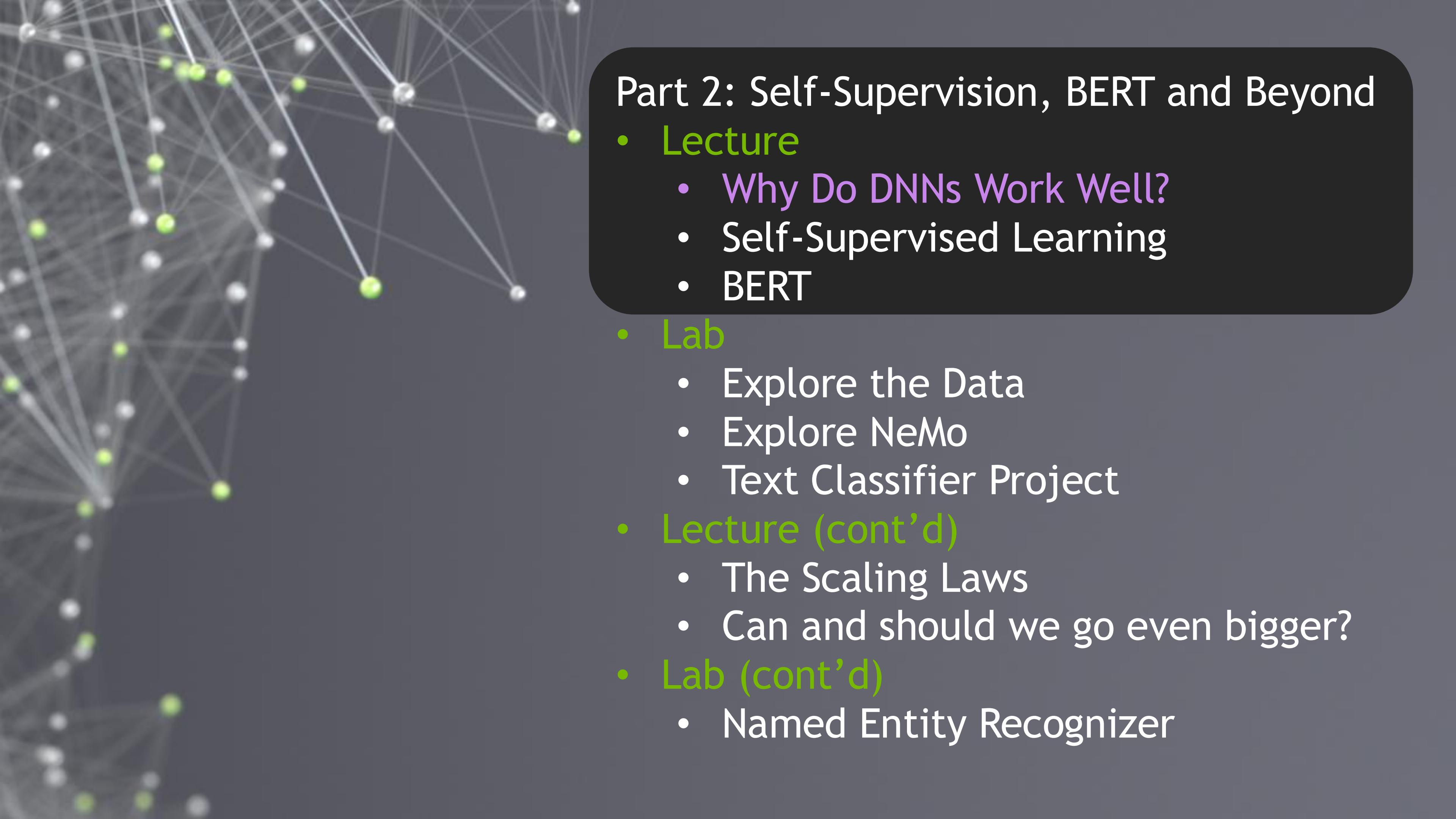
Lecture: Discussion of production deployment considerations and NVIDIA Triton Inference Server

Lab: Hands-on deployment of an example *question answering task* to NVIDIA Triton



## Part 2: Self-Supervision, BERT and Beyond

- **Lecture**
  - Why Do DNNs Work Well?
  - Self-Supervised Learning
  - BERT
- **Lab**
  - Explore the Data
  - Explore NeMo
  - Text Classifier Project
- **Lecture (cont'd)**
  - The Scaling Laws
  - Can and should we go even bigger?
- **Lab (cont'd)**
  - Named Entity Recognizer



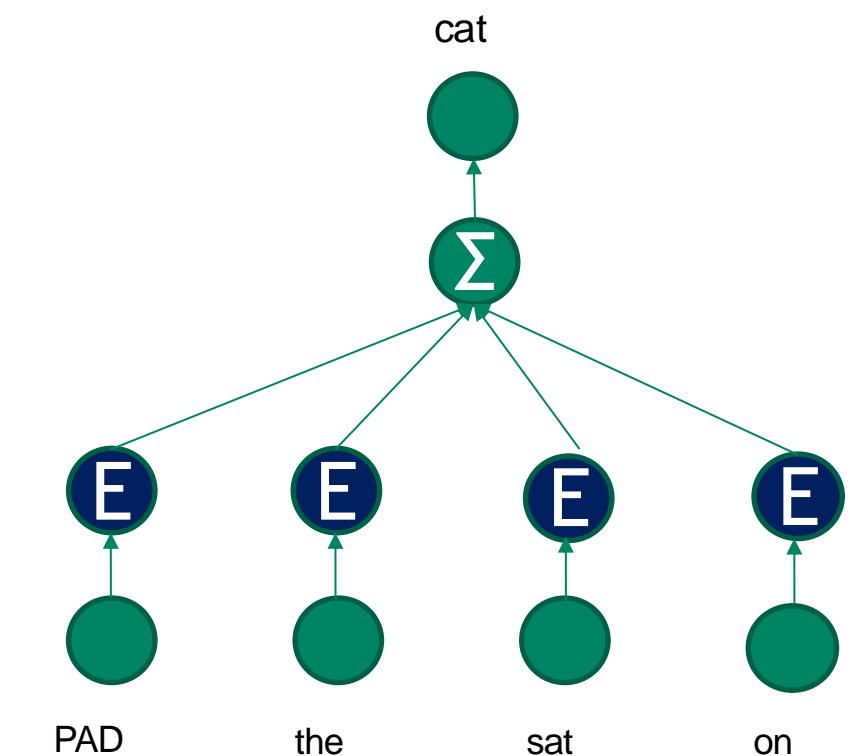
## Part 2: Self-Supervision, BERT and Beyond

- **Lecture**
  - Why Do DNNs Work Well?
  - Self-Supervised Learning
  - BERT
- **Lab**
  - Explore the Data
  - Explore NeMo
  - Text Classifier Project
- **Lecture (cont'd)**
  - The Scaling Laws
  - Can and should we go even bigger?
- **Lab (cont'd)**
  - Named Entity Recognizer

# COMPUTE

## Or lack of thereof

In this section, we propose two new model architectures for learning distributed representations of words that try to minimize computational complexity. The main observation from the previous section was that most of the complexity is caused by the non-linear hidden layer in the model. While this is what makes neural networks so attractive, we decided to explore simpler models that might not be able to represent the data as precisely as neural networks, but can possibly be trained on much more data efficiently.

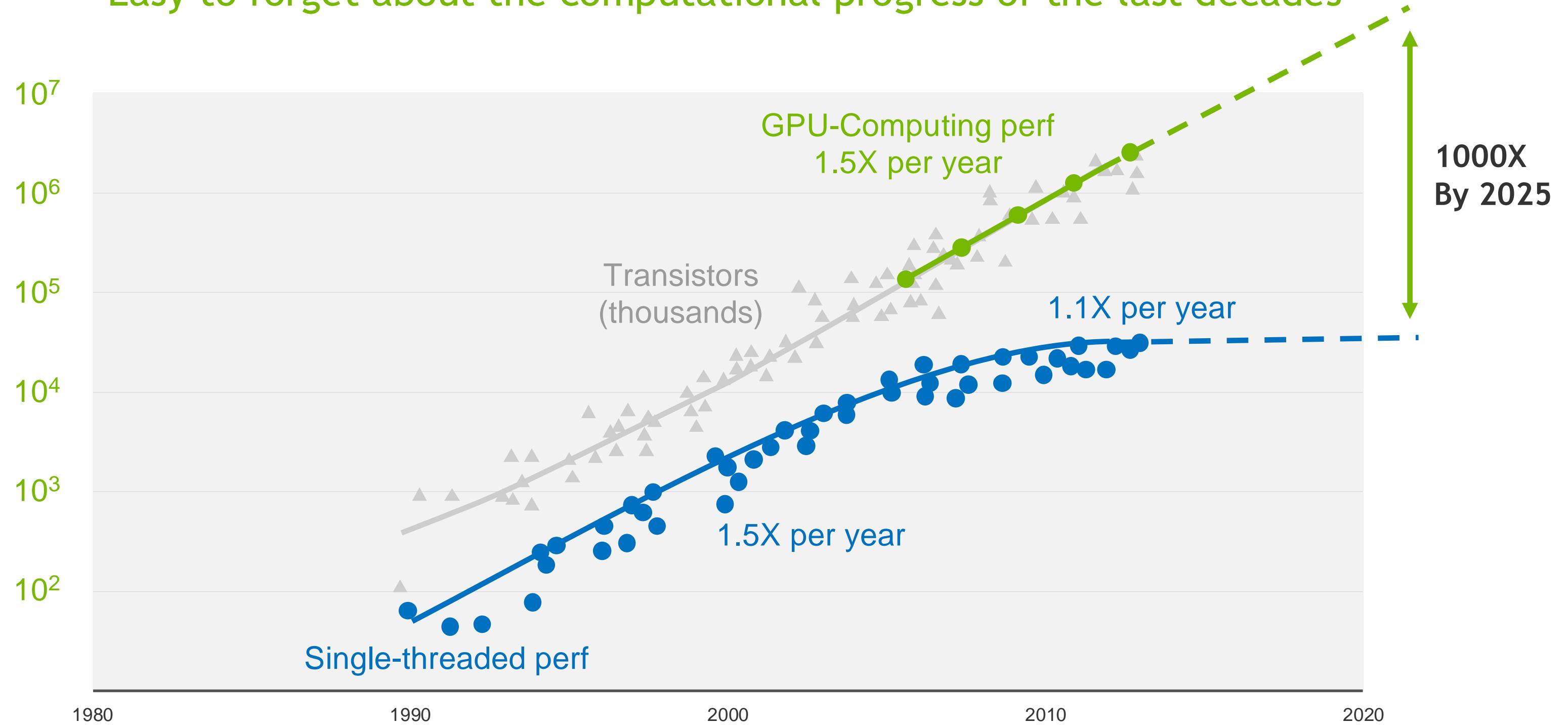


Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781* (2013).



# COMPUTE

Easy to forget about the computational progress of the last decades





CONTEXT

# CONTEXT

8 petaFLOPs in June 2011 (K Computer)



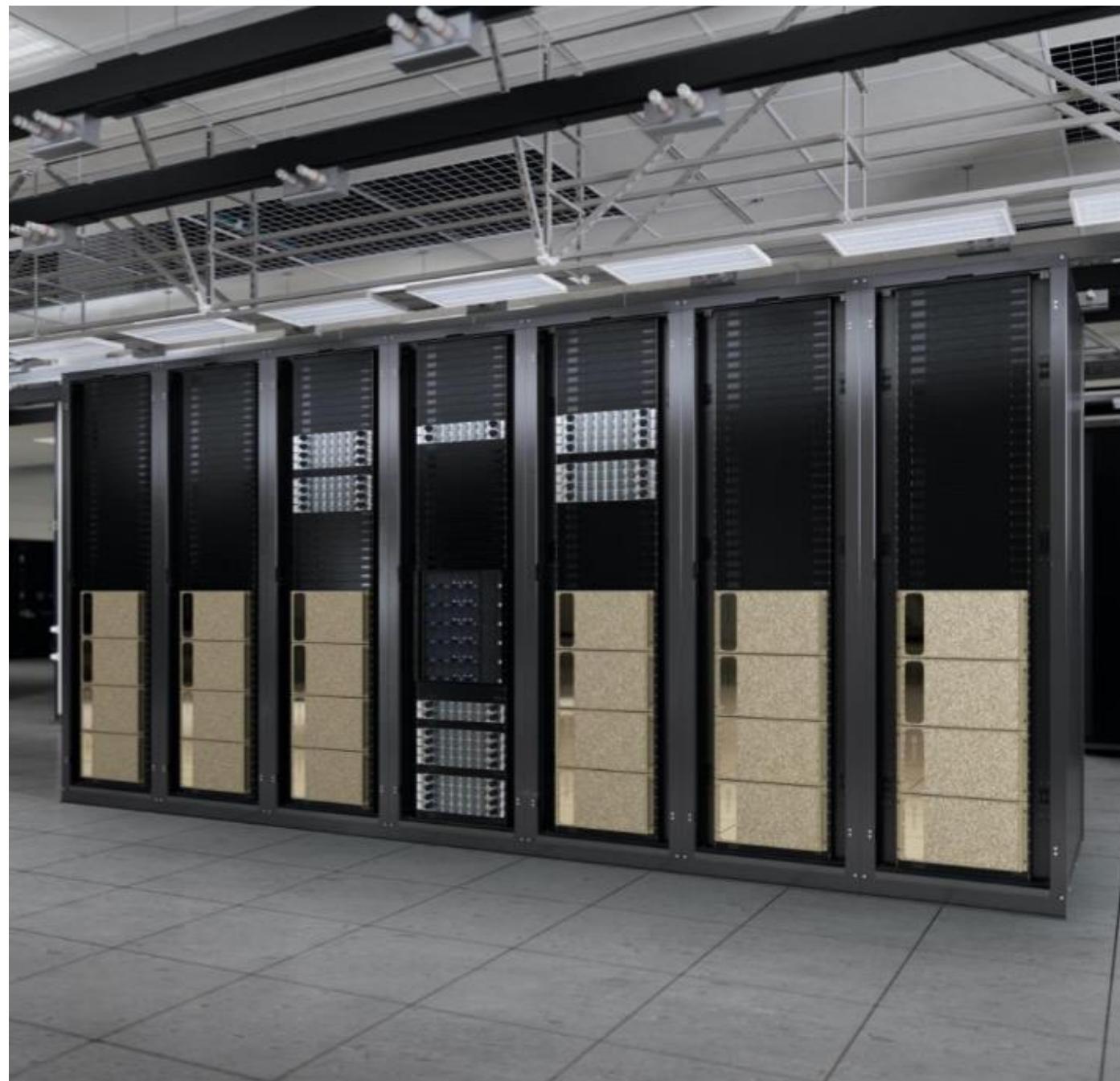
# CONTEXT

5 petaFLOPs for AI - today



# CONTEXT

~100 PFLOPS (FP16) or 48 PFLOPS (TF32) for AI - today





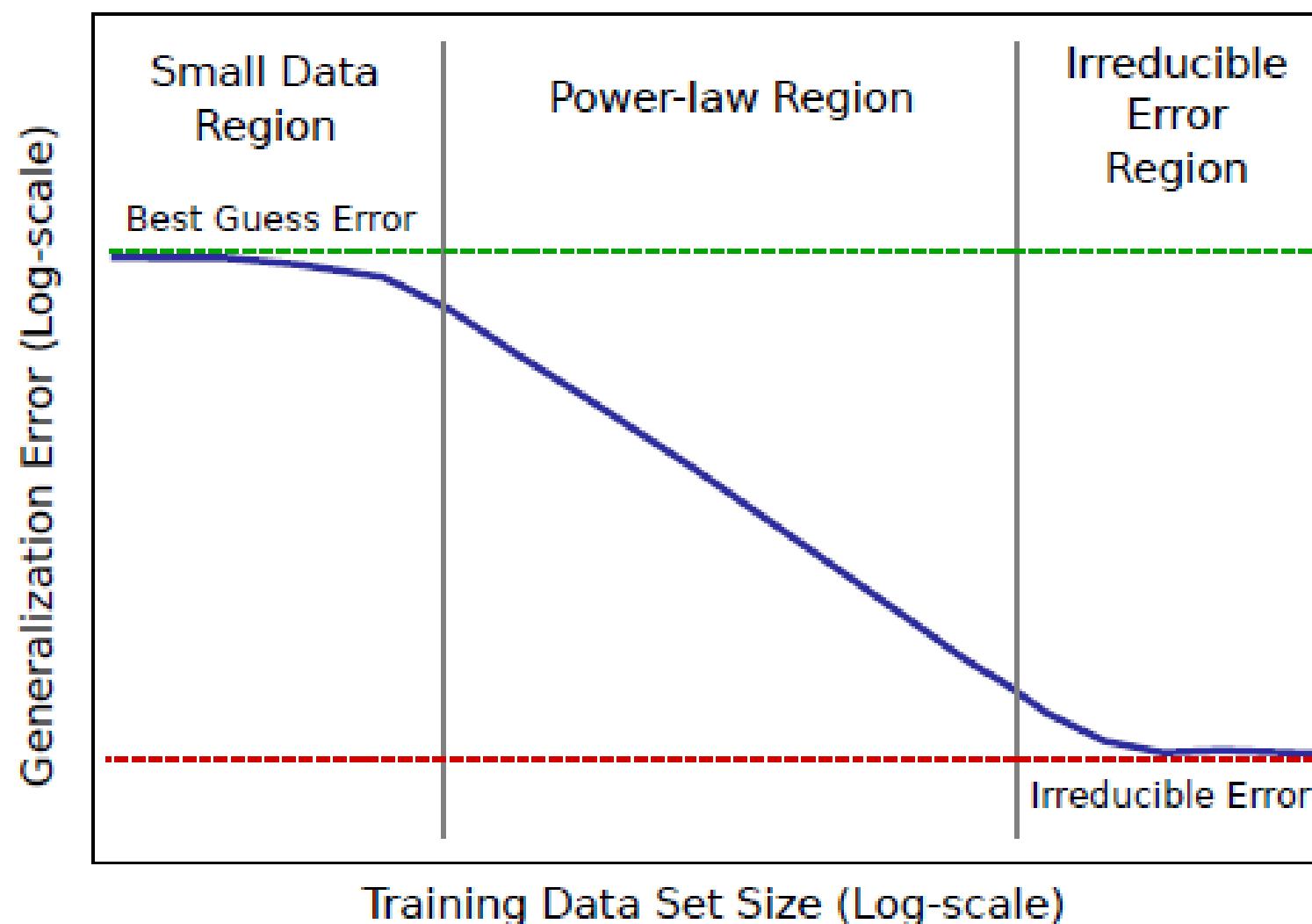
*100 EXAFLOPS*  
~=  
2 YEARS ON A DUAL CPU  
SERVER



SCALING LAWS

# SCALING LAWS

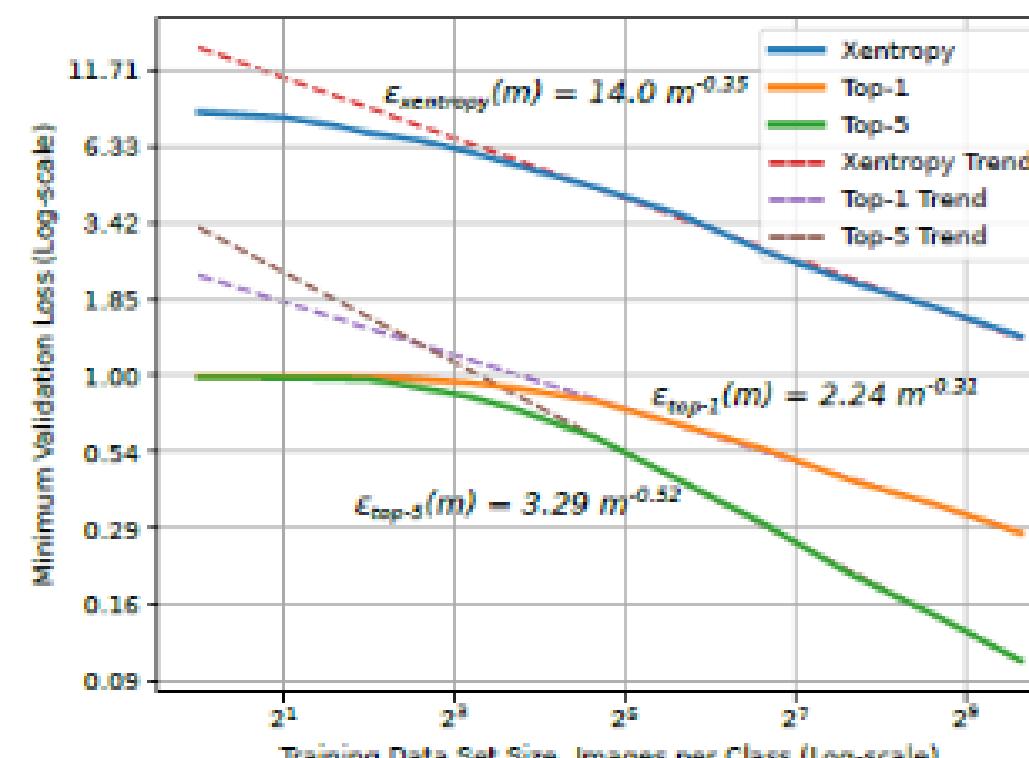
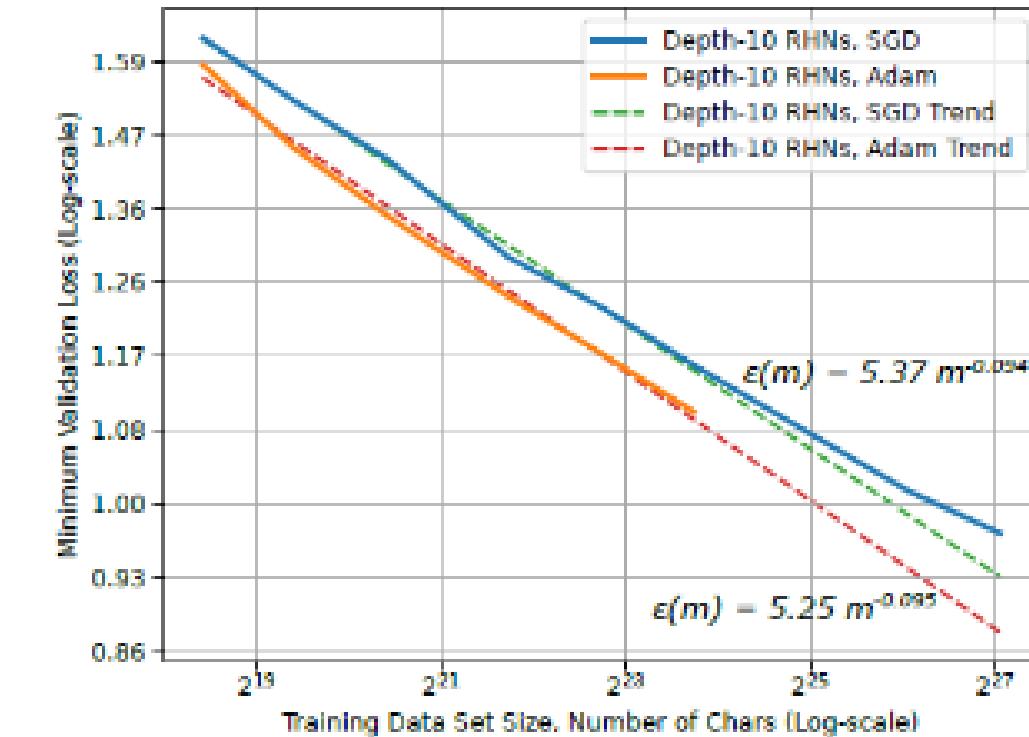
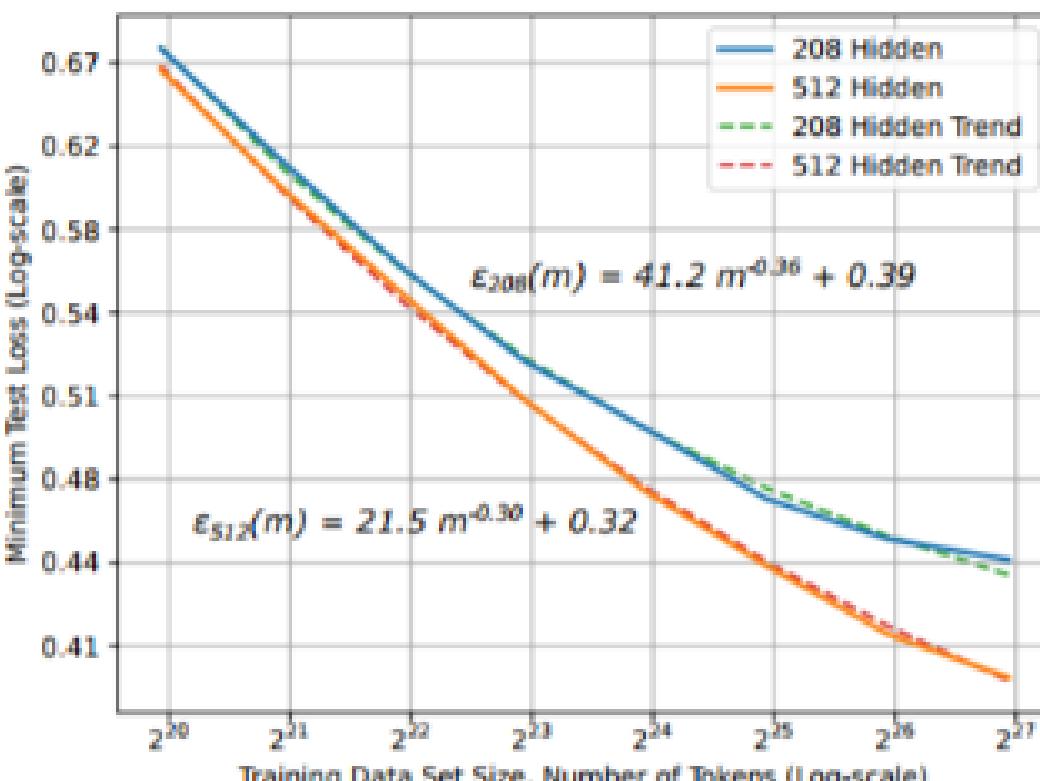
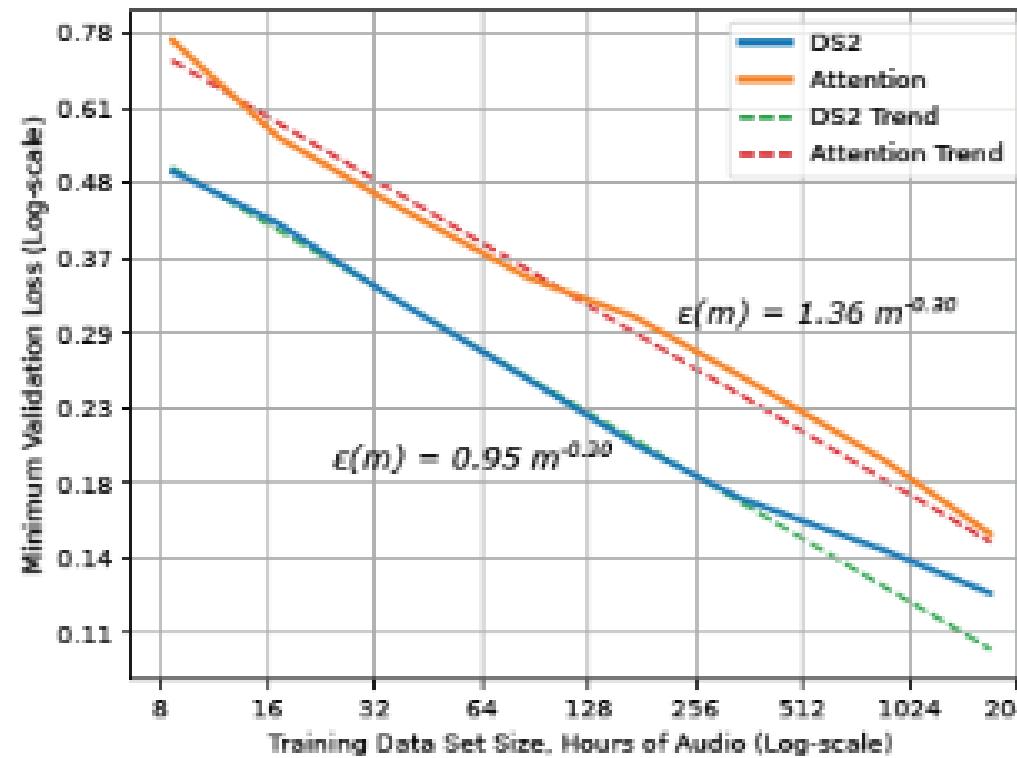
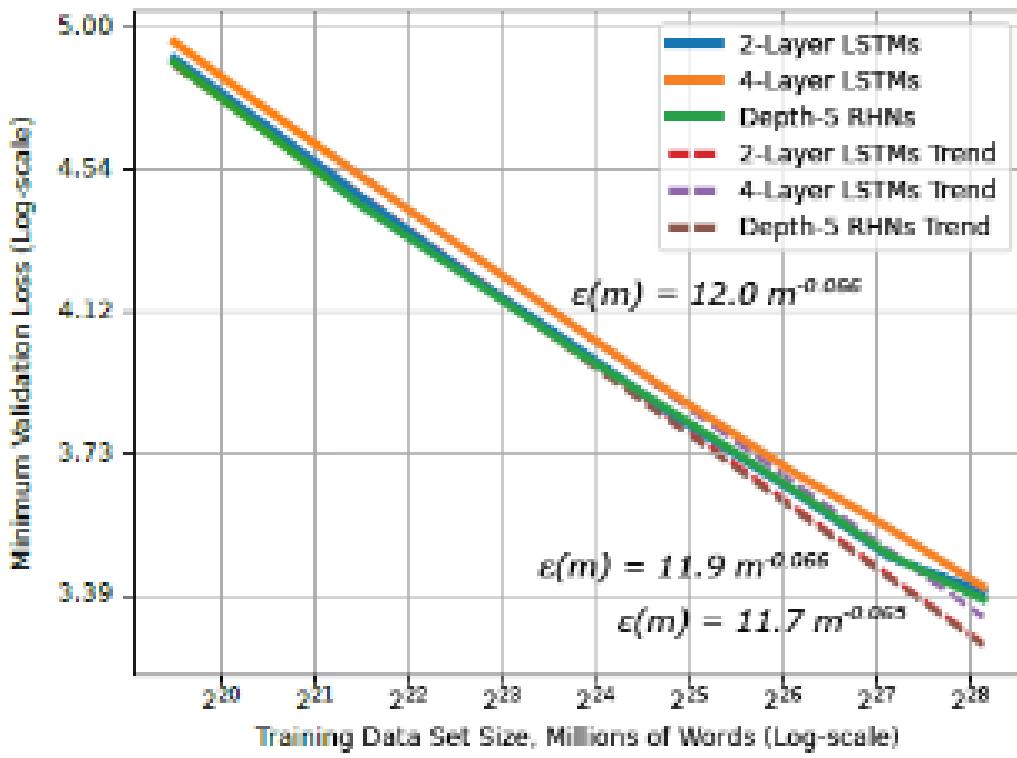
Power Law relationship between the dataset size and accuracy



# SCALING LAWS

## Applicable across all AI tasks

- Translation
- Language Models
- Character Language Models
- Image Classification
- Attention Speech Models

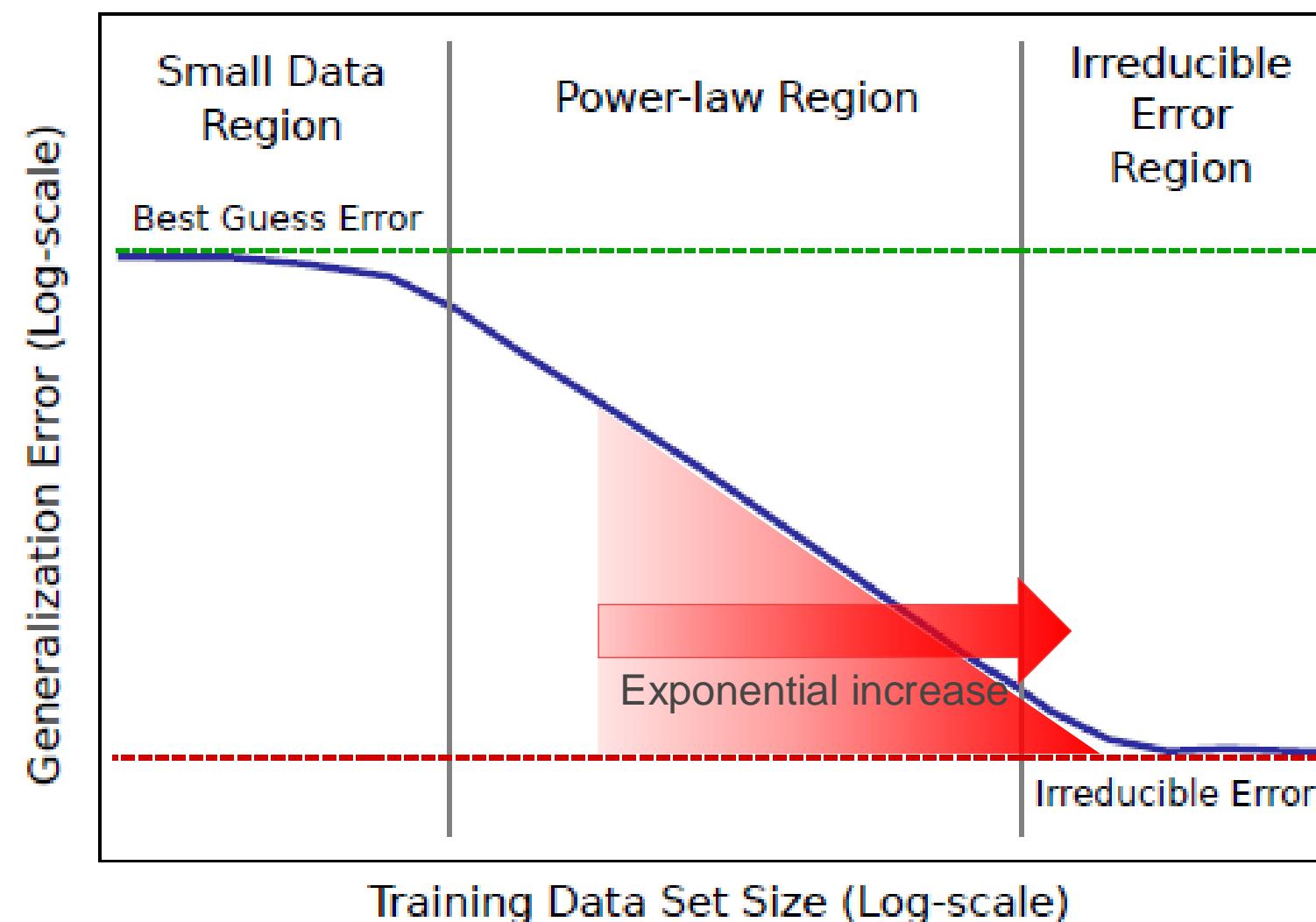


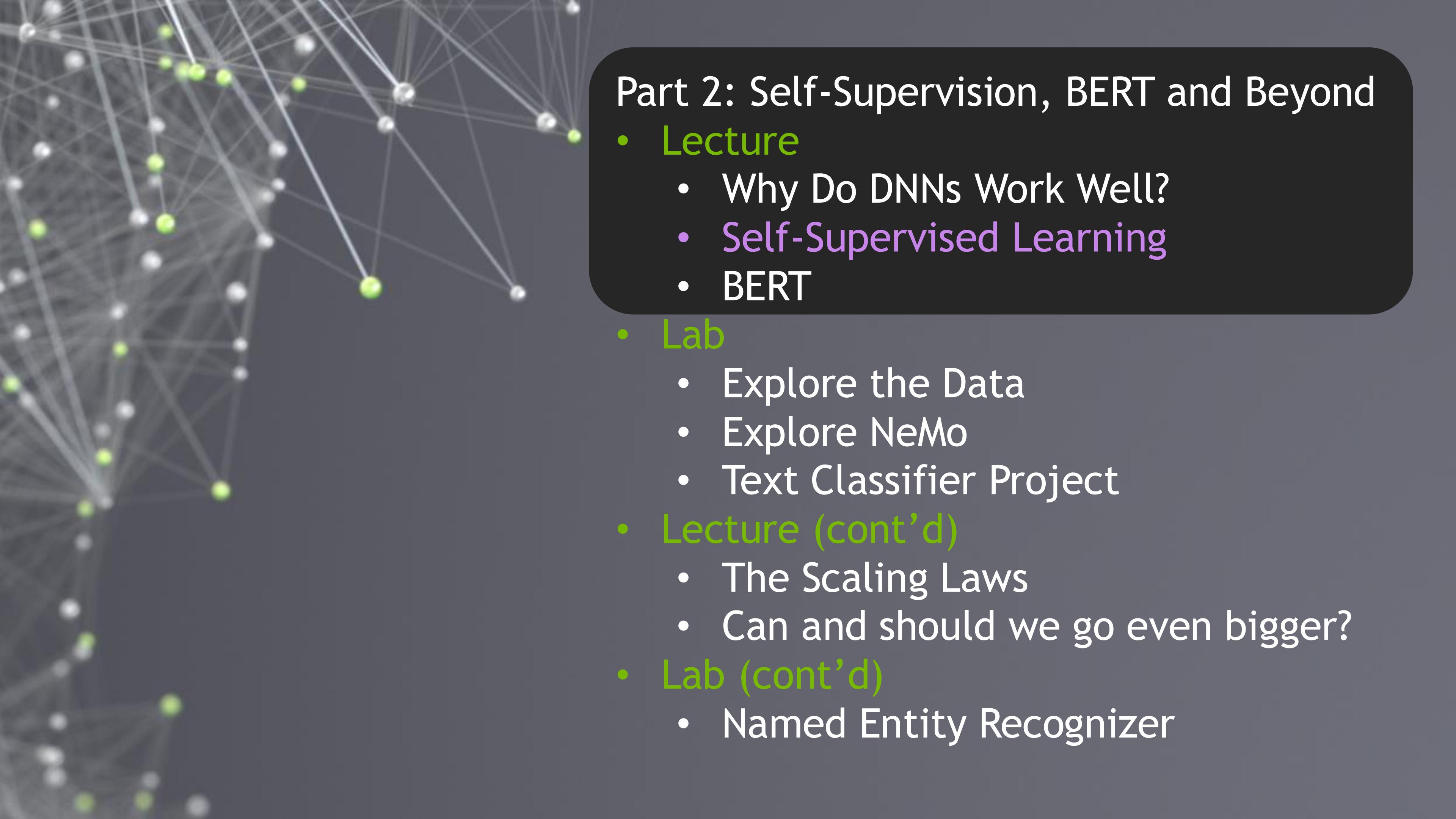
The background of the slide features a complex network graph. It consists of numerous small, semi-transparent white and light green circular nodes scattered across a dark gray background. These nodes are interconnected by a dense web of thin, light gray lines representing edges. Some clusters of nodes are more densely packed than others, creating a sense of organic connectivity.

THE COST

# THE COST OF LABELING

Limits the utility of deep learning models





## Part 2: Self-Supervision, BERT and Beyond

- **Lecture**
  - Why Do DNNs Work Well?
  - **Self-Supervised Learning**
  - BERT
- **Lab**
  - Explore the Data
  - Explore NeMo
  - Text Classifier Project
- **Lecture (cont'd)**
  - The Scaling Laws
  - Can and should we go even bigger?
- **Lab (cont'd)**
  - Named Entity Recognizer

# SELF-SUPERVISED LEARNING

## Example training tasks

- Natural Language Processing:
  - Masked Language Model: We mask a percentage of the input tokens at random (say 15%) and ask the neural network to predict the entire sentence
  - Next Sentence Prediction: We choose either two consecutive sentences from text, or two random sentences from the text. We ask the neural network to establish whether the two sentences occur one after another.
  - We use another simpler neural network to replace random words in the sequence and ask the primary neural network to detect which words were replaced (using a GAN like configuration).
- Computer Vision:
  - Contrastive Learning: Randomly modify (crop and resize, flip, distort color, rotate, cut-out, noise, blur, etc.) and either feed the same image, or two randomly selected images, into the neural network, asking it to say whether it is the same image or not
  - Noisy labels/Self Training: Use labels generated by a weak algorithm (potentially older generation of the target model) to train a target-robust feature extractor

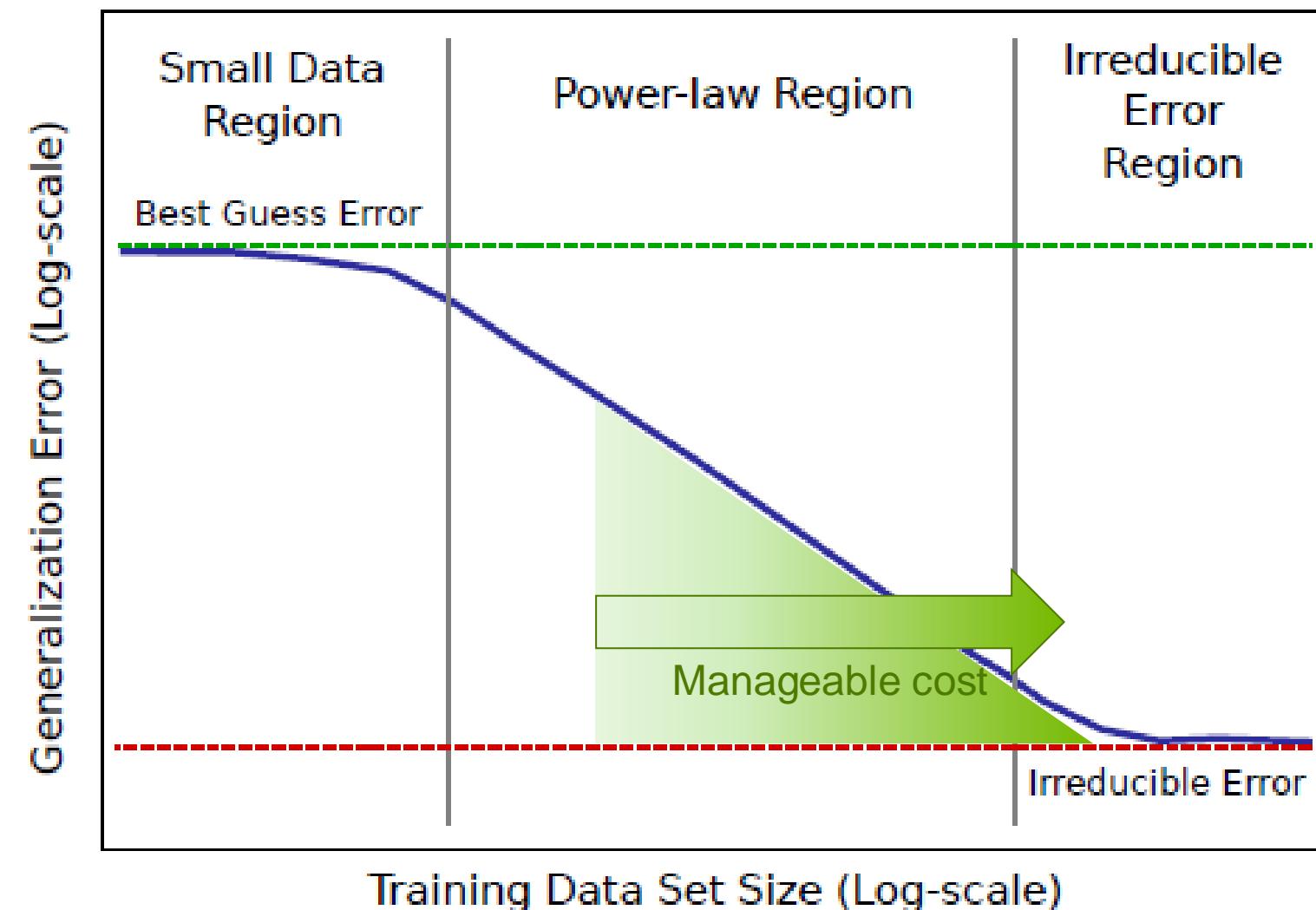
Dai, A. M., & Le, Q. V. (2015). Semi-supervised sequence learning. In Advances in neural information processing systems (pp. 3079-3087).

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. arXiv preprint arXiv:2002.05709.

Xie, Q., Hovy, E., Luong, M. T., & Le, Q. V. (2019). Self-training with Noisy Student improves ImageNet classification. arXiv preprint arXiv:1911.04252.

# THE COST OF LABELING

## Semi-supervised models



# SELF-SUPERVISED LEARNING

Abundance of unlabeled data

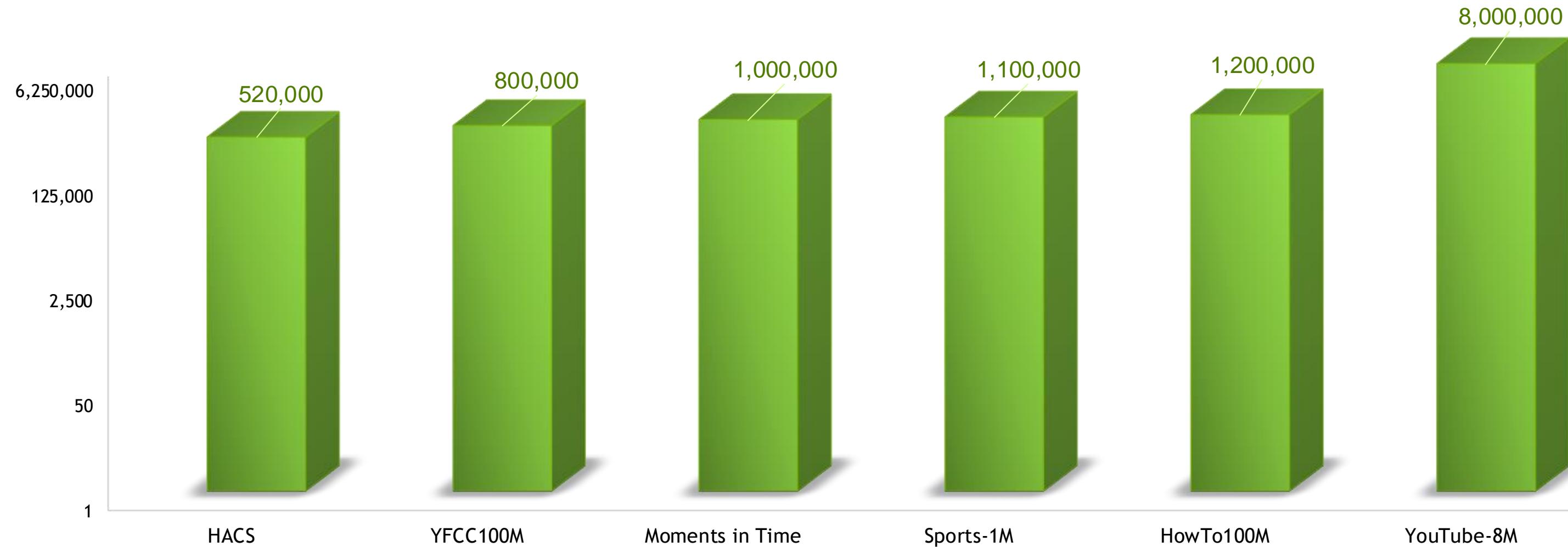
Number of Words (in Millions)



# SELF-SUPERVISED LEARNING

Abundance of unlabeled data

Number of videos



A complex network graph is displayed against a dark gray background. The graph consists of numerous small white dots representing nodes, connected by thin gray lines representing edges. Interspersed among these are several larger, glowing green dots. Some of these green nodes are interconnected by a dense cluster of lines, while others are isolated or part of smaller groups. The overall effect is one of a vast, interconnected system.

OLD IDEAS

# SELF-SUPERVISED LEARNING

## What was missing?

---

### Semi-supervised Sequence Learning

---

**Andrew M. Dai**

Google Inc.

[ada@google.com](mailto:ada@google.com)

**Quoc V. Le**

Google Inc.

[qvl@google.com](mailto:qvl@google.com)

#### Abstract

We present two approaches that use unlabeled data to improve sequence learning with recurrent networks. The first approach is to predict what comes next in a sequence, which is a conventional language model in natural language processing. The second approach is to use a sequence autoencoder, which reads the input sequence into a vector and predicts the input sequence again. These two algorithms can be used as a “pretraining” step for a later supervised sequence learning algorithm. In other words, the parameters obtained from the unsupervised step can be used as a starting point for other supervised training models. In our experiments, we find that long short term memory recurrent networks after being pretrained with the two approaches are more stable and generalize better. With pretraining, we are able to train long short term memory recurrent networks up to a few hundred timesteps, thereby achieving strong performance in many text classification tasks, such as IMDB, DBpedia and 20 Newsgroups.



THE SCALE

# GENERATIVE PRETRAINING (GPT)

## The scale

*“Many previous approaches to NLP tasks train relatively small models on a single GPU from scratch. Our approach requires an expensive pre-training step - 1 month on 8 GPUs. Luckily, this only has to be done once and we’re releasing our model so others can avoid it. It is also a large model (in comparison to prior work) and consequently uses more compute and memory — we used a 37-layer (12 block) Transformer architecture, and we train on sequences of up to 512 tokens. Most experiments were conducted on 4 and 8 GPU systems. The model does fine-tune to new tasks very quickly which helps mitigate the additional resource requirements.”*

# GENERATIVE PRETRAINING (GPT)

## The design

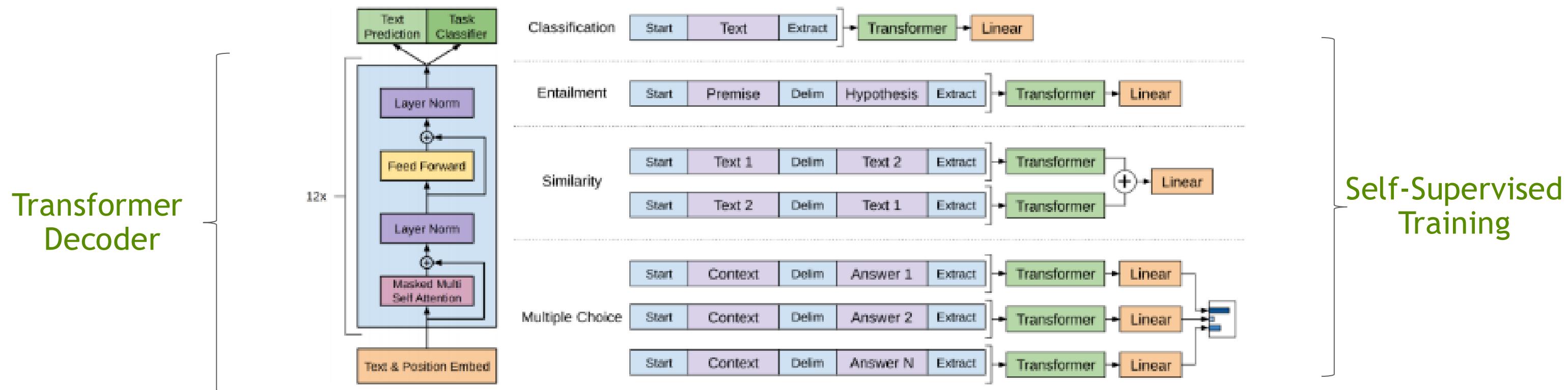


Figure 1: (left) Transformer architecture and training objectives used in this work. (right) Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.



IT BECAME POSSIBLE TO  
TRANSFER LEARN!

# GENERATIVE PRETRAINING (GPT)

## The approach



Pre-training our model on a large corpus of text significantly improves its performance on challenging natural language processing tasks like Winograd Schema Resolution.

# GENERATIVE PRETRAINING (GPT)

## The implications



Pre-training our model on a large corpus of text significantly improves its performance on challenging natural language processing tasks like Winograd Schema Resolution.

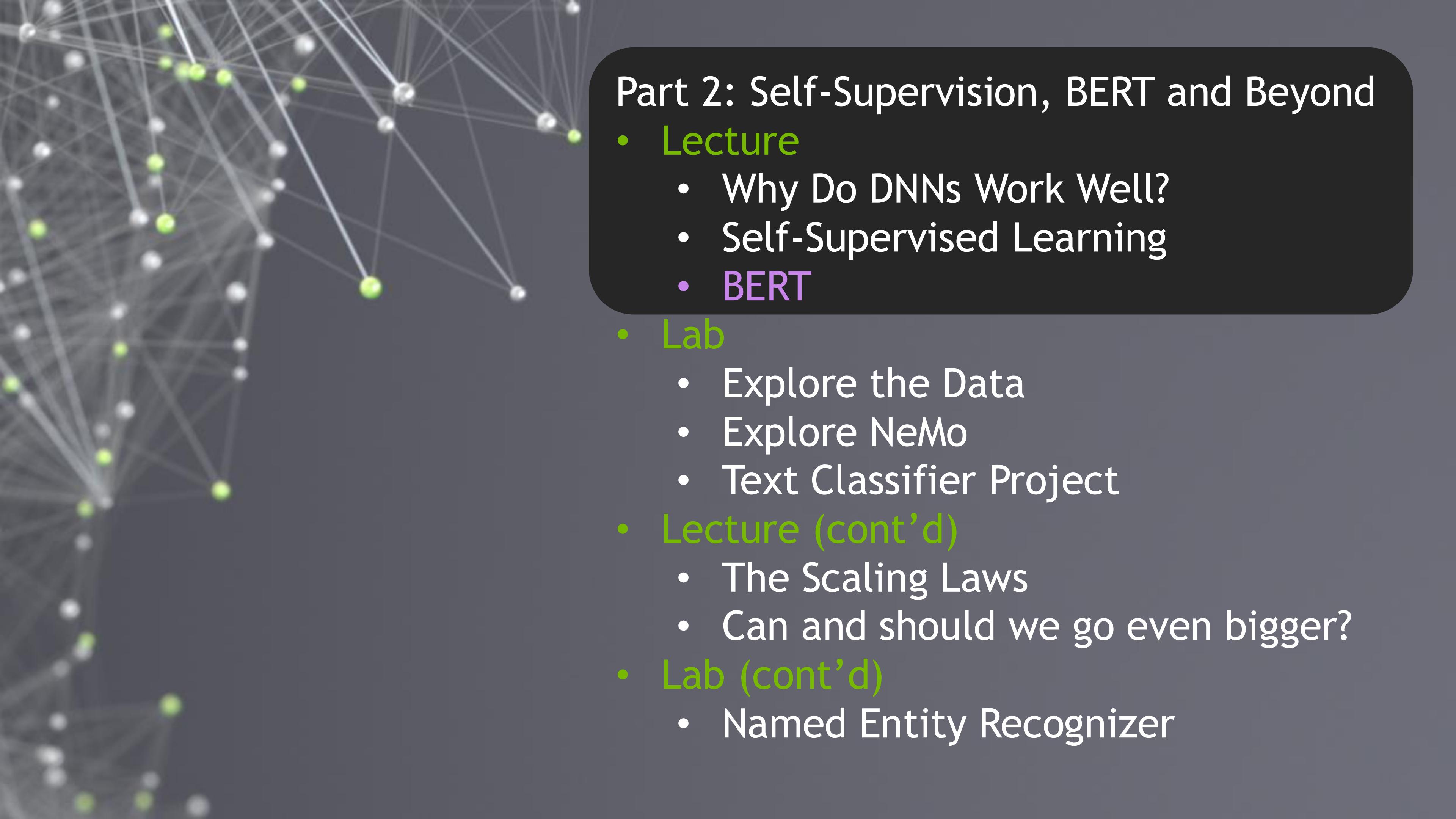


AND IT WORKED VERY  
WELL

# GENERATIVE PRETRAINING (GPT)

## The implications

Dataset	Task	SOTA	Ours
SNLI	Textual Entailment	89.3	<b>89.9</b>
MNLI Matched	Textual Entailment	80.6	<b>82.1</b>
MNLI Mismatched	Textual Entailment	80.1	<b>81.4</b>
SciTail	Textual Entailment	83.3	<b>88.3</b>
ONLI	Textual Entailment	82.3	<b>88.1</b>
RTE	Textual Entailment	<b>61.7</b>	56.0
STS-B	Semantic Similarity	81.0	<b>82.0</b>
QQP	Semantic Similarity	66.1	<b>70.3</b>
MRPC	Semantic Similarity	<b>86.0</b>	82.3
RACE	Reading Comprehension	53.3	<b>59.0</b>
ROCStories	Commonsense Reasoning	77.6	<b>86.5</b>
COPA	Commonsense Reasoning	71.2	<b>78.6</b>
SST-2	Sentiment Analysis	<b>93.2</b>	91.3
CoLA	Linguistic Acceptability	35.0	<b>45.4</b>
GLUE	Multi Task Benchmark	68.9	<b>72.8</b>

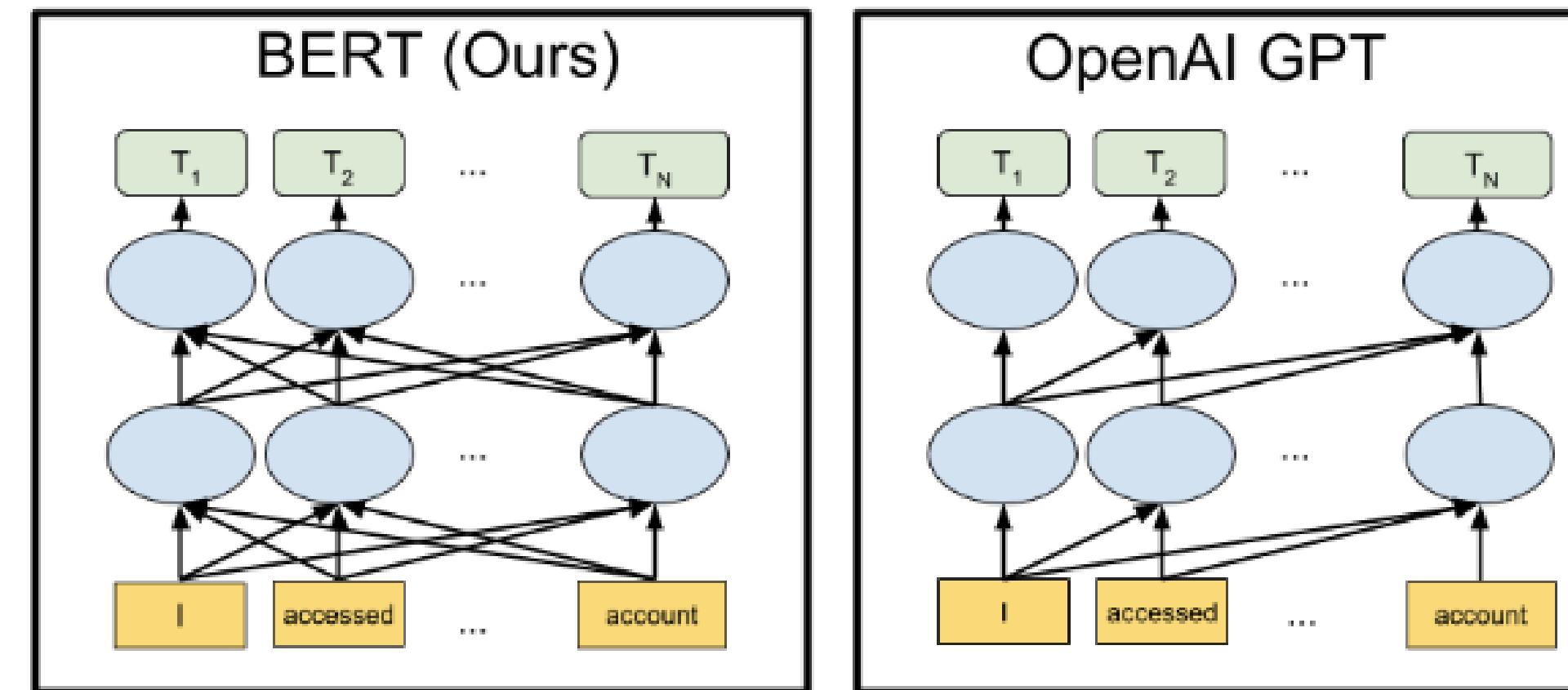


## Part 2: Self-Supervision, BERT and Beyond

- **Lecture**
  - Why Do DNNs Work Well?
  - Self-Supervised Learning
  - BERT
- **Lab**
  - Explore the Data
  - Explore NeMo
  - Text Classifier Project
- **Lecture (cont'd)**
  - The Scaling Laws
  - Can and should we go even bigger?
- **Lab (cont'd)**
  - Named Entity Recognizer

# BIDIRECTIONAL TRANSFORMERS (BERT)

Building on the shoulders of giants



# BIDIRECTIONAL TRANSFORMERS (BERT)

## The “pre” and “post” OpenAI ages

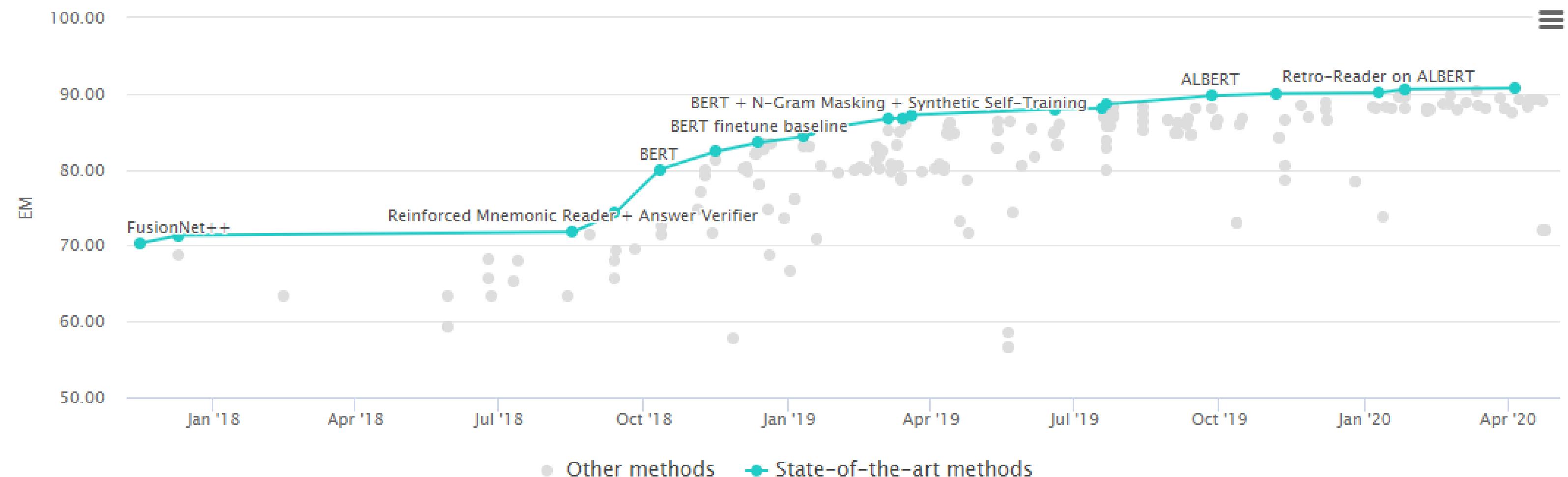
System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	Average
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
BERT <sub>BASE</sub>	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
BERT <sub>LARGE</sub>	<b>86.7/85.9</b>	<b>72.1</b>	<b>92.7</b>	<b>94.9</b>	<b>60.5</b>	<b>86.5</b>	<b>89.3</b>	<b>70.1</b>	<b>82.1</b>

Table 1: GLUE Test results, scored by the evaluation server (<https://gluebenchmark.com/leaderboard>). The number below each task denotes the number of training examples. The “Average” column is slightly different than the official GLUE score, since we exclude the problematic WNLI set.<sup>8</sup> BERT and OpenAI GPT are single-model, single task. F1 scores are reported for QQP and MRPC, Spearman correlations are reported for STS-B, and accuracy scores are reported for the other tasks. We exclude entries that use BERT as one of their components.

# SQuAD 2.0

Human performance 91.2

## Question Answering on SQuAD2.0



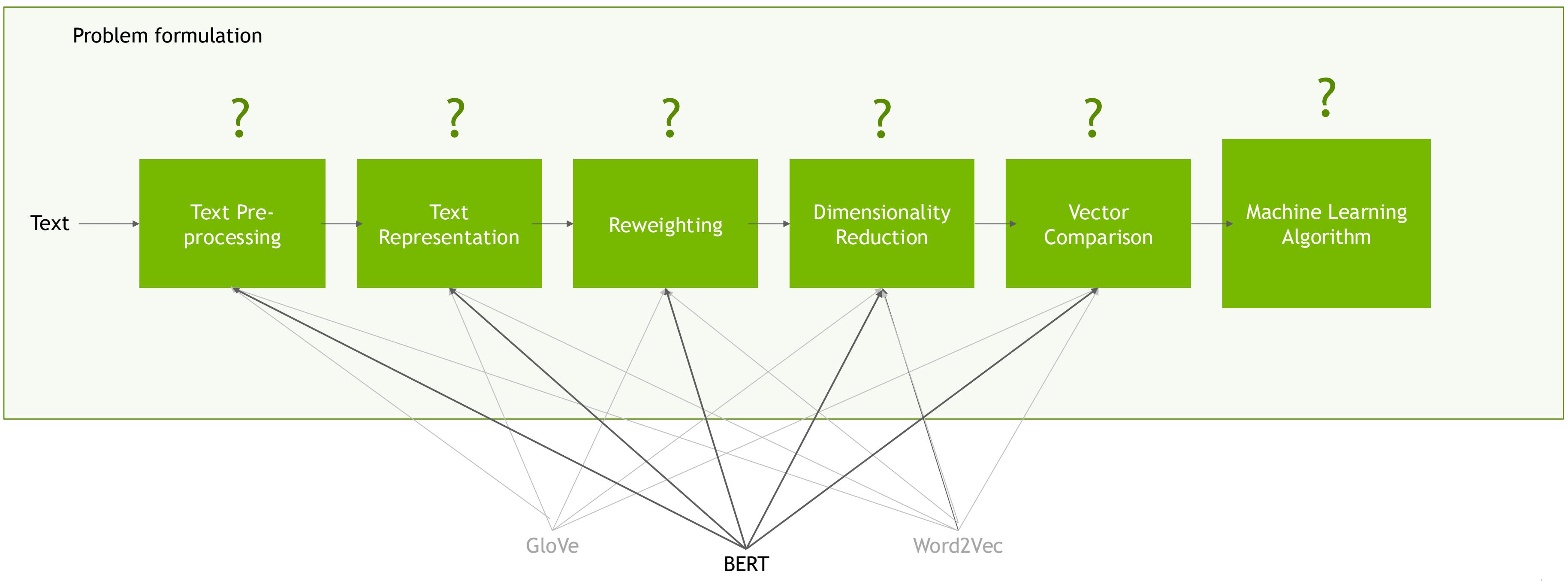


A network graph is displayed against a dark gray background. The graph consists of numerous small, semi-transparent circular nodes. Some nodes are white, while others are a vibrant lime green. These nodes are interconnected by a dense web of thin, light gray lines, representing connections or edges within the network. The overall effect is one of a complex, abstract system.

JUST YET ANOTHER  
UNSUPERVISED  
REPRESENTATION

# USING BERT

## Feature extractor





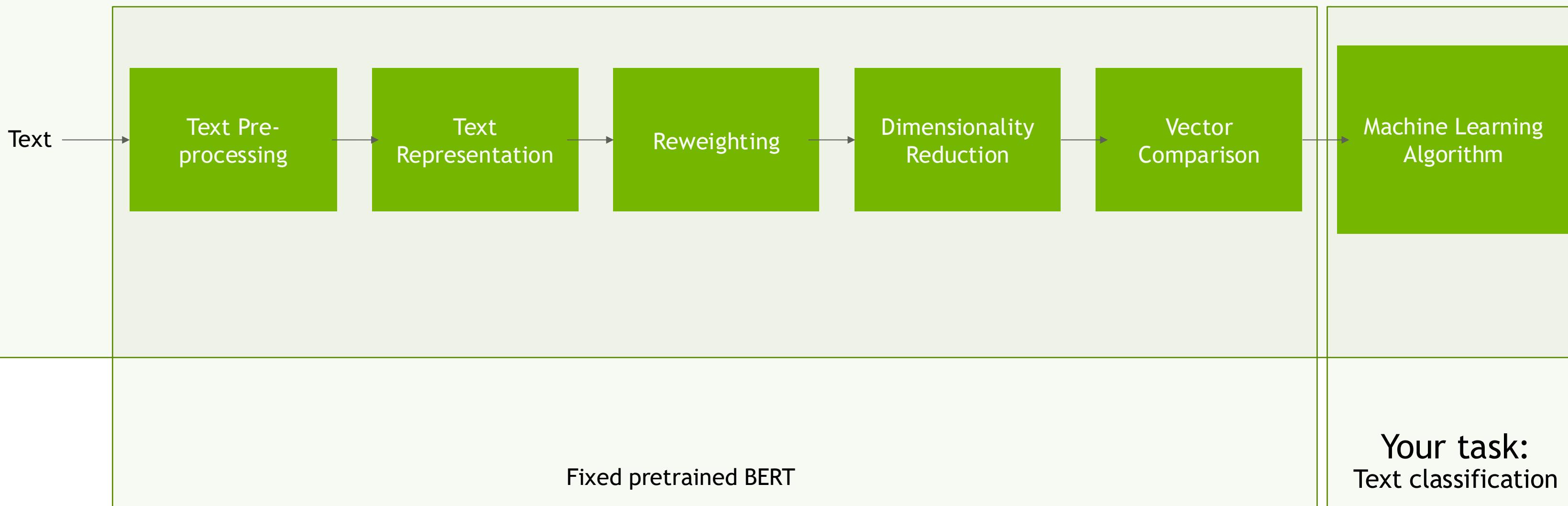
THE LAB

# LAB OVERVIEW

Notebooks 1, 2, 3

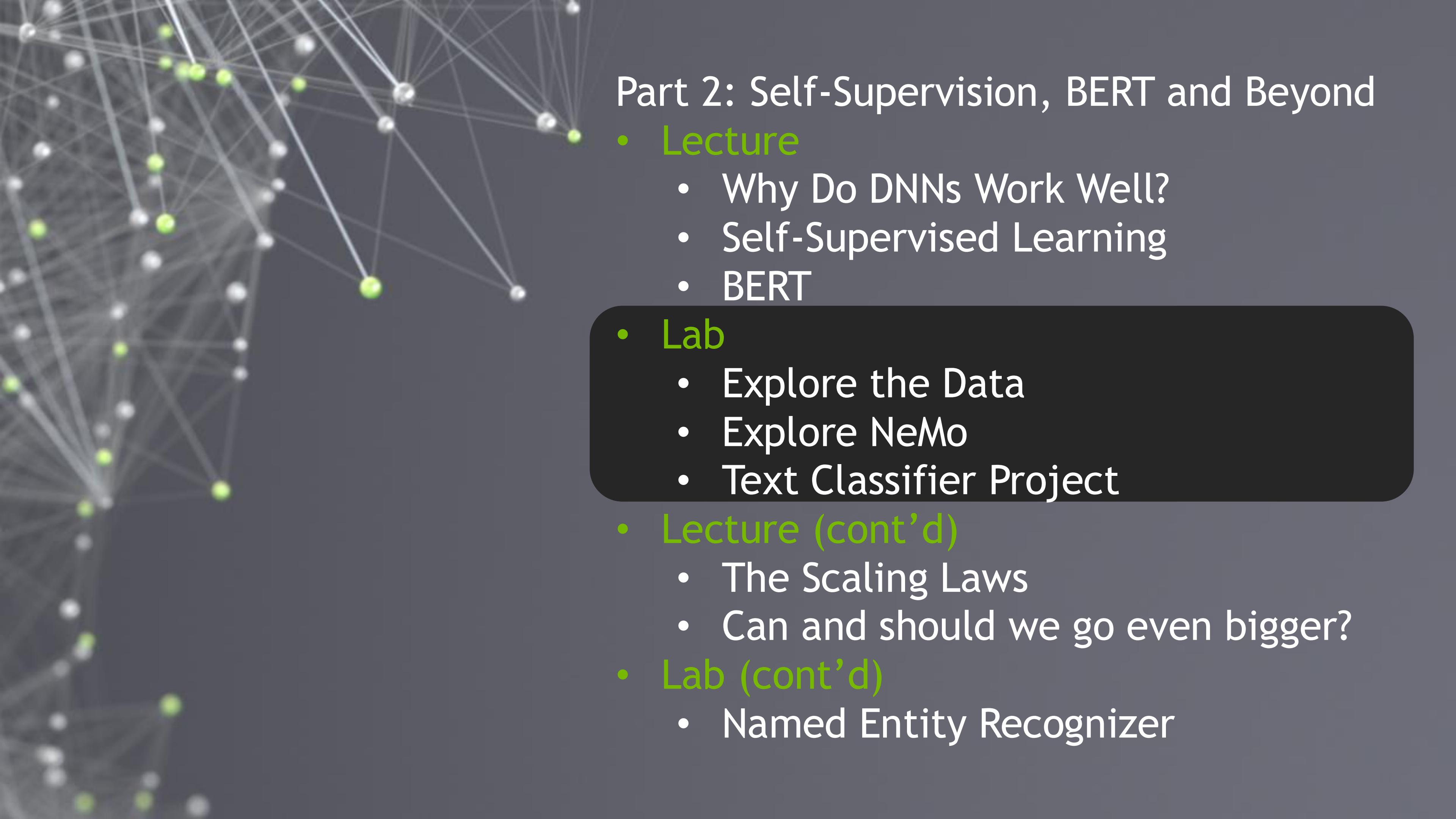
## Text classification

Problem formulation



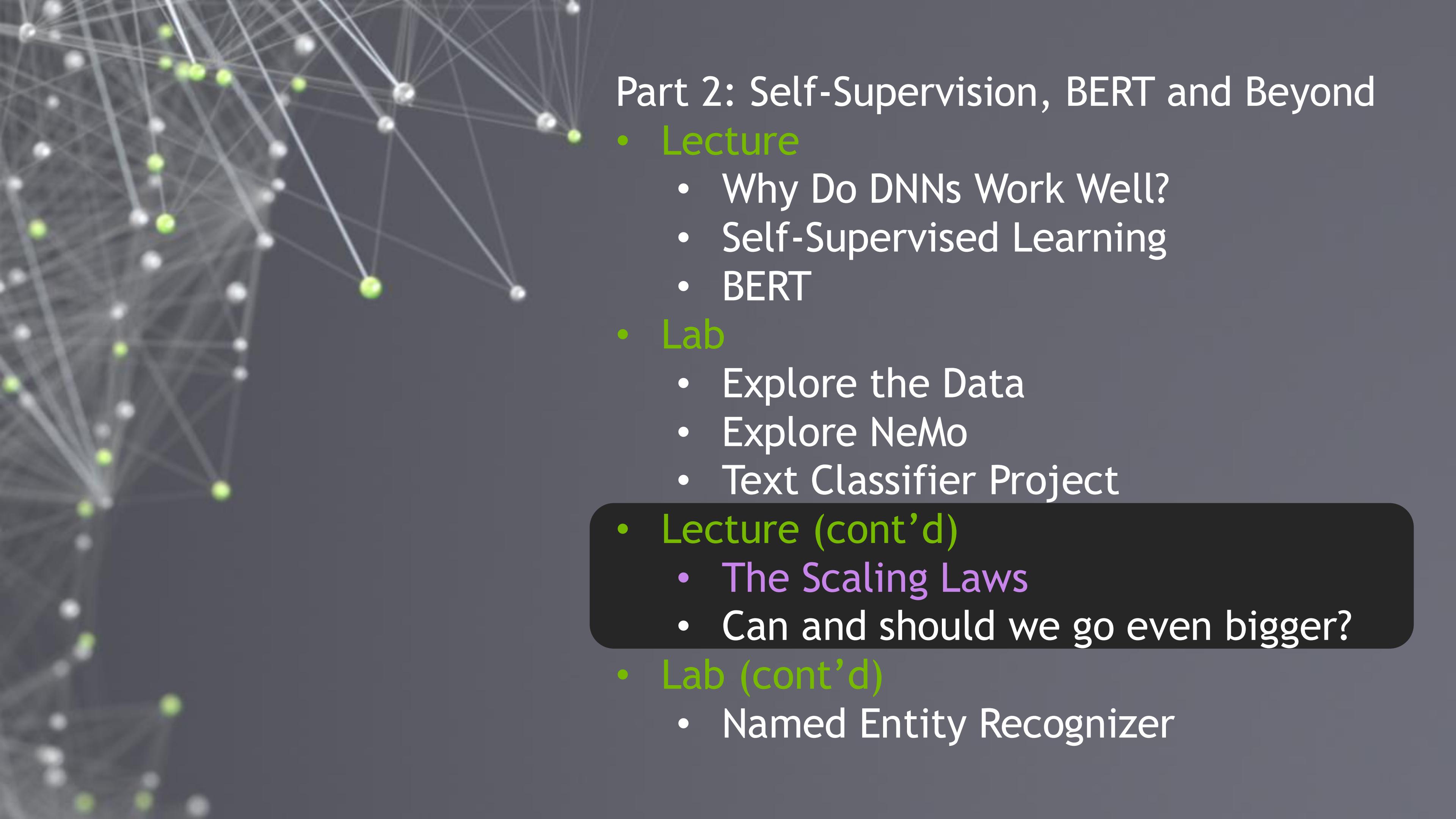
Fixed pretrained BERT

Your task:  
Text classification



## Part 2: Self-Supervision, BERT and Beyond

- **Lecture**
  - Why Do DNNs Work Well?
  - Self-Supervised Learning
  - BERT
- **Lab**
  - Explore the Data
  - Explore NeMo
  - Text Classifier Project
- **Lecture (cont'd)**
  - The Scaling Laws
  - Can and should we go even bigger?
- **Lab (cont'd)**
  - Named Entity Recognizer



## Part 2: Self-Supervision, BERT and Beyond

- **Lecture**
  - Why Do DNNs Work Well?
  - Self-Supervised Learning
  - BERT
- **Lab**
  - Explore the Data
  - Explore NeMo
  - Text Classifier Project
- **Lecture (cont'd)**
  - The Scaling Laws
  - Can and should we go even bigger?
- **Lab (cont'd)**
  - Named Entity Recognizer

# BIDIRECTIONAL TRANSFORMERS (BERT)

## Base vs Large

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	Average
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
BERT <sub>BASE</sub>	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
BERT <sub>LARGE</sub>	<b>86.7/85.9</b>	<b>72.1</b>	<b>92.7</b>	<b>94.9</b>	<b>60.5</b>	<b>86.5</b>	<b>89.3</b>	<b>70.1</b>	<b>82.1</b>

Table 1: GLUE Test results, scored by the evaluation server (<https://gluebenchmark.com/leaderboard>). The number below each task denotes the number of training examples. The “Average” column is slightly different than the official GLUE score, since we exclude the problematic WNLI set.<sup>8</sup> BERT and OpenAI GPT are single-model, single task. F1 scores are reported for QQP and MRPC, Spearman correlations are reported for STS-B, and accuracy scores are reported for the other tasks. We exclude entries that use BERT as one of their components.

# GPT-2

## GPT vs GPT-2

- Largely the same but:
  - Larger in every way:
    - More decoder layers: 12->48
    - Larger vocabulary: 50,257
    - Larger context: 512 -> 1024
    - Larger batch size
  - Changes to layer normalization
  - Different initialization scheme

Parameters	Layers	$d_{model}$
117M	12	768
345M	24	1024
762M	36	1280
1542M	48	1600

Table 2. Architecture hyperparameters for the 4 model sizes.

# GPT-2

## The Impact



### Transformer LM - 100M Parameters

Trained on Wikipedia minus "Abraham Lincoln" article.

Abraham Lincoln (February 27, 1724, Salem, Massachusetts - August 29, 1755) was an American Prime Minister who was an early proponent of an anti-witchcraft situation and was President of the Salem government from 1732 to 1743, President of the Massachusetts Supreme Court during that time, and President of the Institute for Religious Education to 1777, President of the Massachusetts Supreme Court, as well as a member of the Board of Missionaries of the Episcopal Church, The Reverend John Schuyler, Mercer Bishop, and Father New York State. He received his education at Paterson College and graduated from Paterson College and the Trenton College.

He wrote his autobiography in Mantua, Massachusetts, for which he was the supreme liberal. He later became one of Massachusetts's greatest religious leaders, choosing blacks many of whom became leaders of the Puritans. He received three honorary degrees from the universities of Boston and Virginia;

...



### Transformer LM - 5B Parameters

Trained on Wikipedia minus "Abraham Lincoln" article.

Abraham Lincoln (March 1, 1809 - March 15, 1865) was an American lawyer, jurist, and politician. He served as the 19th President of the United States, from March 1861 until his assassination in April 1865.

Born in Newark, New Jersey, Lincoln attended Knox College of New York, studied law, and was admitted to the bar Missouri Territory in 1826. He relocated to Illinois, and was elected to the Illinois House of Representatives in 1827, 1822, 1823, and 1826. He was elected Speaker of the Illinois House in 1825, served as Attorney General of the Illinois Territory in 1829, and was elected to the Illinois Senate in 1831. Lincoln was elected to the United States House of Representatives as a Democrat in 1833, and he was elected as a representative in the United States House of Representatives, serving two terms from 1839 to 1843. He was elected as a Whig in 1842, but he only served one term in Congress, and returned to the House of Representatives, serving

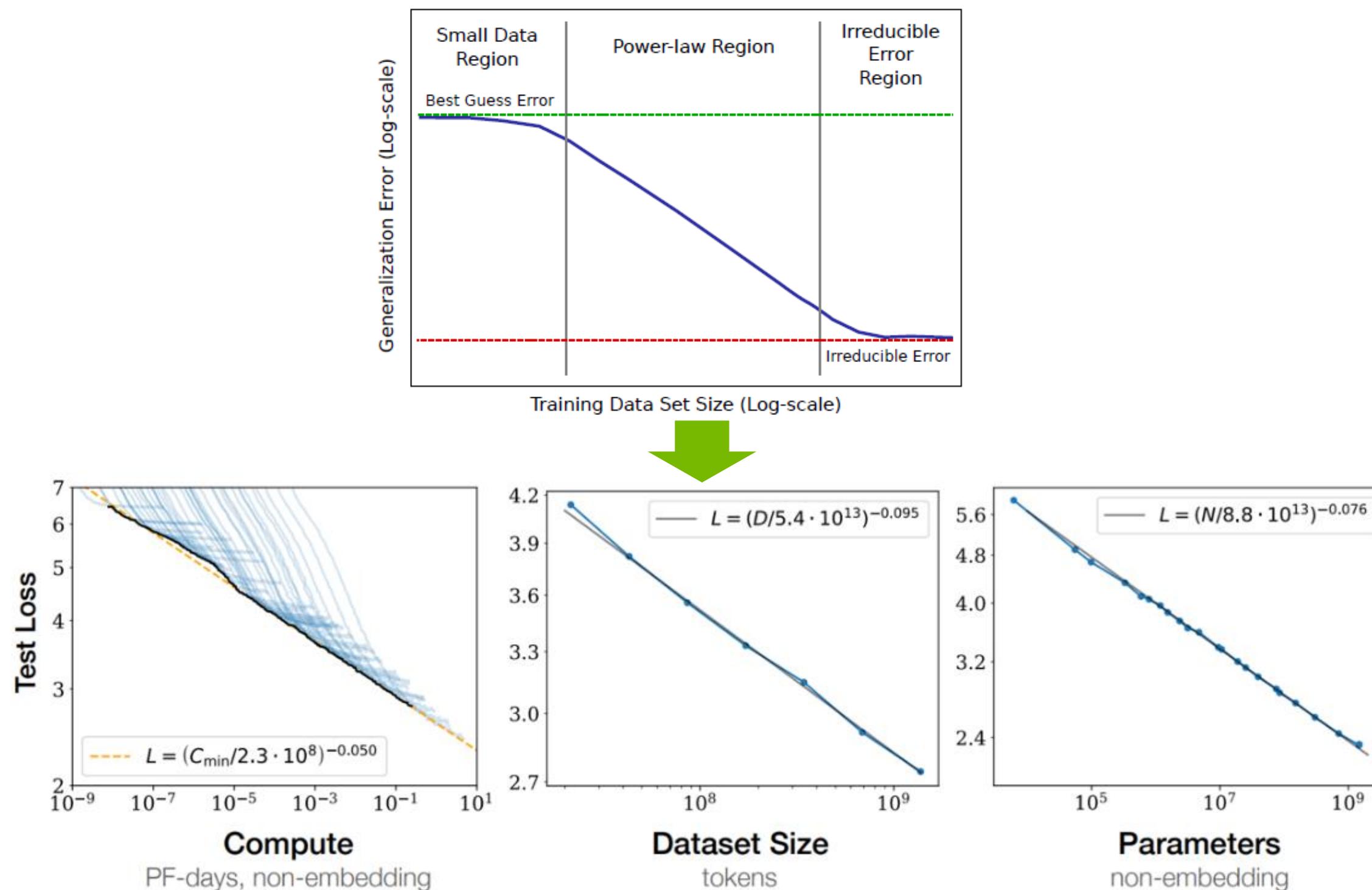
...



MEASURABLE  
IMPROVEMENT

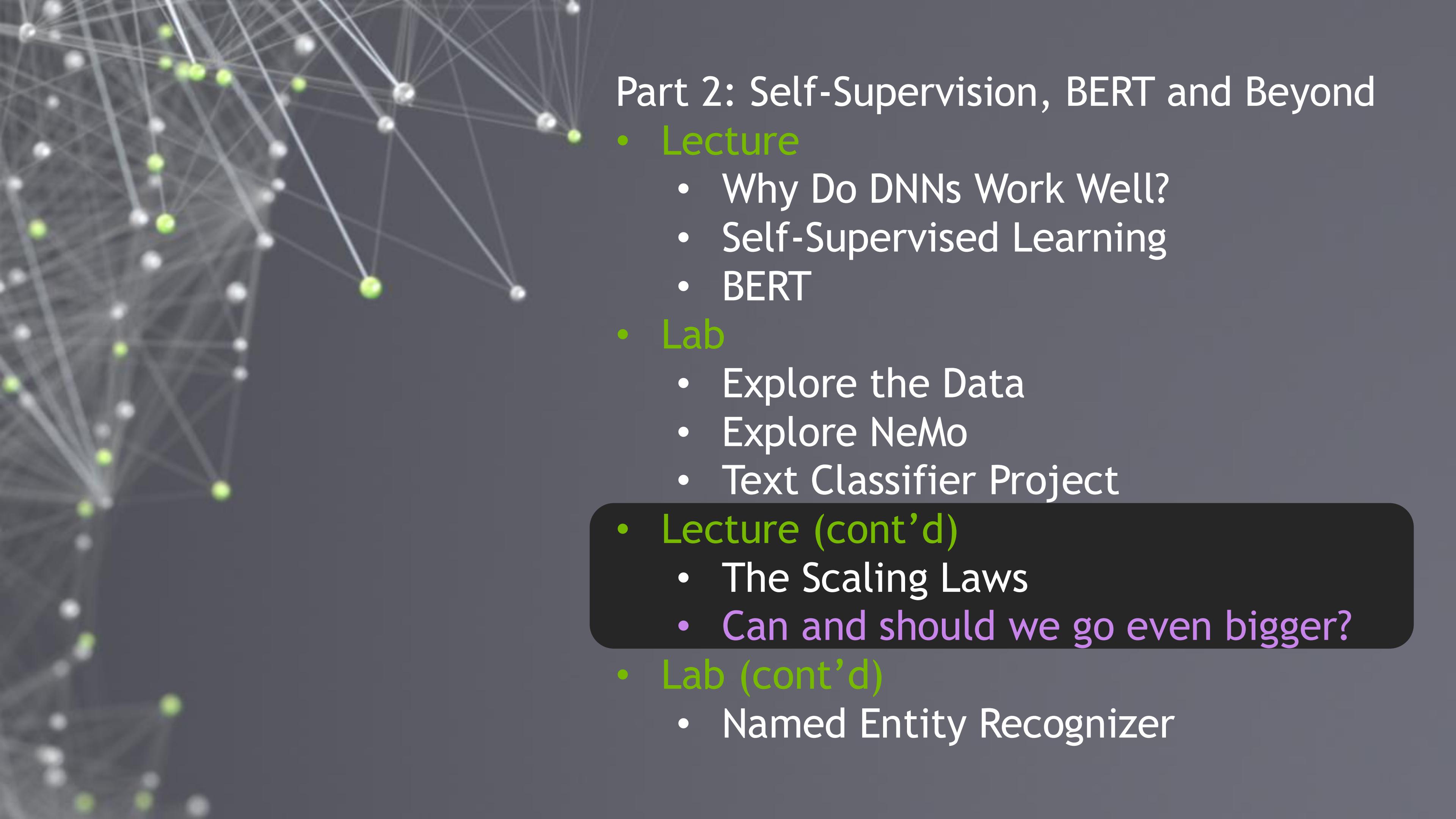
# THE SCALING LAWS OF NEURAL LANGUAGE MODELS

## Continuous improvement



Hestness, J., Narang, S., Ardalani, N., Diamos, G., Jun, H., Kianinejad, H., ... & Zhou, Y. Deep Learning Scaling is Predictable, Empirically. arXiv preprint arXiv:1712.00409. 2017

Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B. Brown, Prafulla Dhariwal, Scott Gray, Chris Hallacy, Benjamin Mann, Alec Radford, Aditya Ramesh, Nick Ryder, Daniel M. Ziegler, John Schulman, Dario Amodei, Sam McCandlish. Scaling Laws for Autoregressive Generative Modeling. 2020



## Part 2: Self-Supervision, BERT and Beyond

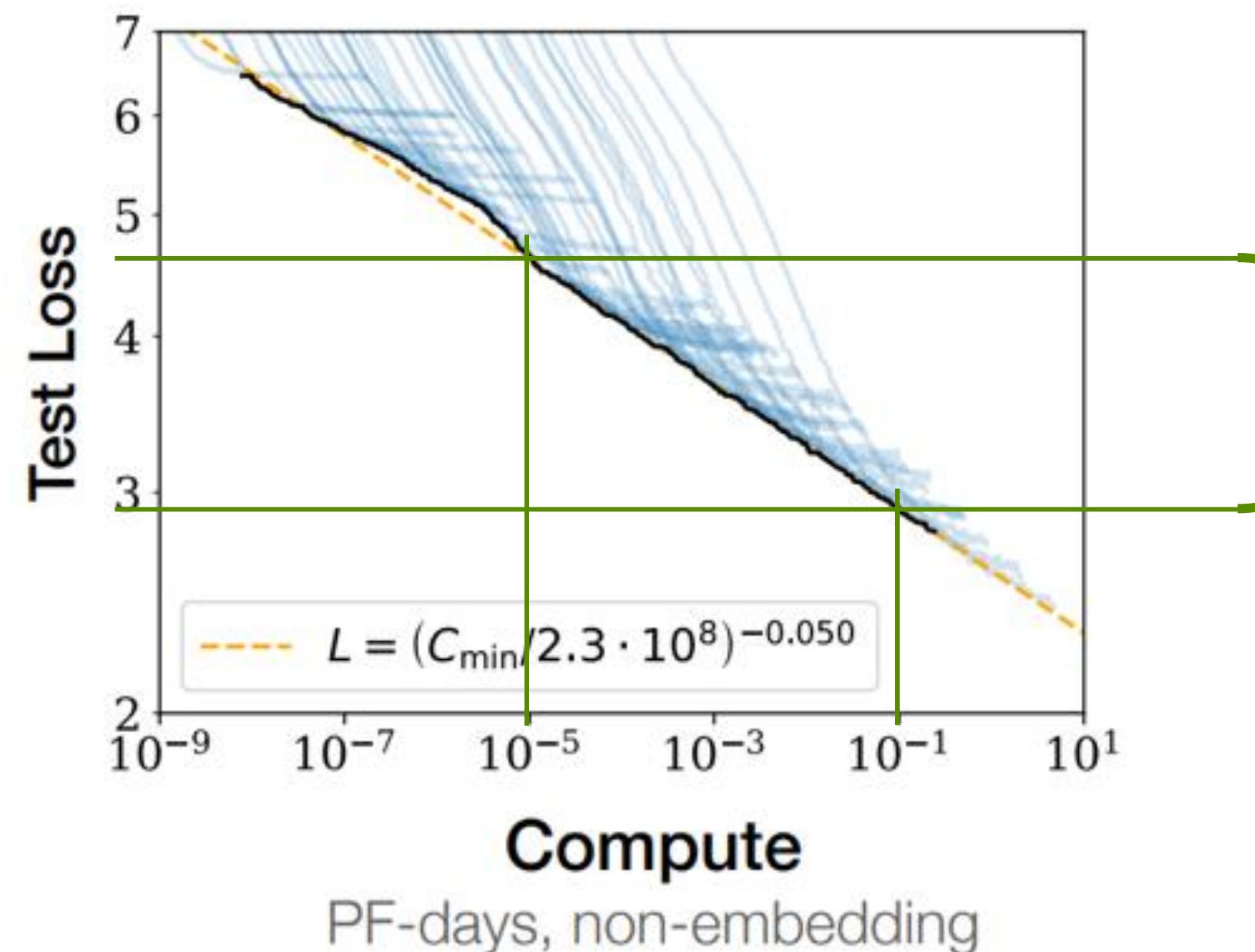
- Lecture
  - Why Do DNNs Work Well?
  - Self-Supervised Learning
  - BERT
- Lab
  - Explore the Data
  - Explore NeMo
  - Text Classifier Project
- Lecture (cont'd)
  - The Scaling Laws
  - Can and should we go even bigger?
- Lab (cont'd)
  - Named Entity Recognizer



SHOULD WE BUILD  
LARGER MODELS?

# ARE LARGE LANGUAGE MODELS WORTH IT?

The cost of incremental improvement



Are we building those models only for the small incremental improvement in their performance?

Is it worth all of the engineering and computational investment?



IS THIS REALLY THE ONLY  
THING WE HAVE  
ACHIEVED?



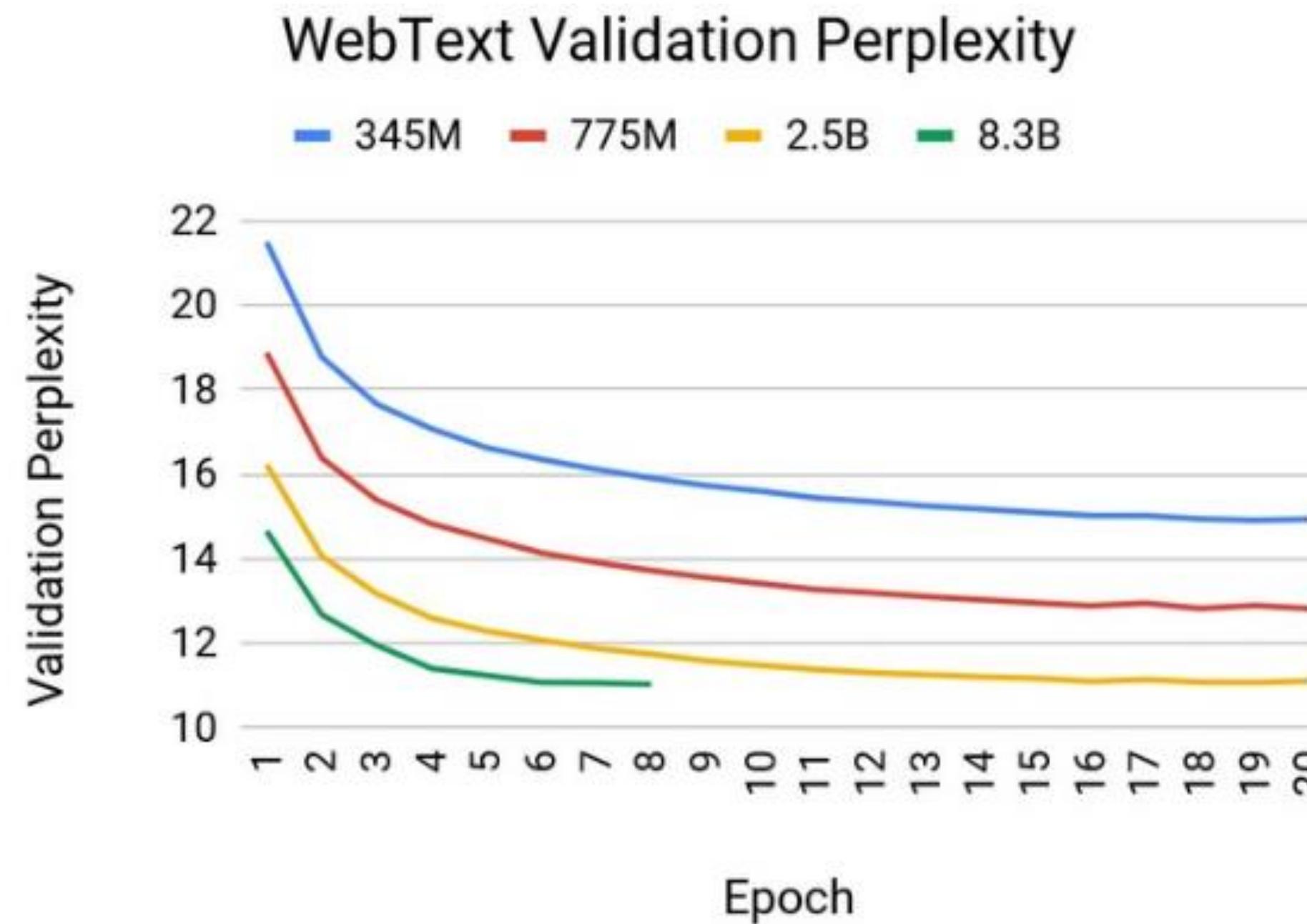
IT IS MUCH MORE THAN  
JUST INCREMENTAL  
INCREASE IN ACCURACY!



# 1. SAMPLE EFFICIENCY

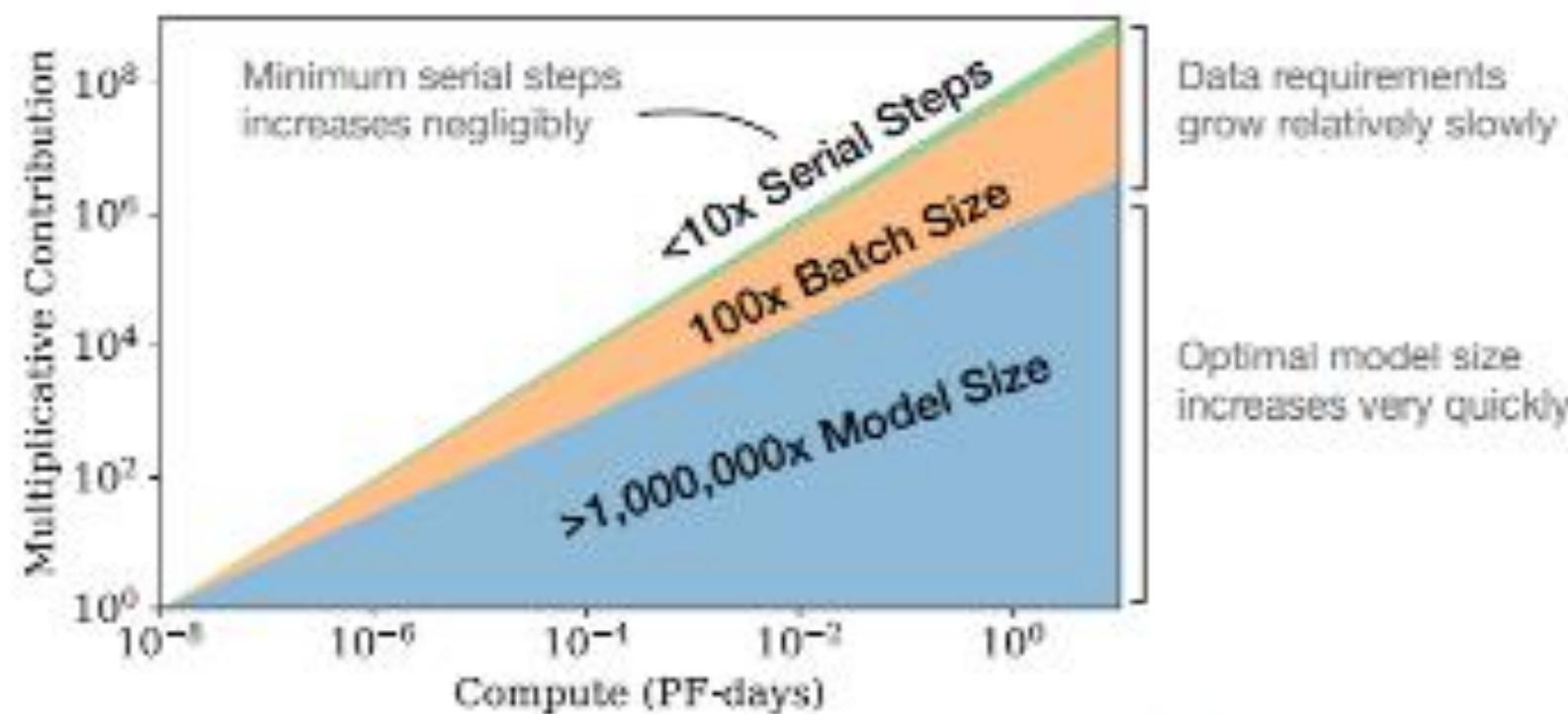
# NOT ABOUT INCREMENTAL IMPROVEMENT

Sample efficiency



# LARGER MODELS ARE CHEAPER TO TRAIN

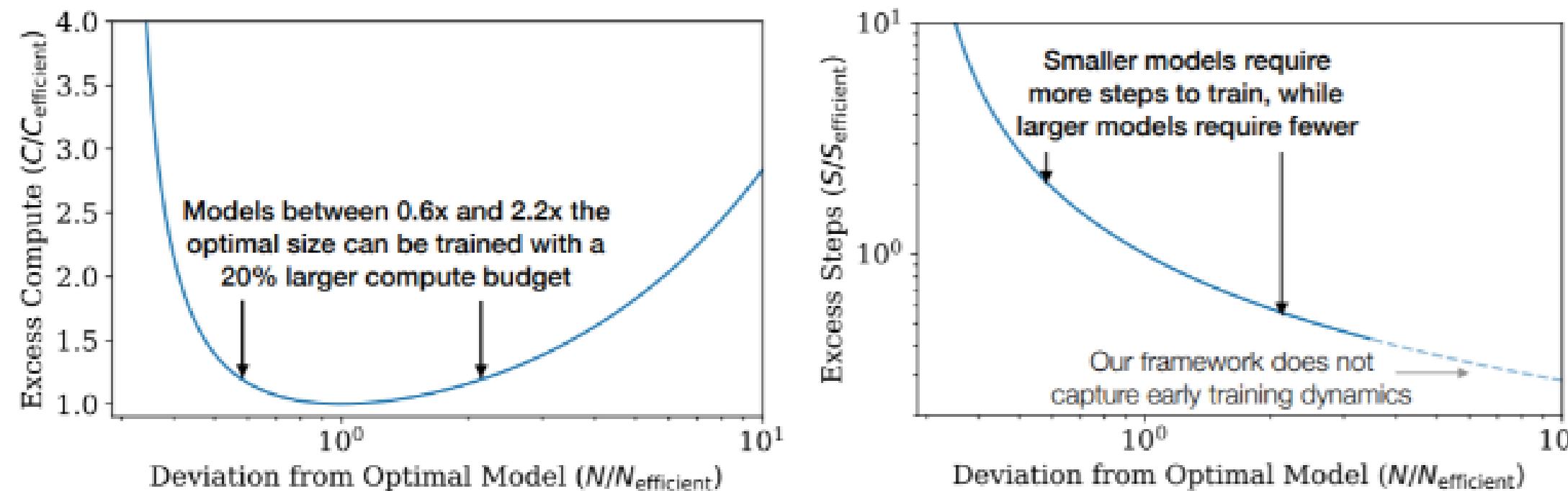
## Optimal allocation of computational budget



**Figure 3** As more compute becomes available, we can choose how much to allocate towards training larger models, using larger batches, and training for more steps. We illustrate this for a billion-fold increase in compute. For optimally compute-efficient training, most of the increase should go towards increased model size. A relatively small increase in data is needed to avoid reuse. Of the increase in data, most can be used to increase parallelism through larger batch sizes, with only a very small increase in serial training time required.

# LARGER MODELS ARE CHEAPER TO TRAIN

For every dataset there exists an optimal model size minimizing compute



**Figure 12** **Left:** Given a fixed compute budget, a particular model size is optimal, though somewhat larger or smaller models can be trained with minimal additional compute. **Right:** Models larger than the compute-efficient size require fewer steps to train, allowing for potentially faster training if sufficient additional parallelism is possible. Note that this equation should not be trusted for very large models, as it is only valid in the power-law region of the learning curve, after initial transient effects.



## 2. ARCHITECTURAL HYPERPARAMETERS

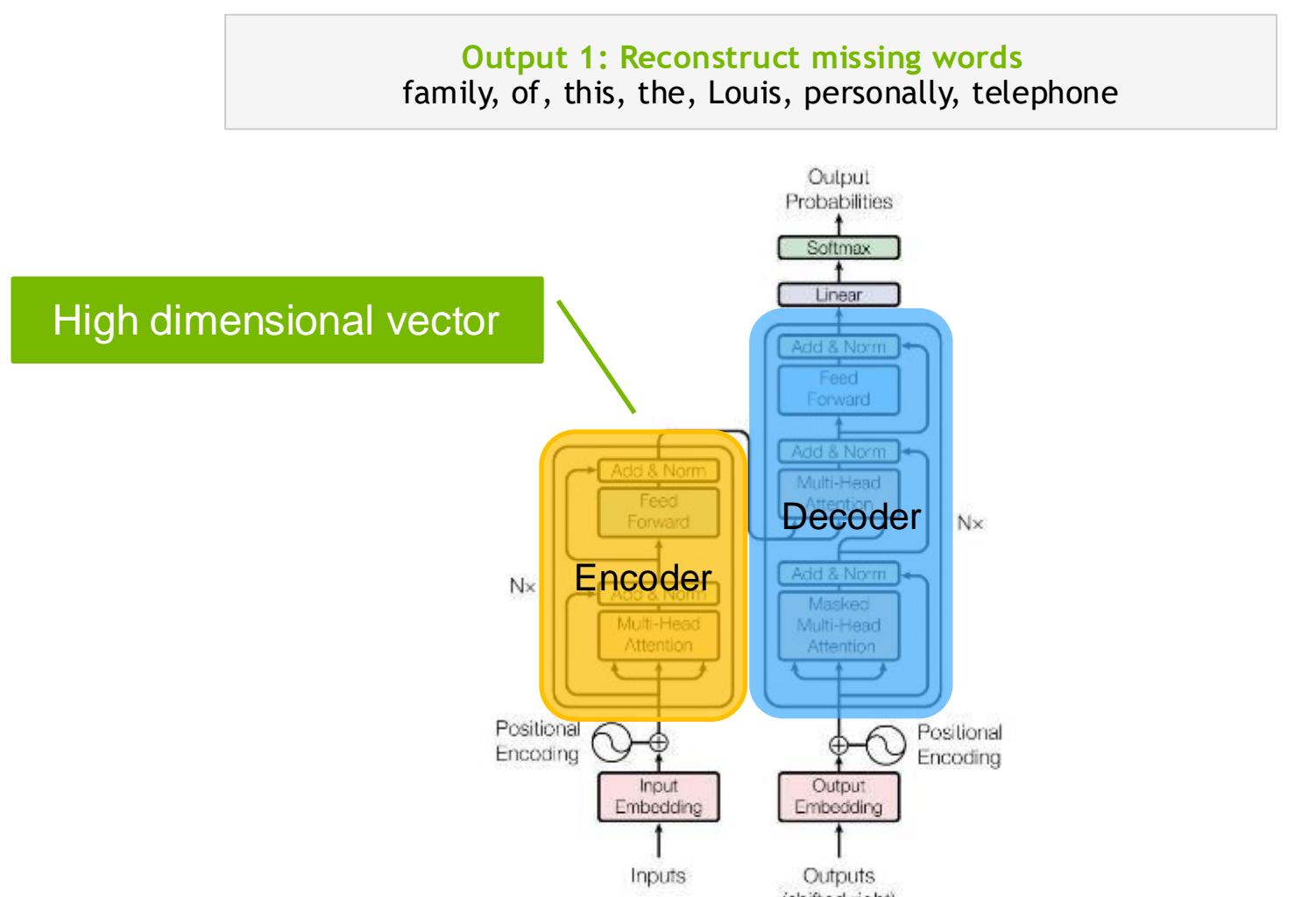
# LARGE MODELS ARE CHEAPER TO DESIGN

Impact of architectural hyperparameters

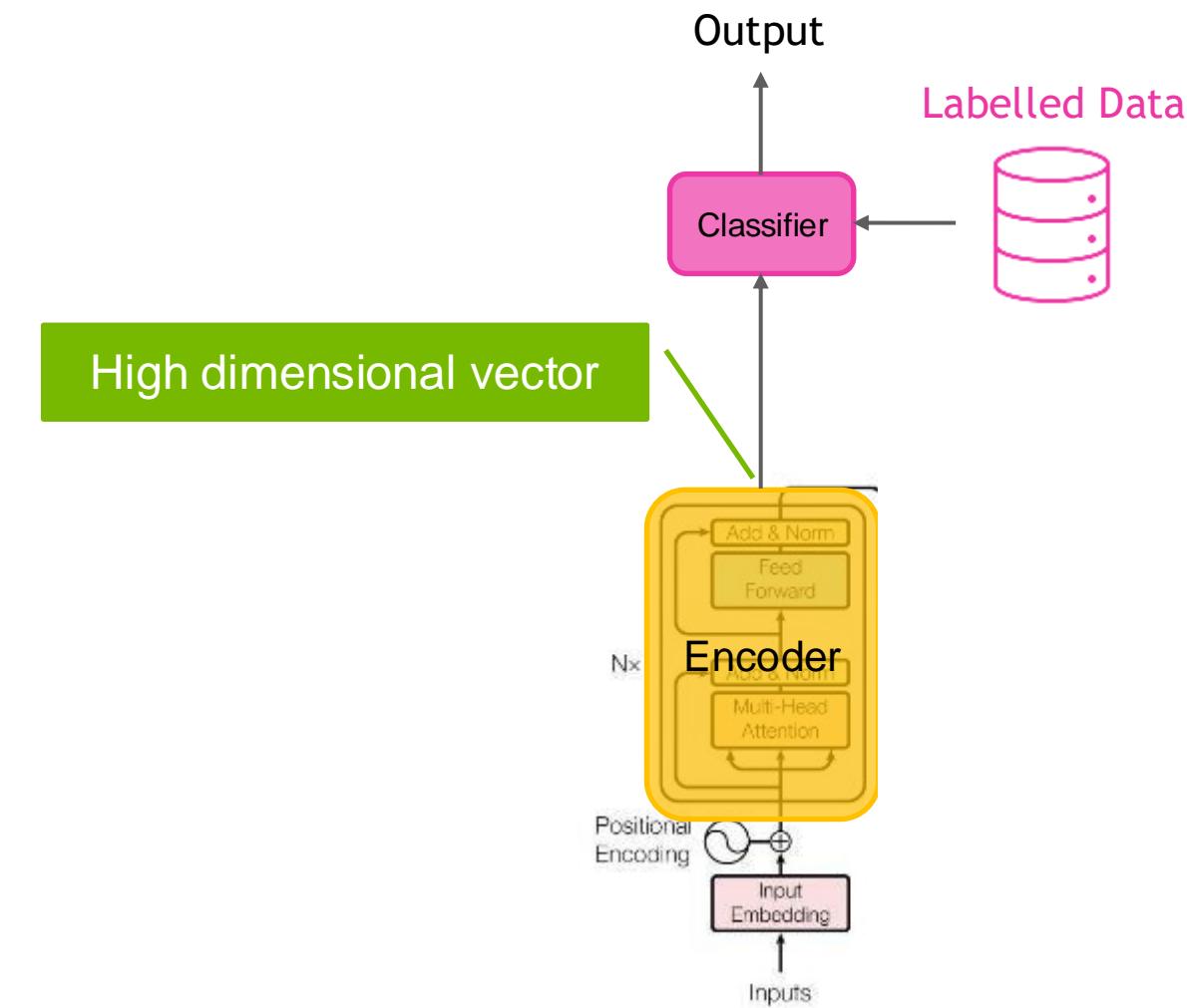
*“... more importantly, we find that the precise architectural hyperparameters are unimportant compared to the overall scale of the language model.”*

# NLP APPROACH (CIRCA 2019)

## Step 1: Pre-training a Transformer



## Step 2. Fine tune for a specific task



Input: Two sentences with 15% of words masked out

1 = “Initially he supported himself and his █ by farming on a plot █ family land.”

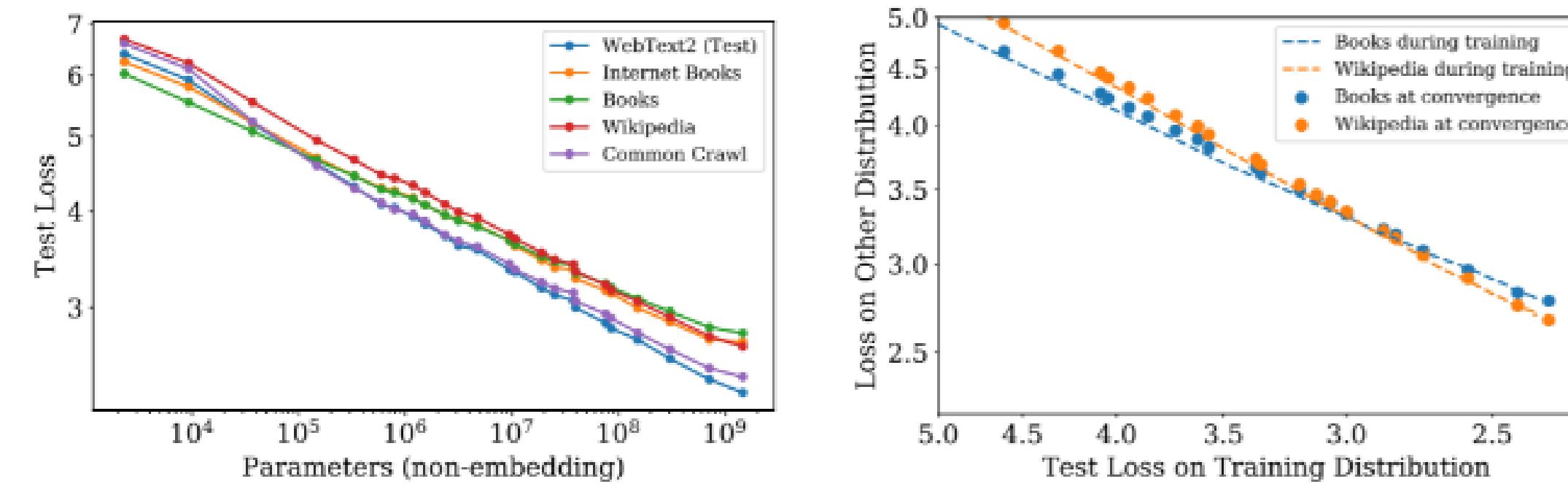
2 = “█ in turn attracted the attention of █ St. █ Post-Dispatch, which sent a reporter to Murray to █ review Stubblefield's wireless █.”



### 3. GENERALIZATION

# YES THEY CREATE INCREMENTAL IMPROVEMENT IN ACCURACY

Larger models generalize better



**Figure 8** **Left:** Generalization performance to other data distributions improves smoothly with model size, with only a small and very slowly growing offset from the WebText2 training distribution. **Right:** Generalization performance depends only on training distribution performance, and not on the phase of training. We compare generalization of converged models (points) to that of a single large model (dashed curves) as it trains.

# DOWNSTREAM TASKS

## Zero/Few Shot Learners

### Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

- 1 Translate English to French: ← task description
- 2 cheese => ..... ← prompt

### One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

- 1 Translate English to French: ← task description
- 2 sea otter => loutre de mer ← example
- 3 cheese => ..... ← prompt

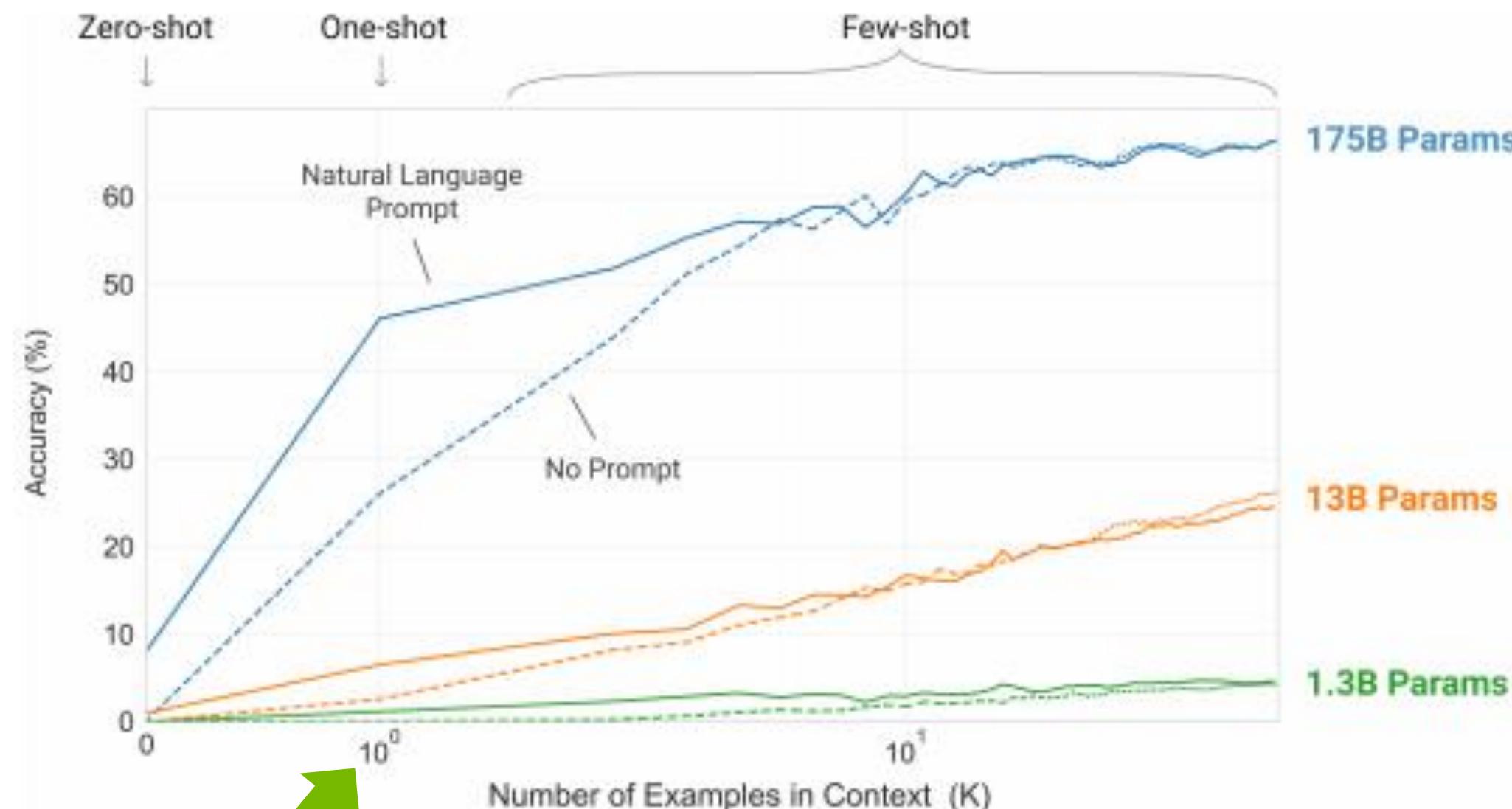
### Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

- 1 Translate English to French: ← task description
- 2 sea otter => loutre de mer ← examples
- 3 peppermint => menthe poivrée
- 4 plush girafe => girafe peluche
- 5 cheese => ..... ← prompt

# DOWNSTREAM TASKS

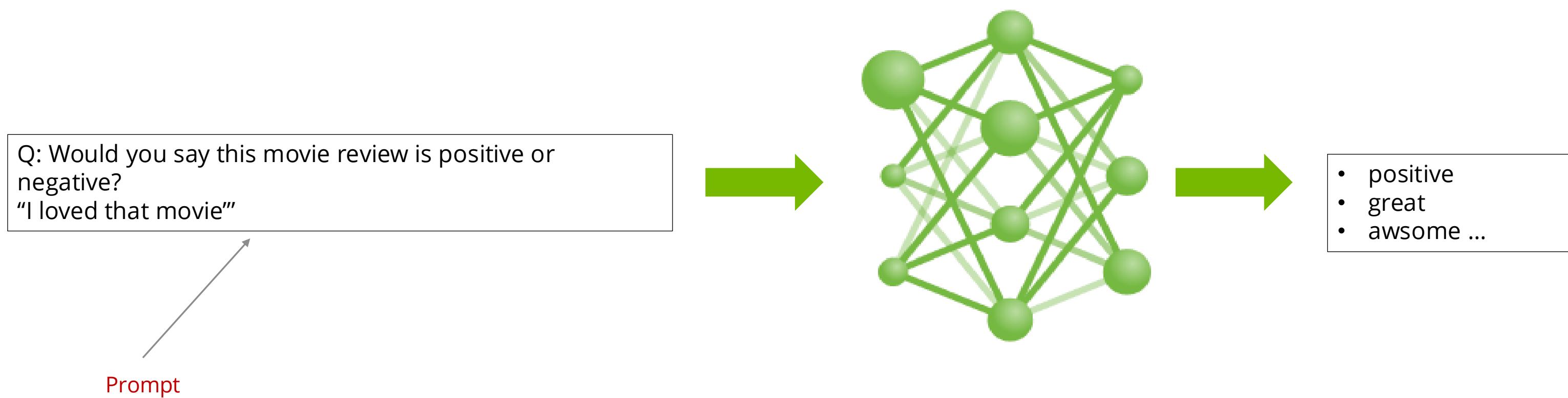
## Zero/Few Shot Learners



**Figure 1.2: Larger models make increasingly efficient use of in-context information.** We show in-context learning performance on a simple task requiring the model to remove random symbols from a word, both with and without a natural language task description (see Sec. 3.9.2). The steeper “in-context learning curves” for large models demonstrate improved ability to learn a task from contextual information. We see qualitatively similar behavior across a wide range of tasks.

# DOWNSTREAM TASKS

Zero/Few Shot Learners



# DOWNSTREAM TASKS

## Prompt Engineering

Type	Task	Input ([X])	Template	Answer([Y])
Text CLS	Sentiment	I love this movie.	[X] The movie is [Y]	great fantastic ...
	Topics	He prompted the LM.	[X] The text is about [Y]	sports science ...
	Intention	What is taxi fare to Denver?	[X] The question is about [Y]	quantity city ...
Text-span CLS	Aspect Sentiment	Poor service but good food.	[X] What about service? [Y]	Bad Terrible ...
Text-pair CLS	NLI	[X1]: An old man with ... [X2]: A man walks ...	Hypothesis: [X1], Premise: [X2], Answer: [Y]	Contradiction Entailment ...
Tagging	NER	[X1]: Mike went to Paris. [X2]: Paris	[X1] [X2] is a [Y]	Yes No ...
Text Generation	Summarization	Las Vegas police ...	[X] TL;DR: [Y]	The victim ... A woman ... ...
	Translation	Je vous aime.	French [X] English: [Y]	I love you. I fancy you. ...

Prompts				
manual	<u>DirectX is developed by</u> $y_{\text{man}}$	mined	<u><math>y_{\text{mine}}</math> released the DirectX</u>	
paraphrased	<u>DirectX is created by</u> $y_{\text{para}}$			
Top 5 predictions and log probabilities				
	$y_{\text{man}}$	$y_{\text{mine}}$	$y_{\text{para}}$	
1	Intel	-1.06	<u>Microsoft</u> -1.77	<u>Microsoft</u> -2.23
2	<u>Microsoft</u>	-2.21	They -2.43	Intel -2.30
3	IBM	-2.76	It -2.80	default -2.96
4	Google	-3.40	Sega -3.01	Apple -3.44
5	Nokia	-3.58	Sony -3.19	Google -3.45

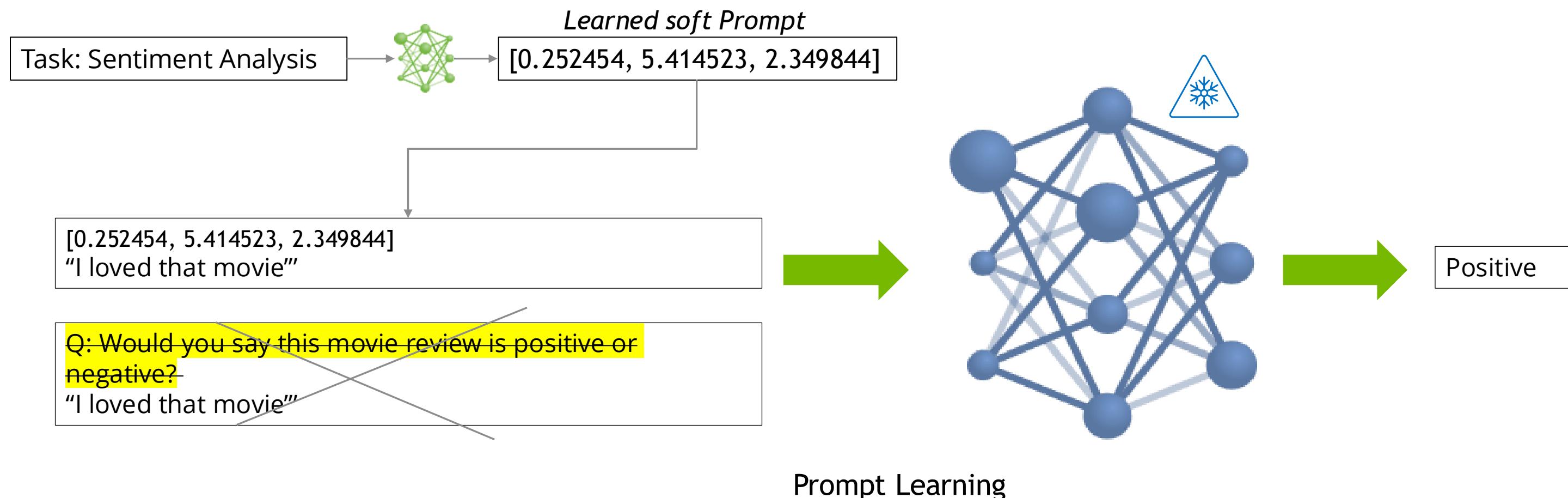
Figure 1: Top-5 predictions and their log probabilities using different prompts (manual, mined, and paraphrased) to query BERT. Correct answer is underlined.

ID	Modifications	Acc. Gain
P413	$x$ plays <del>in</del> <ins>→at</ins> $y$ position	+23.2
P495	$x$ was <del>created</del> <ins>→made</ins> in $y$	+10.8
P495	$x$ <del>was</del> <ins>→is</ins> created in $y$	+10.0
P361	$x$ is <del>a</del> part of $y$	+2.7
P413	$x$ plays <del>in</del> <ins>in</ins> $y$ position	+2.2

Table 6: Small modifications (~~update~~, insert, and ~~delete~~) in paraphrase lead to large accuracy gain (%).

# DOWNSTREAM TASKS

## Prompt Learning on a Small Training Dataset



# DOWNSTREAM TASKS

## Prompt Tuning / P-Tuning

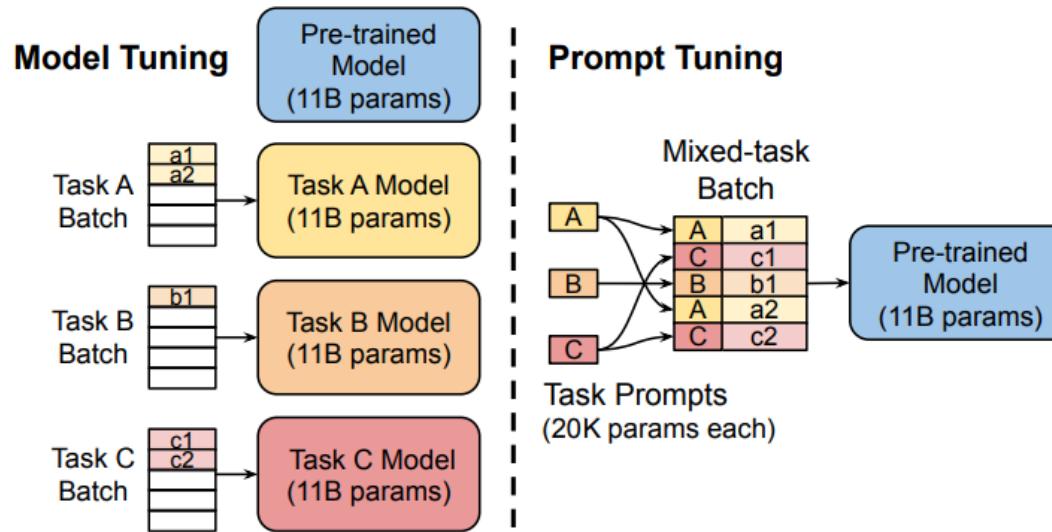
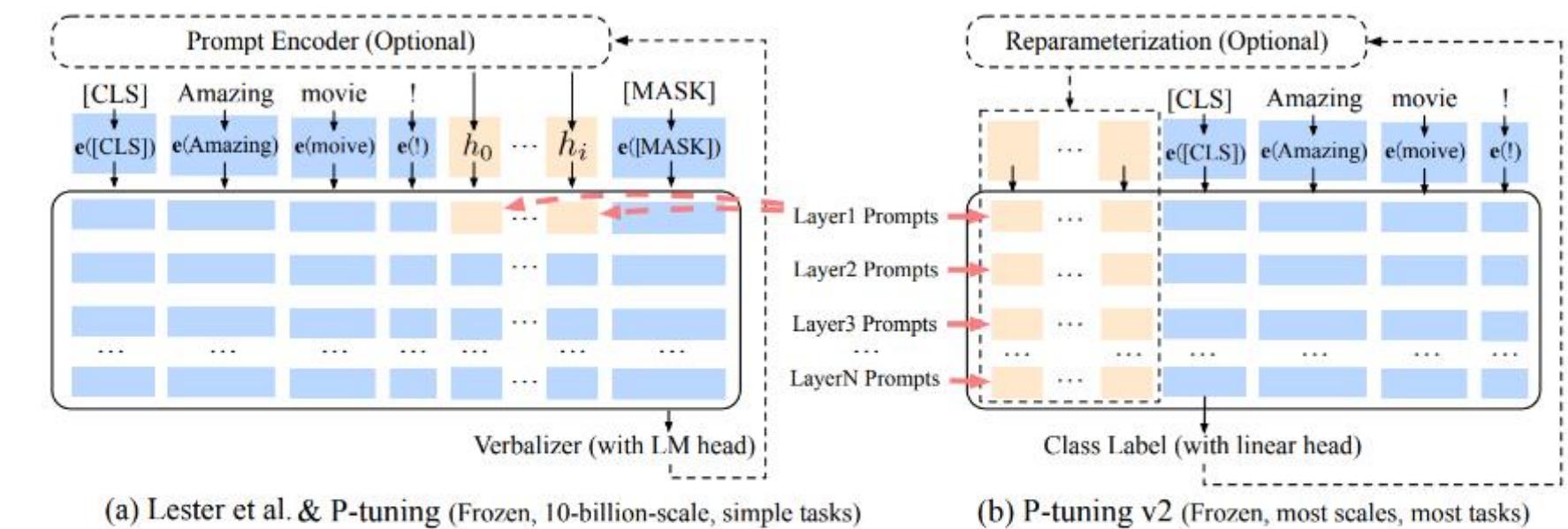


Figure 2: **Model tuning** requires making a task-specific copy of the entire pre-trained model for each downstream task and inference must be performed in separate batches. **Prompt tuning** only requires storing a small task-specific prompt for each task, and enables mixed-task inference using the original pre-trained model. With a T5 “XXL” model, each copy of the tuned model requires 11 billion parameters. By contrast, our tuned prompts would only require 20,480 parameters per task—a reduction of *over five orders of magnitude*—assuming a prompt length of 5 tokens.



(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks)

(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

(b) P-tuning v2 (Frozen, most scales, most tasks

# DOWNSTREAM TASKS

Customize Models using Parameter-efficient tuning | Adapters

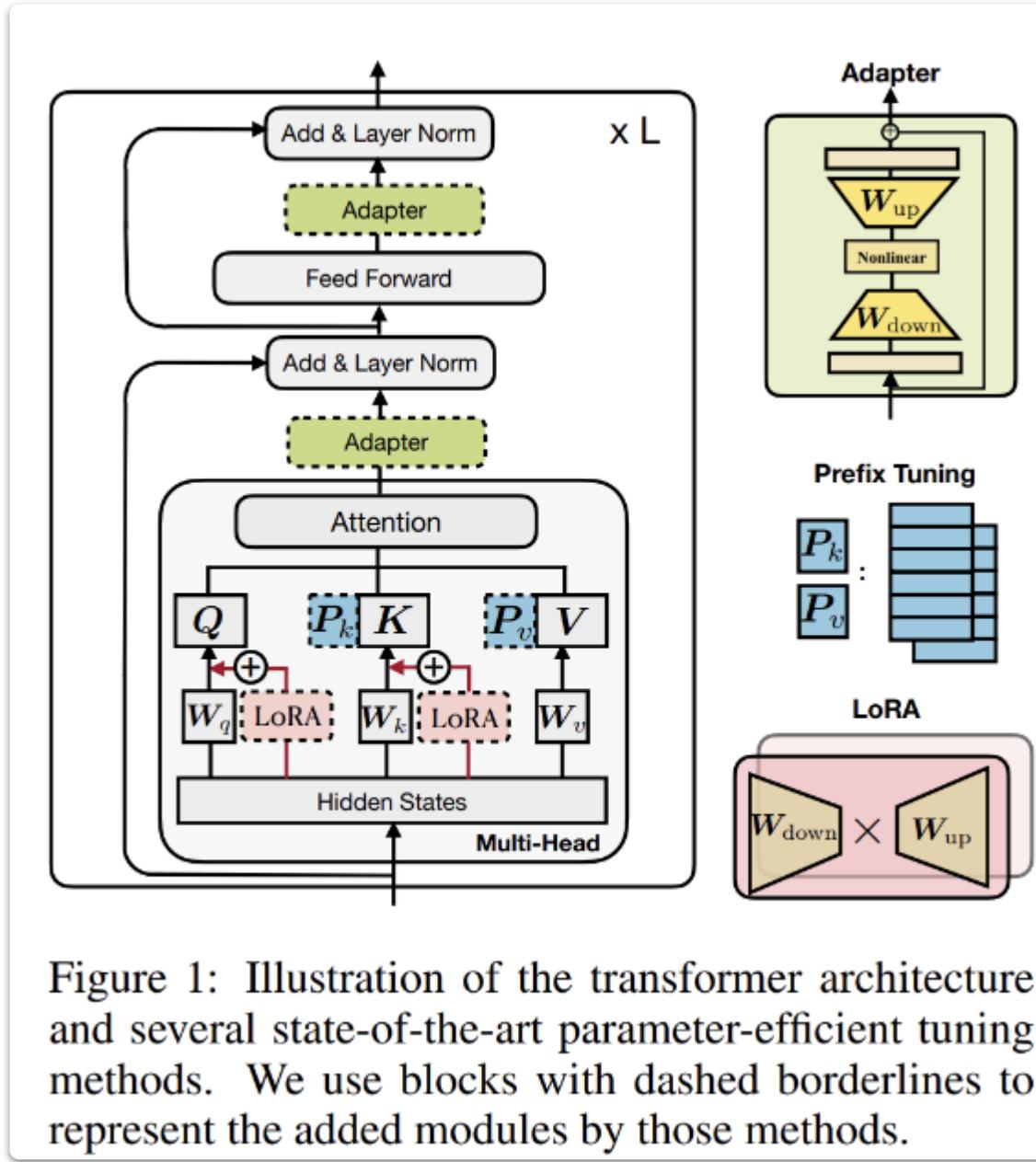


Figure 1: Illustration of the transformer architecture and several state-of-the-art parameter-efficient tuning methods. We use blocks with dashed borderlines to represent the added modules by those methods.

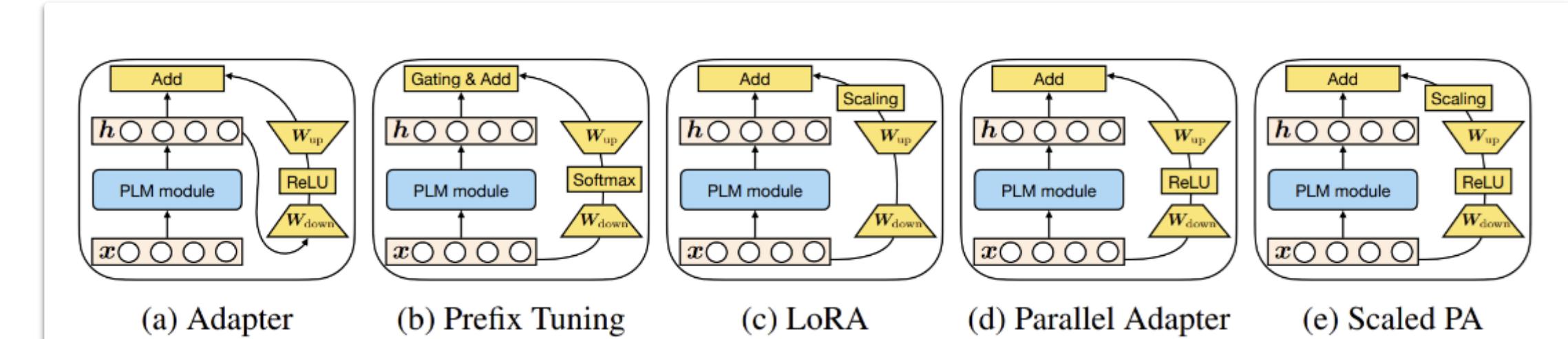


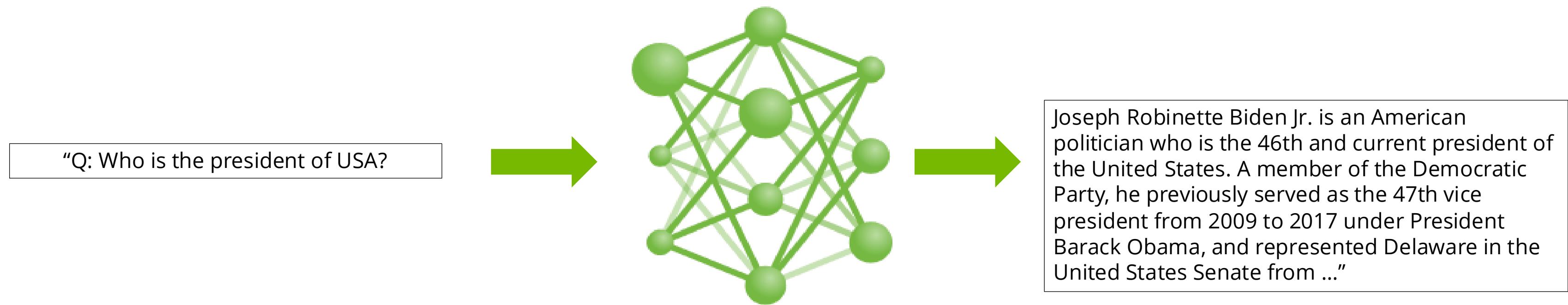
Figure 3: Graphical illustration of existing methods and the proposed variants. “PLM module” represents a certain sublayer of the PLM (e.g. attention or FFN) that is frozen. “Scaled PA” denotes scaled parallel adapter. We do not include multi-head parallel adapter here to save space.



INSTRUCTED LLM

# INSTRUCTIONS FINETUNING

Dataset of Instructions (Queries and Answers)



# INSTRUCTIONS FINETUNING

FLAN [Google] 

Published as a conference paper at ICLR 2022

## FINETUNED LANGUAGE MODELS ARE ZERO-SHOT LEARNERS

Jason Wei\*, Maarten Bosma\*, Vincenzo Gómez, Brian Lester, Nan Du, Andrew M. Dai  
Google Research

This paper explores a simple idea for improving zero-shot learning of language models. We show that finetuning a large language model on a collection of datasets designed to capture user intent substantially improves its zero-shot performance on unseen tasks.

We take a 137B parameter pre-trained model and finetune it over 60 NLP datasets verbalized by humans to evaluate this instruction-tuned model. Our approach, called FLAN, substantially improves zero-shot performance and surpasses zero-shot 175B GPT-3 on many benchmarks. FLAN outperforms few-shot GPT-3 10x faster and requires less memory. We evaluate our model on OpenbookQA, and StoryCloze, and show that it generalizes well across different datasets, model scale, and natural language processing tasks.

### LaMDA: Language Models for Dialog Applications

Romal Thoppilan Daniel De Freitas \* Jamie Hall Noam Shazeer \* Apoorv Kulshreshtha  
Heng-Tze Cheng Alicia Jin Taylor Bos Leslie Baker Yu Du YaGuang Li Hongrae Lee  
Huaixiu Steven Zheng Amin Ghafouri Marcelo Menegali Yanping Huang Maxim Krikun  
Dmitry Lepikhin James Qin Dehao Chen Yuanzhong Xu Zhifeng Chen Adam Roberts  
Maarten Bosma Vincent Zhao Yanqi Zhou Chung-Ching Chang Igor Krivokon Will Rusch  
Marc Pickett Pranesh Srinivasan Laichee Man Kathleen Meier-Hellstern  
Meredith Ringel Morris Tulsee Doshi Renelito Delos Santos Toju Duke Johnny Soraker  
Ben Zevenbergen Vinodkumar Prabhakaran Mark Diaz Ben Hutchinson Kristen Olson  
Alejandra Molina Erin Hoffman-John Josh Lee Lora Aroyo Ravi Rajakumar  
Alena Butryna Matthew Lamm Viktoriya Kuzmina Joe Fenton Aaron Cohen  
Rachel Bernstein Ray Kurzweil Blaise Aguera-Arcas Claire Cui Marian Croak Ed Chi  
Quoc Le  
Google

#### Abstract

We present LaMDA: Language Models for Dialog Applications. LaMDA is a family of Transformer-based neural language models specialized for dialog, which have up to 137B parameters and are pre-trained on 1.56T words of public dialog data and web text. While model scaling alone can improve quality, it shows less improvements on safety and factual grounding. We demonstrate that fine-tuning with annotated data and enabling the model to consult external knowledge sources can lead to significant improvements towards the two key challenges of safety and factual grounding. The first challenge, safety, involves ensuring that the model’s responses are consistent with a set of human values, such as preventing harmful suggestions and unfair bias. We quantify safety using a metric based on an illustrative set of human values, and we find that filtering candidate responses using a LaMDA classifier fine-tuned with a small amount of crowdworker-annotated data offers a promising approach to improving model safety. The second challenge, factual grounding, involves enabling the model to consult external knowledge sources, such as an information retrieval system, a language translator, and a calculator. We quantify factuality using a groundedness metric, and we find that our approach enables the model to generate responses grounded in known sources, rather than responses that merely sound plausible. Finally, we explore the use of LaMDA in the domains of education and content recommendations, and analyze their helpfulness and role consistency.

LaMDA [Google] 

InstructGPT [OpenAI] 

## Training language models to follow instructions with human feedback

Long Ouyang\* Jeff Wu\* Xu Jiang\* Diogo Almeida\* Carroll L. Wainwright\*

Pamela Mishkin\* Chong Zhang Sandhini Agarwal Katarina Slama Alex Ray

John Schulman Jacob Hilton Fraser Kelton Luke Miller Maddie Simens

Amanda Askell† Peter Welinder Paul Christiano\*†

Jan Leike\* Ryan Lowe\*

OpenAI

#### Abstract

Making language models bigger does not inherently make them better at following a user’s intent. For example, large language models can generate outputs that are untruthful, toxic, or simply not helpful to the user. In other words, these models are not *aligned* with their users. In this paper, we show an avenue for aligning language models with user intent on a wide range of tasks by fine-tuning with human feedback. Starting with a set of labeler-written prompts and prompts submitted through the OpenAI API, we collect a dataset of labeler demonstrations of the desired model behavior, which we use to fine-tune GPT-3 using supervised learning. We then collect a dataset of rankings of model outputs, which we use to further fine-tune this supervised model using reinforcement learning from human feedback. We call the resulting models *InstructGPT*. In human evaluations on our prompt distribution, outputs from the 1.3B parameter InstructGPT model are preferred to outputs from the 175B GPT-3, despite having 100x fewer parameters. Moreover, InstructGPT models show improvements in truthfulness and reductions in toxic output generation while having minimal performance regressions on public NLP datasets. Even though InstructGPT still makes simple mistakes, our results show that fine-tuning with human feedback is a promising direction for aligning language models with human intent.

# INSTRUCTIONS FINETUNING

FLAN [Google]



]

Published as a conference paper at ICLR 2022

## FINETUNED LANGUAGE MODELS ARE ZERO-SHOT LEARNERS

Jason Wei\*, Maarten Bosma\*, Vincenzo Gómez, Brian Lester, Nan Du, Andrew M. Dai  
Google Research

### LaMDA: Language Models for Dialog Applications

Romal Thoppilan Daniel De Freitas \* Jamie Hall Noam Shazeer \* Apoorv Kulshreshtha  
Heng-Tze Cheng Alicia Jin Taylor Bos Leslie Baker Yu Du YaGuang Li Hongrae Lee  
Huaixiu Steven Zheng Amin Ghafouri Marcelo Menegali Yanping Huang Maxim Krikun  
Dmitry Lepikhin James Qin Dehao Chen Yuanzhong Xu Zhifeng Chen Adam Roberts  
Maarten Bosma Vincent Zhao Yanqi Zhou Chung-Ching Chang Igor Krivokon Will Rusch  
Marc Pickett Pranesh Srinivasan Laichee Man Kathleen Meier-Hellstern  
Meredith Ringel Morris Tulsee Doshi Renelito Delos Santos Toju Duke Johnny Soraker  
Ben Zevenbergen Vinodkumar Prabhakaran Mark Diaz Ben Hutchinson Kristen Olson  
Alejandra Molina Erin Hoffman-John Josh Lee Lora Aroyo Ravi Rajakumar  
Alena Butryna Matthew Lamm Viktoriya Kuzmina Joe Fenton Aaron Cohen  
Rachel Bernstein Ray Kurzweil Blaise Aguera-Arcas Claire Cui Marian Croak Ed Chi  
Quoc Le

Google

#### Abstract

We present LaMDA: Language Models for Dialog Applications. LaMDA is a family of Transformer-based neural language models specialized for dialog, which have up to 137B parameters and are pre-trained on 1.56T words of public dialog data and web text. While model scaling alone can improve quality, it shows less improvements on safety and factual grounding. We demonstrate that fine-tuning with annotated data and enabling the model to consult external knowledge sources can lead to significant improvements towards the two key challenges of safety and factual grounding. The first challenge, safety, involves ensuring that the model's responses are consistent with a set of human values, such as preventing harmful suggestions and unfair bias. We quantify safety using a metric based on an illustrative set of human values, and we find that filtering candidate responses using a LaMDA classifier fine-tuned with a small amount of crowdworker-annotated data offers a promising approach to improving model safety. The second challenge, factual grounding, involves enabling the model to consult external knowledge sources, such as an information retrieval system, a language translator, and a calculator. We quantify factuality using a groundedness metric, and we find that our approach enables the model to generate responses grounded in known sources, rather than responses that merely sound plausible. Finally, we explore the use of LaMDA in the domains of education and content recommendations, and analyze their helpfulness and role consistency.

LaMDA [Google]



InstructGPT [OpenAI]

]

## Training language models to follow instructions with human feedback

Long Ouyang\* Jeff Wu\* Xu Jiang\* Diogo Almeida\* Carroll L. Wainwright\*

Pamela Mishkin\* Chong Zhang Sandhini Agarwal Katarina Slama Alex Ray

John Schulman Jacob Hilton Fraser Kelton Luke Miller Maddie Simens

Amanda Askell† Peter Welinder Paul Christiano\*†

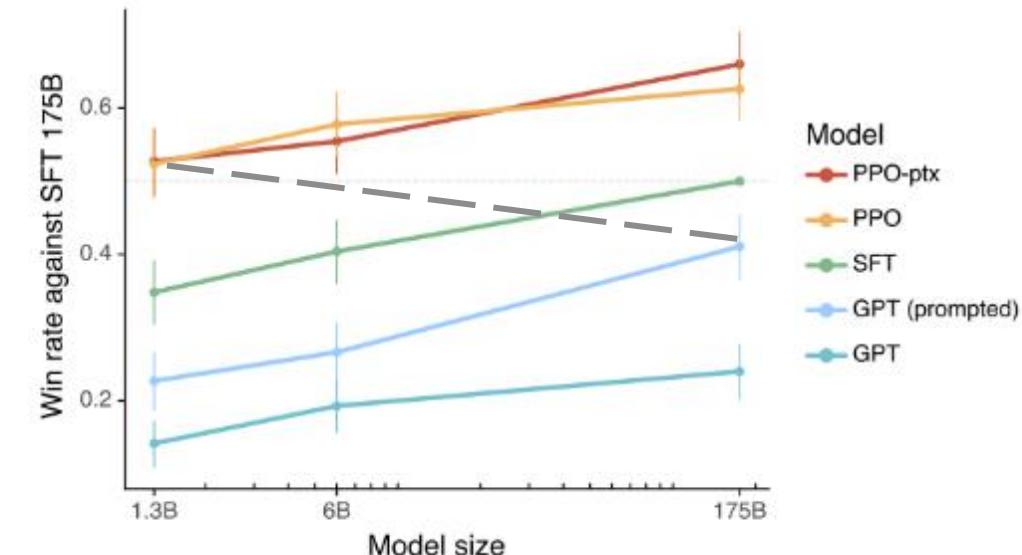


Figure 1: Human evaluations of various models on our API prompt distribution, evaluated by how often outputs from each model were preferred to those from the 175B SFT model. Our InstructGPT models (PPO-ptx) as well as its variant trained without pretraining mix (PPO) significantly outperform the GPT-3 baselines (GPT, GPT prompted); outputs from our 1.3B PPO-ptx model are preferred to those from the 175B GPT-3. Error bars throughout the paper are 95% confidence intervals.

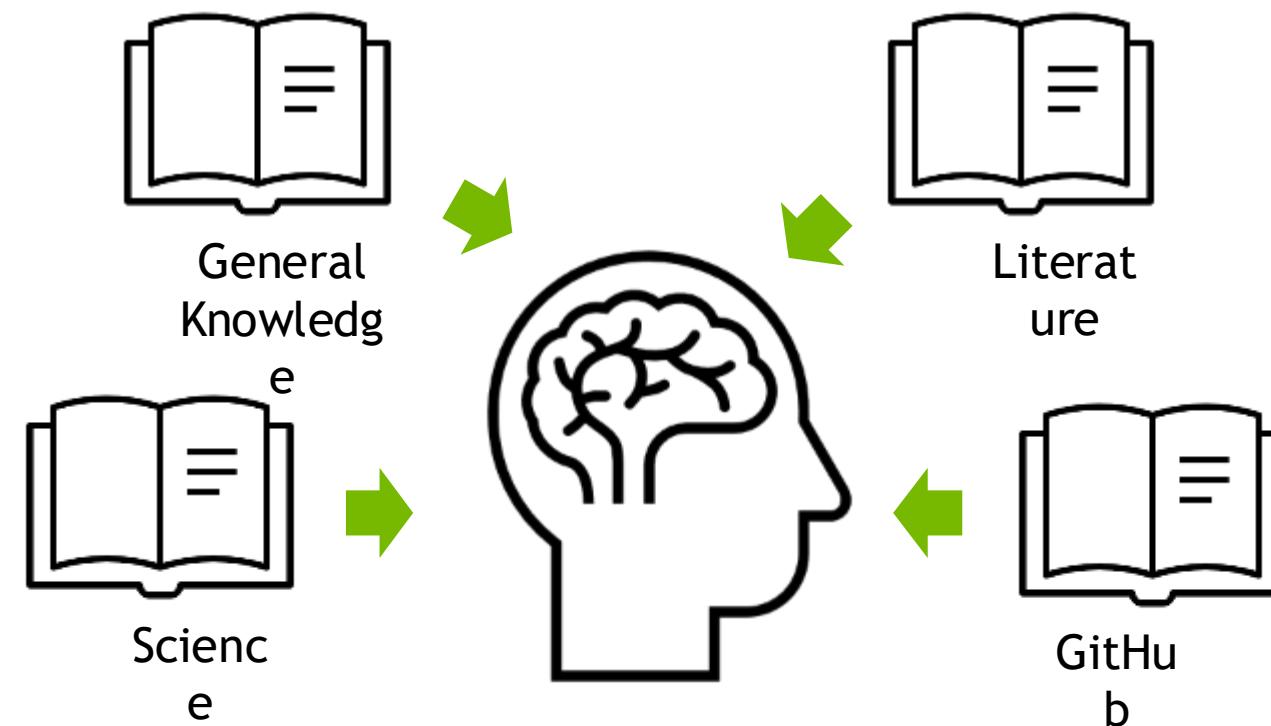
language models with human intent.



# CHANGE IN THE NLP PARADIGM

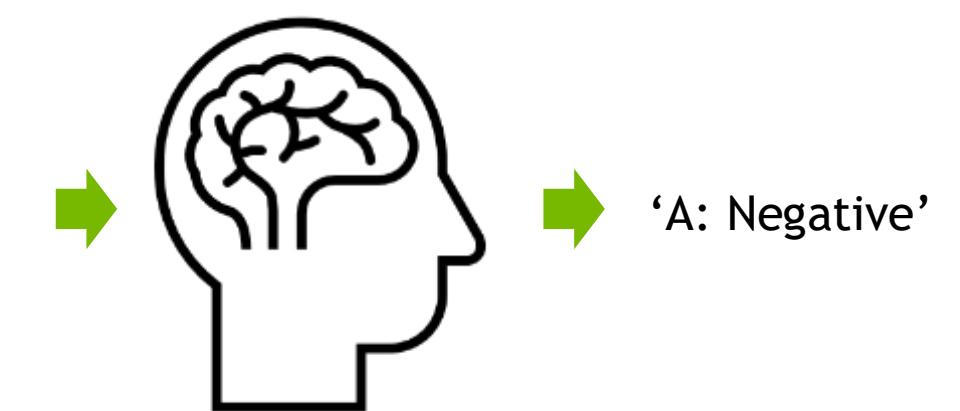
# NEW NLP APPROACH (CIRCA 2021)

Step 1: Train a Very Deep/HUGE model



Step 2. Ask questions

'Q: Would you say this movie  
review is positive or negative?  
“I loved that movie”'



Huge means Billions of parameters

# TOWARDS GENERAL INTELLIGENCE

## Old way

- ★ Needs Labelled data
  - Cost of data collection/labelling
  - Legal/Privacy concerns around using data
- ★ 1 model per task results in
  - Increased model development/tuning cost
  - Increased operational costs
  - Increased money spent on sourcing data
- ★ Relatively Limited generalization
- ★ Computationally cheaper (~300 Million parameters)

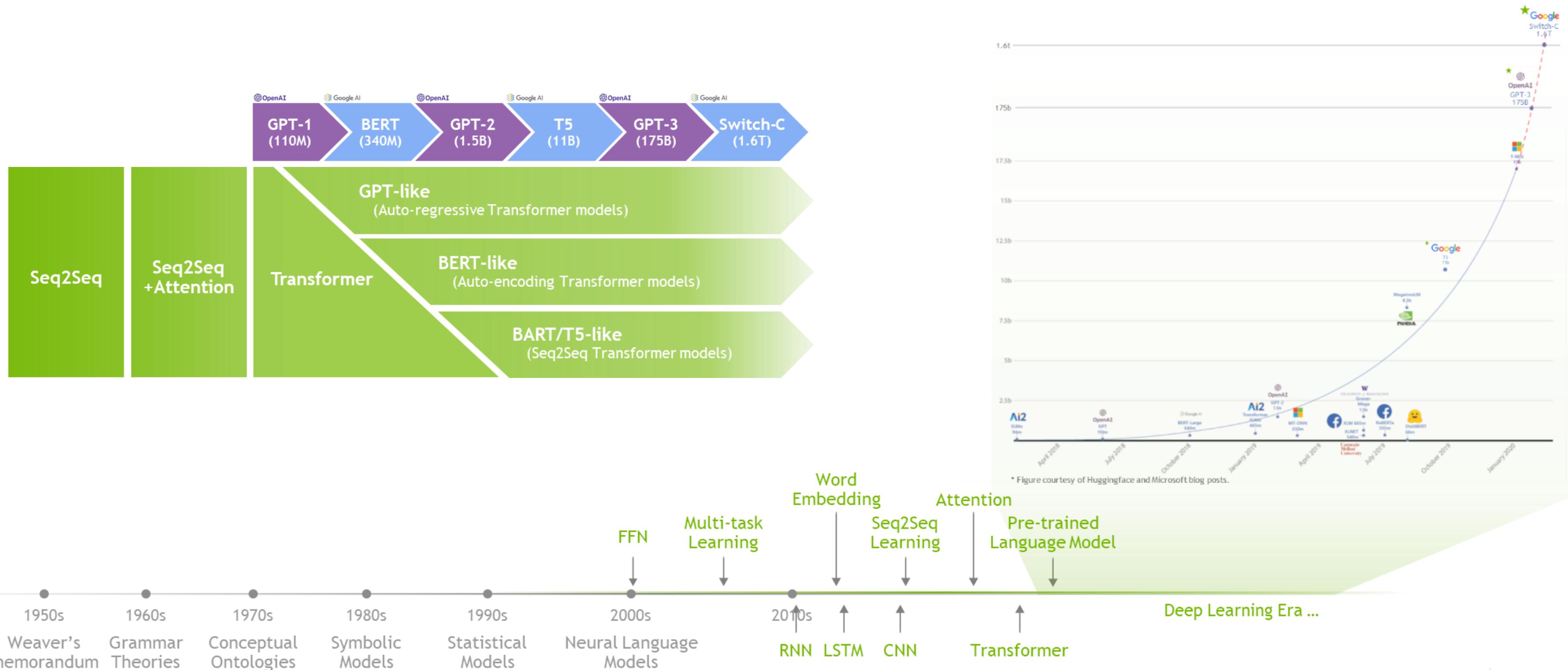
## New way

- ★ Does not need labelled data
- ★ Single generic model can do more than one tasks
- ★ More generalized: in addition to language also learns higher level concepts, styles, etc.
- ★ Computationally Expensive (~500 Billion parameters)

Leveraging more compute to get a general model without significant data/labelling cost

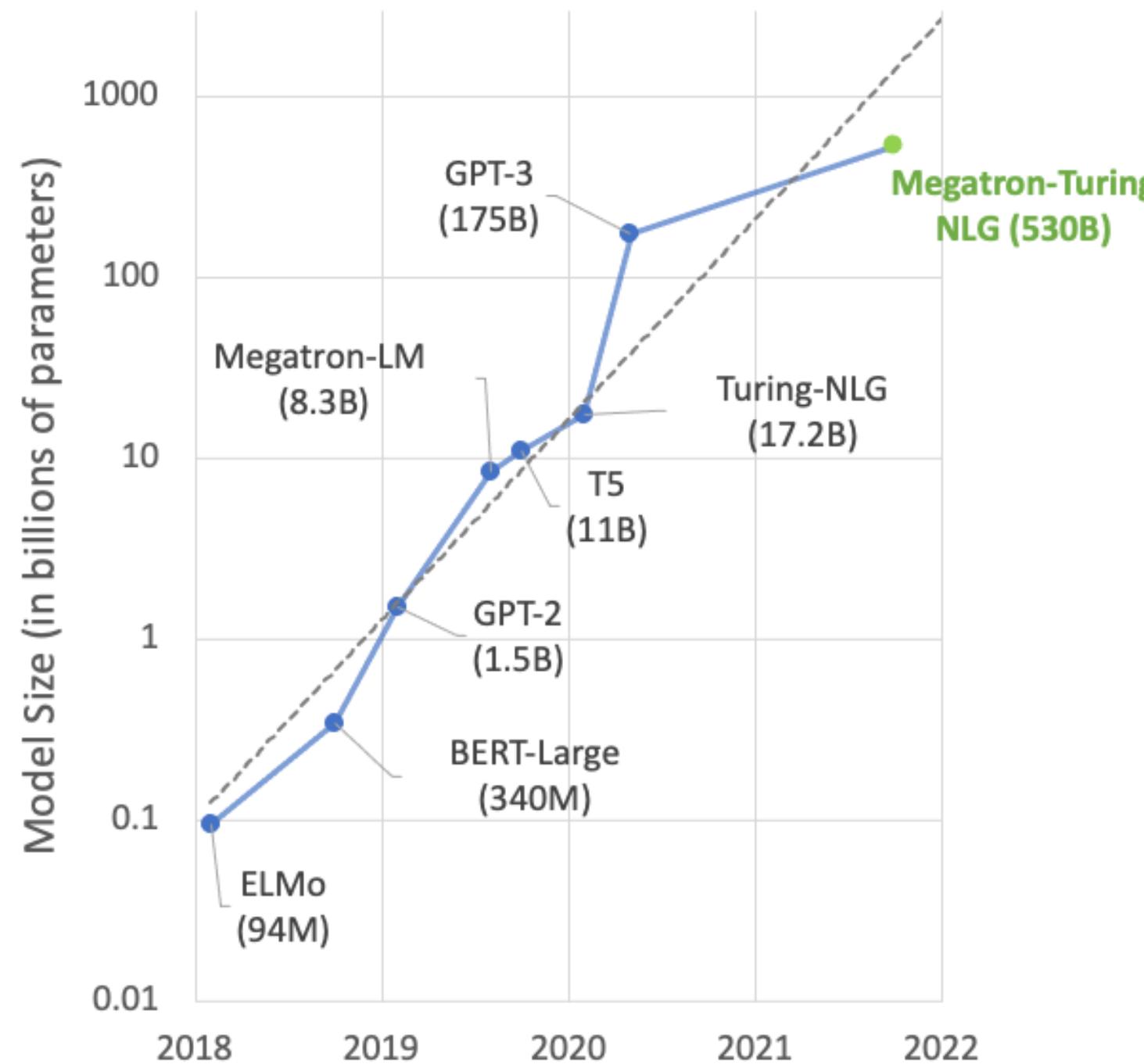
# HISTORY OF LANGUAGE MODELS

Language Model became more complex and larger



# MEGATRON-TURING NLG 530B

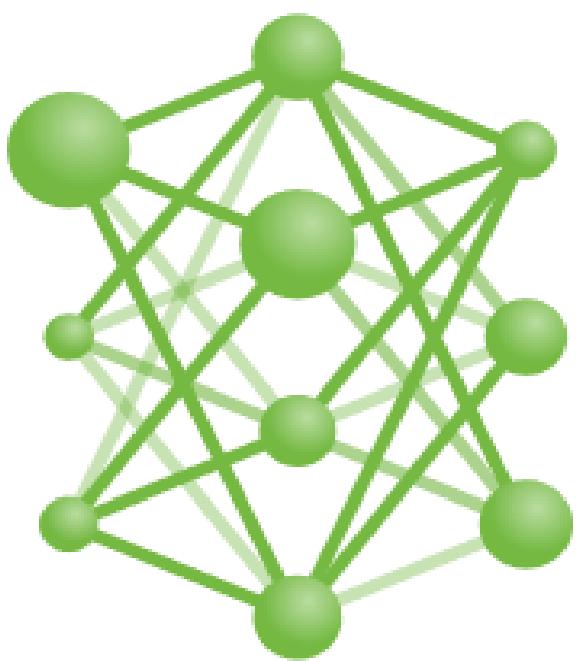
## The Trend Continues



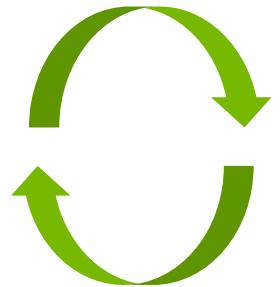


LLM FOR SEARCH

# EXTERNAL KNOWLEDGE SOURCE



Instructed LLM



Knowledge source

# EXTERNAL KNOWLEDGE SOURCE

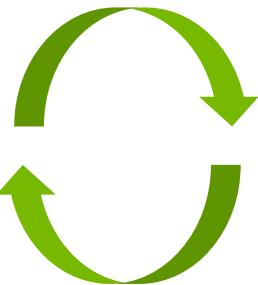
Searching through the web

A Can you formulate a search query that will help me find the above mentioned information?

Certainly! Here's a search query that could help you find more information about synthesizing dimethyl carbonate (DMC) from raw materials:

"synthesis of dimethyl carbonate from methanol and phosgene"

This query includes the relevant keywords for the process (synthesis, dimethyl carbonate, methanol, phosgene) and can help you find articles, research papers, or other sources that discuss the topic in more detail. You may also want to refine your search with additional keywords related to the catalyst, reaction conditions, and purification techniques, depending on the level of detail you're looking for. As always, be sure to evaluate the sources you find for accuracy and reliability.



Microsoft Bing

"synthesis of dimethyl carbonate from methanol and phosgene"

ALL WORK IMAGES VIDEOS MAPS NEWS CHAT MORE

About 4 results Date ▾

**Studies on synthesis of dimethyl carbonate from ...**

<https://www.sciencedirect.com/science/article/pii/S156673670800486X>  
Web Jan 25, 2009 · Thermodynamic and kinetic character of DMC catalytic synthesis process  
3.2.1. Synthesis reaction catalyzed by basic catalysts When the reaction of methanol and carbon dioxide occur using basic compounds, namely K<sub>2</sub>CO<sub>3</sub>, KOH and CH<sub>3</sub> ...  
Author: Qinghai Cai, Qinghai Cai, Bin Lu, ... Publish Year: 2009

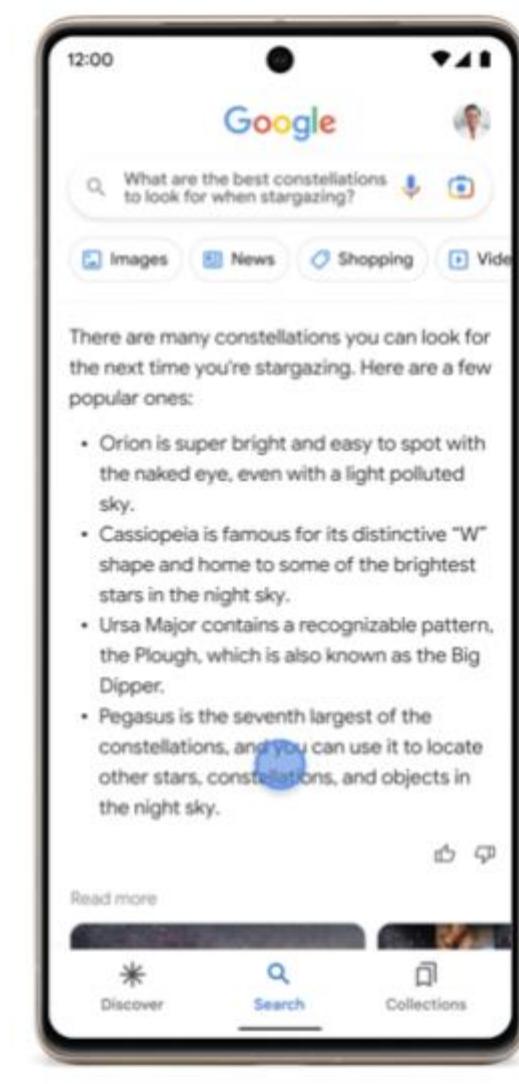
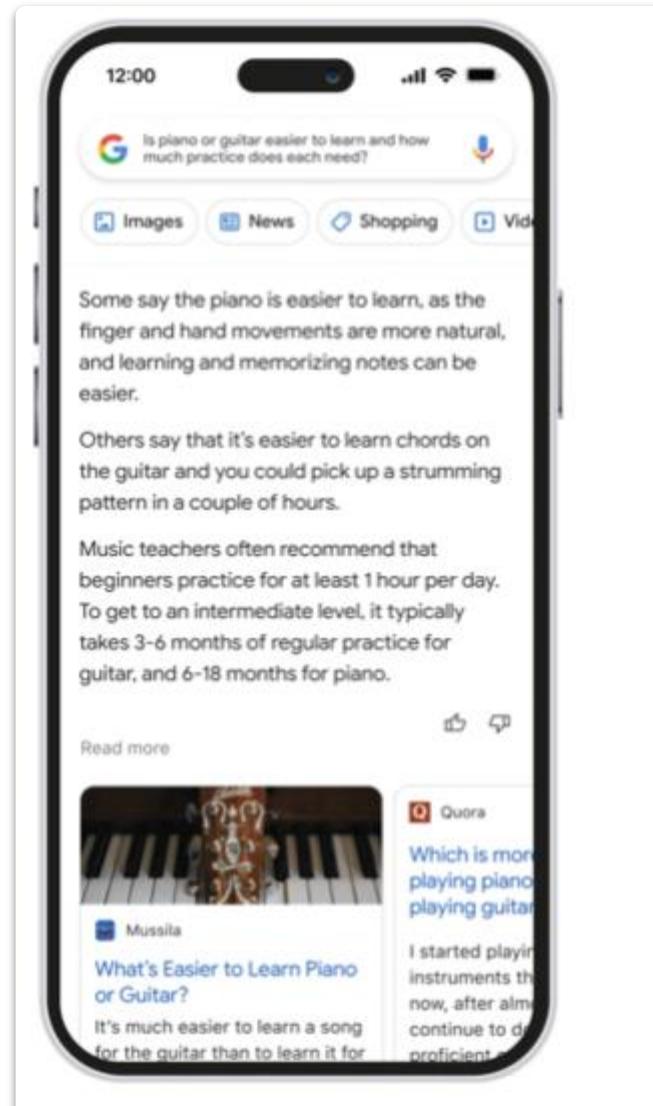
A Given the following text: "The reaction thermodynamics of dimethyl carbonate (DMC) synthesis from CH<sub>3</sub>OH and CO<sub>2</sub> was investigated by estimating ΔH and ΔG of the reaction at different temperatures and pressure. The result shows that the reaction does not spontaneously occur under finite conditions in view of thermodynamics. Thus, in order to seek a new method or a novel idea for the DMC synthesis, the thermodynamics of DMC synthesis catalyzed by basic catalysts, K<sub>2</sub>CO<sub>3</sub>, KOH and CH<sub>3</sub>OK, was discussed in details. A general conclusion was obtained that the subroutine nesting of a coupling reaction is an effective avenue to increase the yield of DMC synthesis from CO<sub>2</sub> and CH<sub>3</sub>OH."

How do you increase the yield of DMC?

According to the given text, the most effective way to increase the yield of dimethyl carbonate (DMC) synthesis from CO<sub>2</sub> and CH<sub>3</sub>OH is through the use of a subroutine nesting of a coupling reaction, which was found to be an effective avenue for improving the reaction yield. Additionally, the text suggests that the thermodynamics of DMC synthesis catalyzed by basic catalysts, such as K<sub>2</sub>CO<sub>3</sub>, KOH, and CH<sub>3</sub>OK, were also discussed in detail, so it's possible that other catalysts or reaction conditions could be used to optimize the reaction yield as well. However, the text does not provide specific details on how the coupling reaction method works, so further research would be needed to explore this approach in more detail.

# EXTERNAL KNOWLEDGE SOURCE

Examples of Search Engine Powered by LLM



This screenshot shows a Microsoft Bing search result for the query "I am planning a trip for our anniversary in September. What are some good destinations?". The search bar at the top has a blue arrow pointing left and a green arrow pointing down. The results page shows a snippet from tripsavvy.com with a list of travel ideas. Below it, there's another snippet from starctmag.com about romantic destinations. A sidebar on the right contains a message from Sam, a AI-generated response, and a "People also ask" section.

**Sam** 33 111 33

I am planning a trip for our anniversary in September. What are some good destinations?

ALL SHOPPING IMAGES VIDEOS MAPS CHAT MORE

About 86,800,000 results Date ▾

tripsavvy.com · Jun 26, 2019

**The Best September Anniversary Travel Ideas - TripSavvy.**

- 1. Make Your Way to Maine. Winter comes early to ...
- 2. Spend September in San Juan. San Juan has ...
- 3. Book a Trip to Prague. Cooler weather and sparser ...
- 4. Choose Buenos Aires. Known as the Paris of South ...

See full list on tripsavvy.com

<https://www.tripsavvy.com/best-september-anniversary-travel-ideas-4173621>

feedback

**People also ask**

Congratulations on your anniversary! 🎉 There are many places you can go that are within a 3 hour flight from London Heathrow. Here are some suggestions based on your preferences and the best destinations in Europe in September [4](#) [5](#) [6](#):

- If you like **beaches and sunshine**, you can fly to **Malaga** in Spain [1](#), which has a pleasant climate, a lively culture, and a beautiful coastline. You can enjoy the sandy beaches, the historic monuments, and the delicious tapas. Malaga is also a great base to explore other nearby attractions, such as Granada, Seville, and Cordoba.
- If you like **mountains and lakes**, you can fly to **Annecy** in France [4](#), which is a charming alpine town near the Swiss border. Annecy is known as the "Venice of France" because of its canals and bridges. You can admire the stunning scenery, visit the medieval castle, and try some local cheese and wine. Annecy is also close to Geneva, Chamonix, and Lyon, if you want to

NORA - No One Right Answer [Google]

New Bing Search Engine [Microsoft]



LLM FOR CHATBOTS

# CHATBOTS POWERED BY INSTRUCTED LLM

## ChatGPT is a sibling model to InstructGPT

### 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

Free Research Preview: ChatGPT is optimized for dialogue. Our goal is to make AI systems more natural to interact with, and your feedback will help us improve our systems and make them even more useful.

A Who is the president of the United States?

As of my knowledge cutoff of September 2021, the President of the United States was Joe Biden.

A Could you tell me where he was born?

Certainly! Joe Biden was born in Scranton, Pennsylvania, United States.

A Could you tell me about his early career?

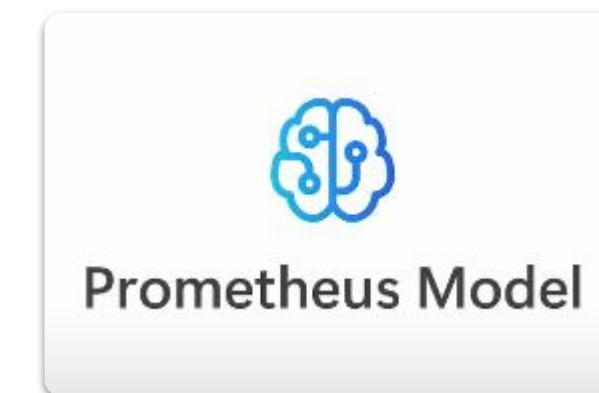
Yes, I'd be happy to give you a brief overview of Joe Biden's early career.  
After graduating from law school in 1968, Biden worked as an attorney in Wilmington, Delaware, before being elected to the New Castle County Council in 1970. In 1972, at the age of 29, he was elected to the United States Senate, becoming one of the youngest senators in American history.  
During his time in the Senate, Biden served on several committees, including the Foreign Relations Committee and the Judiciary Committee, and became known for his work on issues related to crime, drugs, and foreign policy. In 2008, he was elected Vice President of the United States under President Barack Obama, serving two terms in that role before being elected President in 2020.

# CHATBOTS POWERED BY INSTRUCTED LLM

Bing Chat powered by Next generation GPT | Bard powered by LaMDA



This screenshot shows the Microsoft Bing Chat interface. At the top, there are tabs for "SEARCH" and "CHAT", with "CHAT" being active. A user message in a blue box asks about anniversary travel options from London Heathrow. Below, a response suggests destinations like Malaga, Annecy, and Florence. A feedback section at the bottom right includes a "Feedback" button.



## Introducing Bard, an experimental conversational AI service powered by LaMDA

This screenshot shows the Bard AI service interface. A user asks about the pros and cons of buying an electric car. Bard responds with a list of pros (zero emissions, cheaper to operate) and cons (limited driving range, higher initial cost). Below the text are "Like", "Dislike", "Report", and "Check it" buttons. A feedback section at the bottom right includes a "Feedback" button.



TECHNICALLY, CAN WE  
BUILD LARGER MODELS?

# GOING BIGGER

## The challenge

- If we only consider Parameters, Gradients, and Optimizer states and ignore activations
- If we use FP16 data representation (so two bytes)
- If we use Adam as an optimizer (storing twelve bytes per parameter in mixed precision mode)
- If we consider a model with one billion parameters

$$10^9 * ( 2B + 2B + 12B) = 10^9 * 16B = 14.90GB$$

1 billion parameters      2 bytes per parameter      2 bytes per gradient      12 bytes per optimizer state

# GOING BIGGER

## The challenge

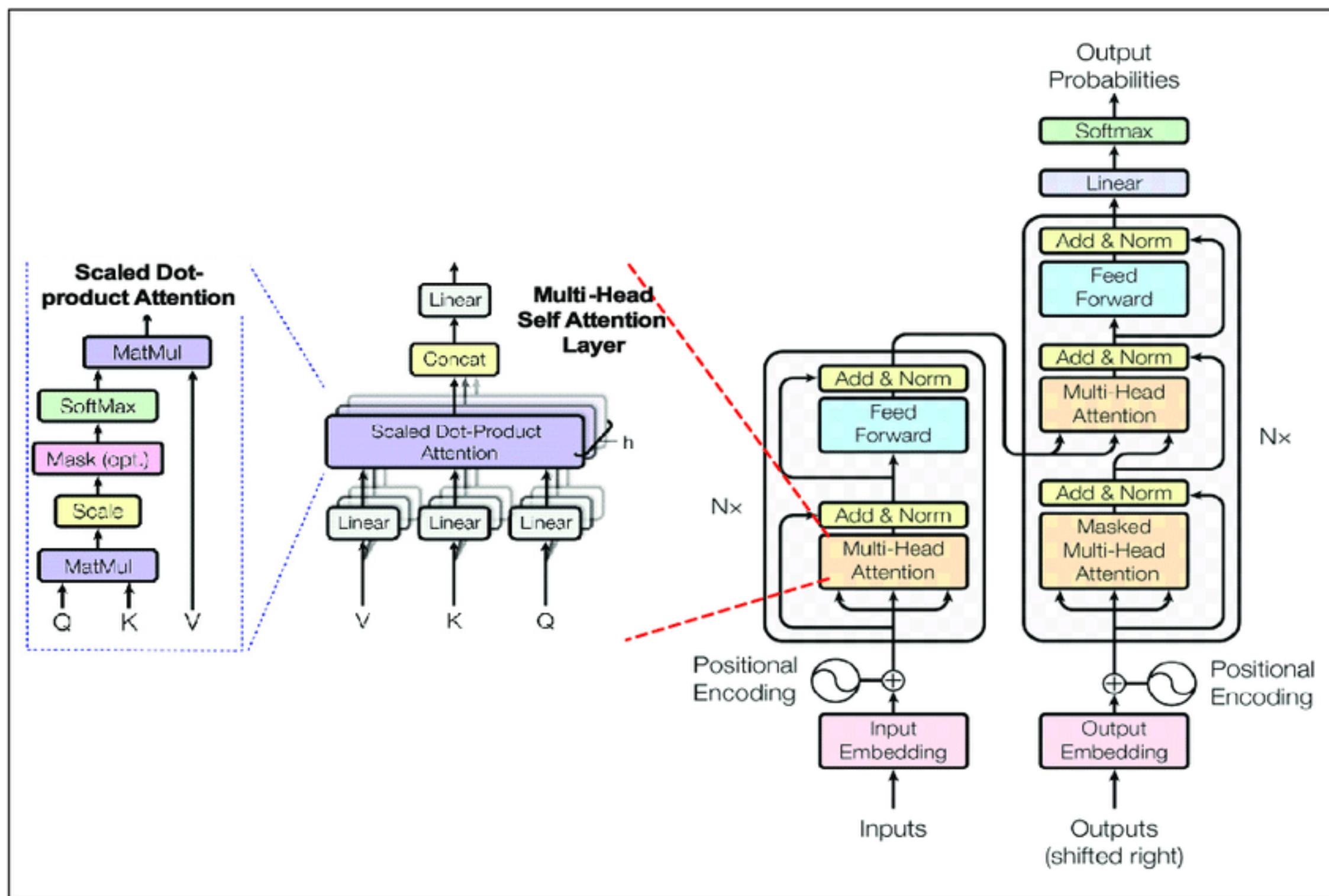
- What about activations?
- What about 2 or 3 billion parameter models?

$$10^9 * ( 2B + 2B + 12B) = 10^9 * 16B = 14.90GB$$

1 billion parameters      2 bytes per parameter      2 bytes per gradient      12 bytes per optimizer state

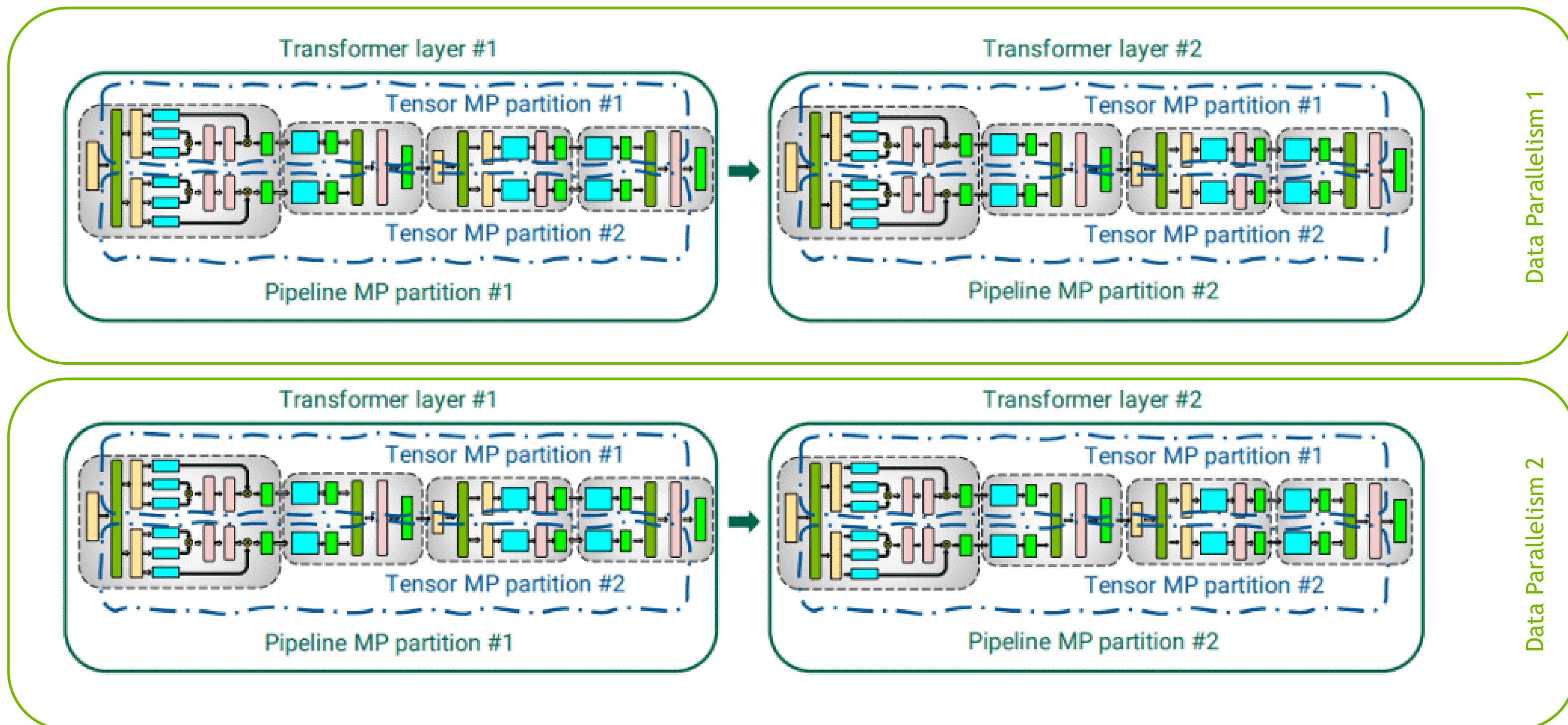
The diagram illustrates the calculation of memory usage for a 1 billion parameter model. It shows the breakdown of memory usage: 1 billion parameters (2 bytes each), 2 bytes per gradient, and 12 bytes per optimizer state, all multiplied by 10^9. Arrows point from the text labels to the corresponding terms in the equation.

# TRANSFORMER MODELS



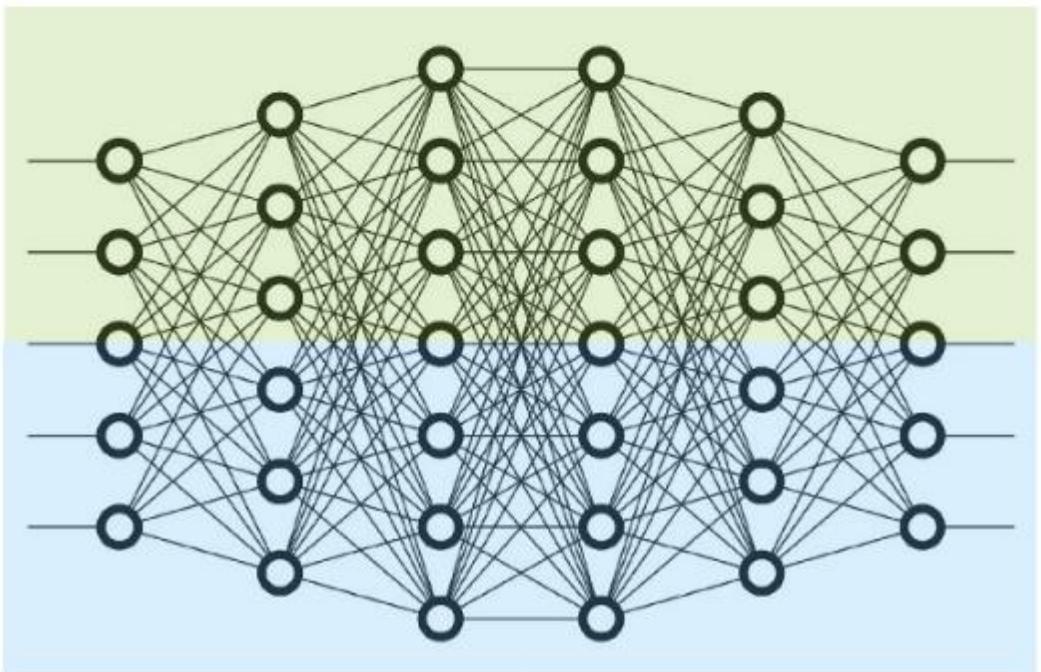
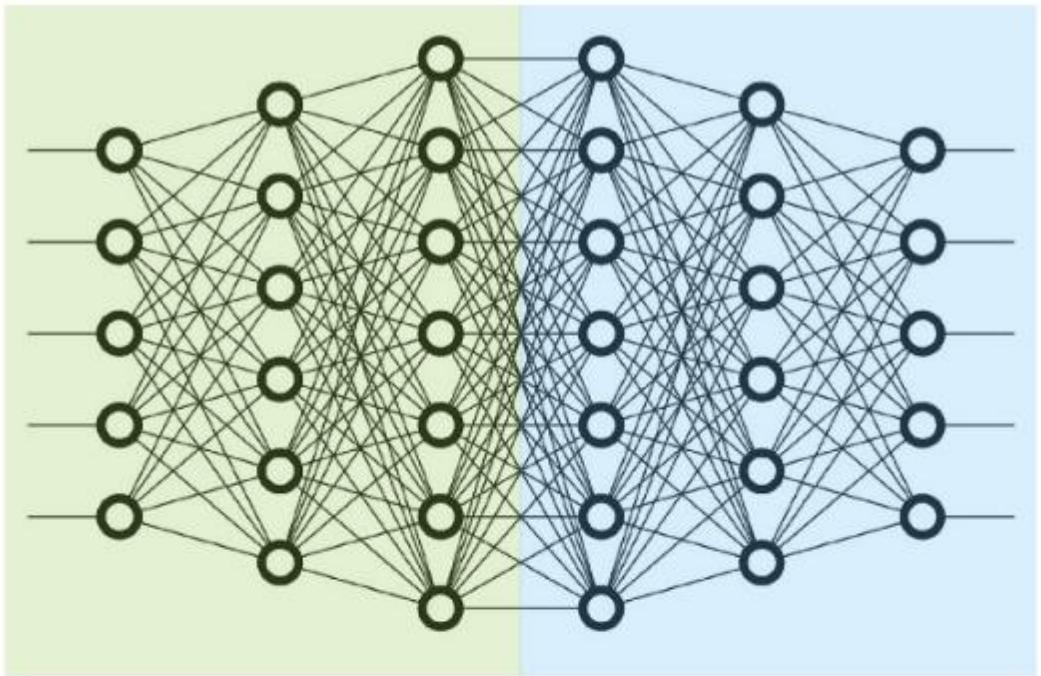
# MODEL IMPLEMENTATION

## Data, Pipeline and Tensor Parallelism

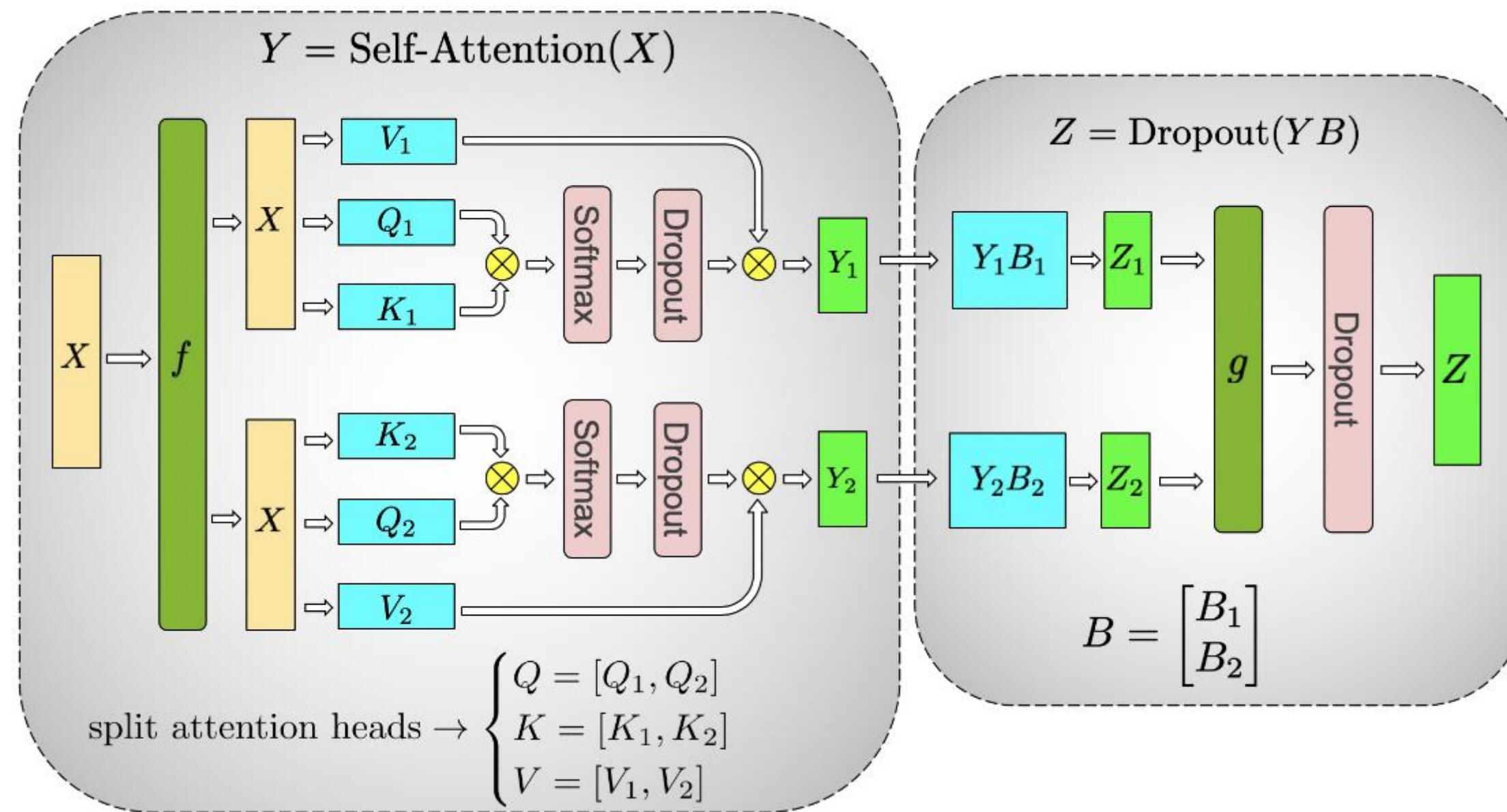


# MODEL PARALLELISM

- Pipeline (Inter-Layer) Parallelism
  - Split sets of layers across multiple devices
  - Layer 0,1,2 and layer 3,4,5 are on difference devices
- Tensor (Intra-Layer) Parallelism
  - Split individual layers across multiple devices
  - Both devices compute difference parts of Layer 0,1,2,3,4,5

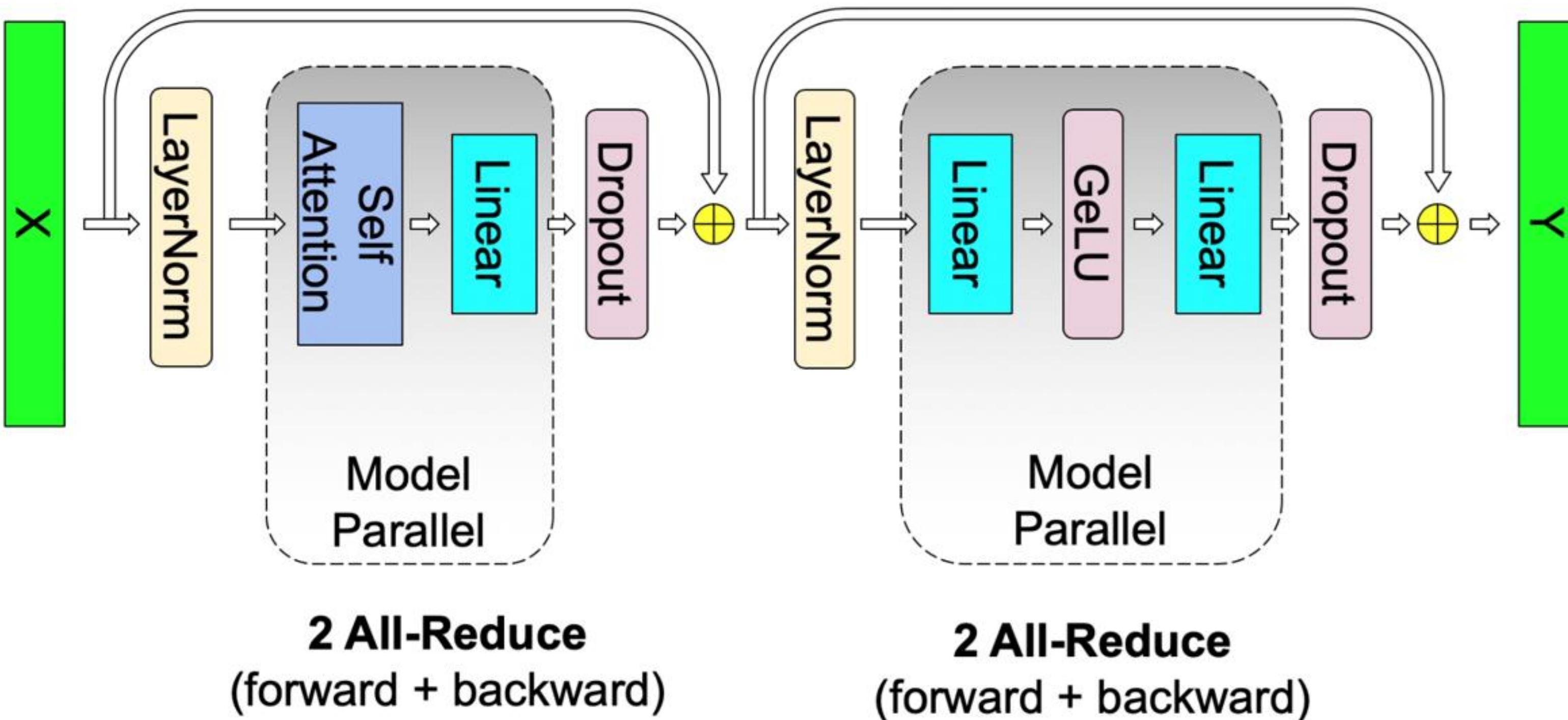


# SELF-ATTENTION



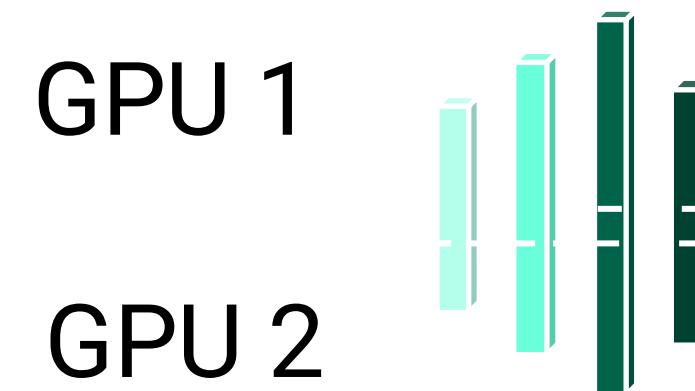
$f$  and  $g$  are conjugate,  $f$  is identity operator in the forward pass and all-reduce in the backward pass while  $g$  is all-reduce in forward and identity in backward.

# PARALLEL TRANSFORMER LAYER



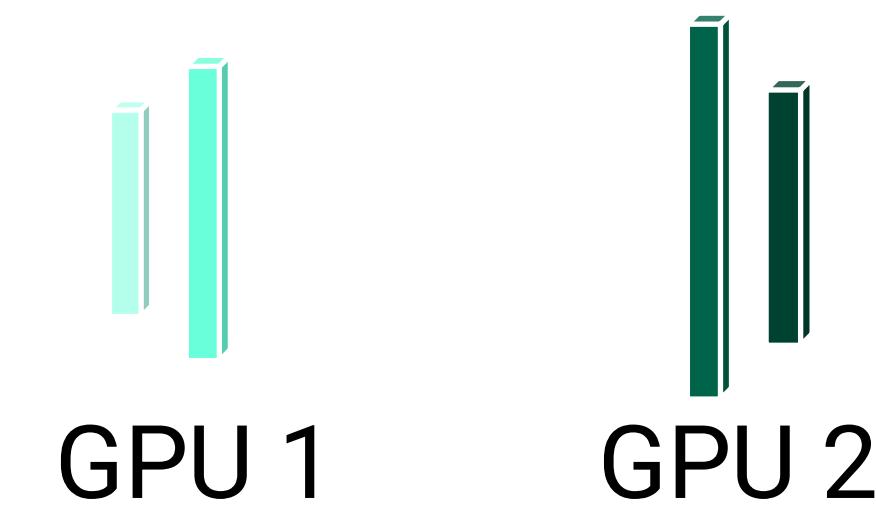
# COMPARING TENSOR AND PIPELINE PARALLELISM

## Tensor Parallelism



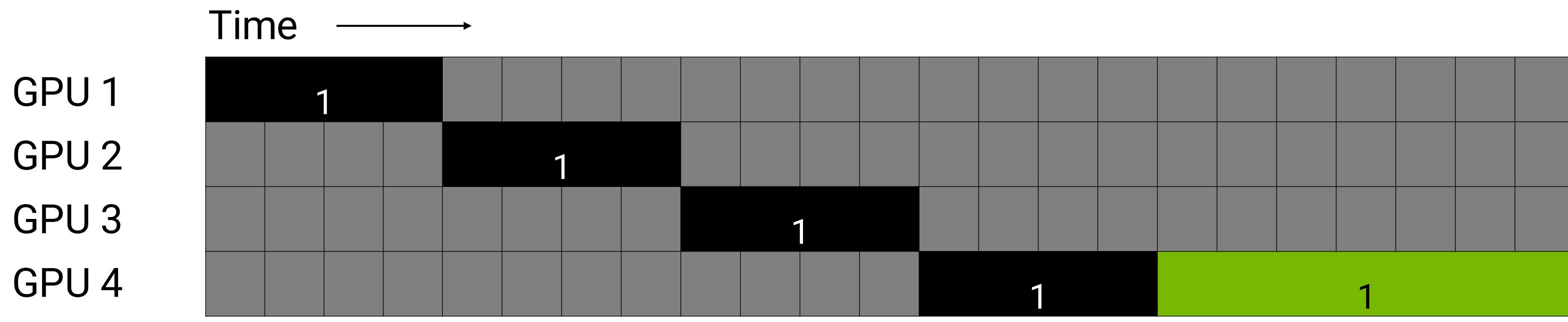
**Communication expensive**  
**Good performance across  
batch sizes**

## Pipeline Parallelism

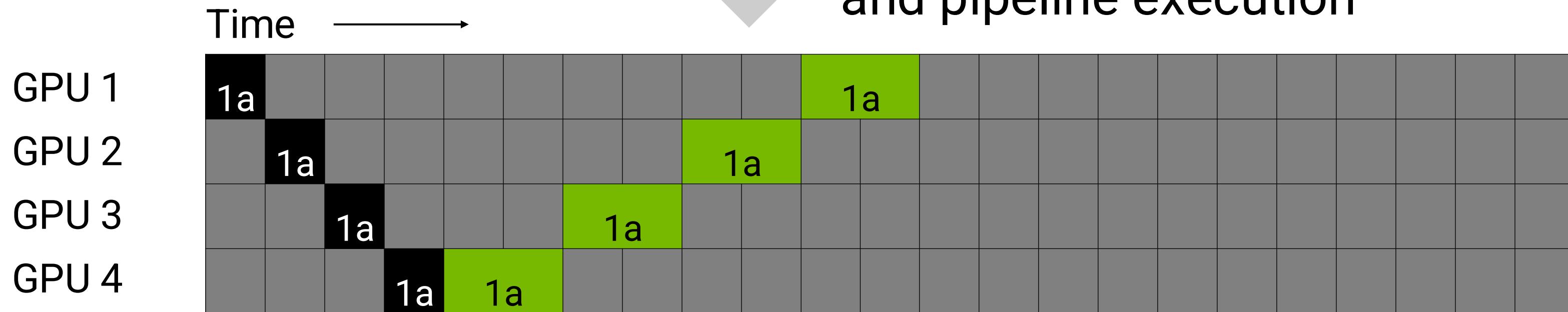


**Communication cheap**  
**Good performance at  
larger batch sizes (pipeline  
stall amortized)**

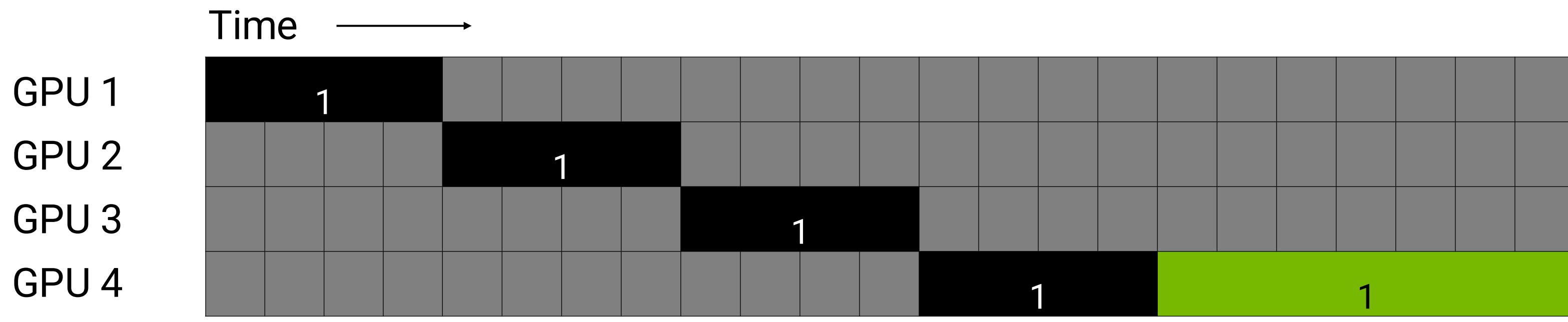
# PIPELINING



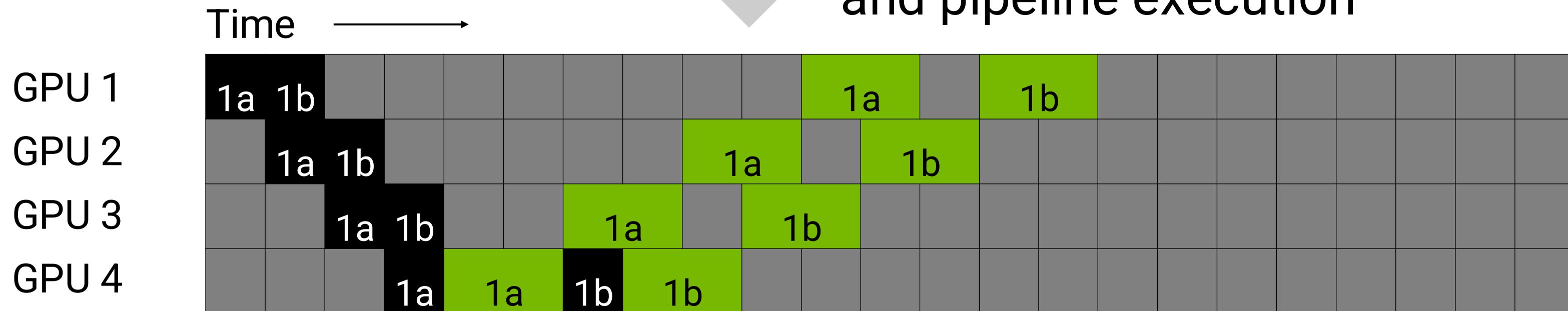
# Split batch into microbatches and pipeline execution



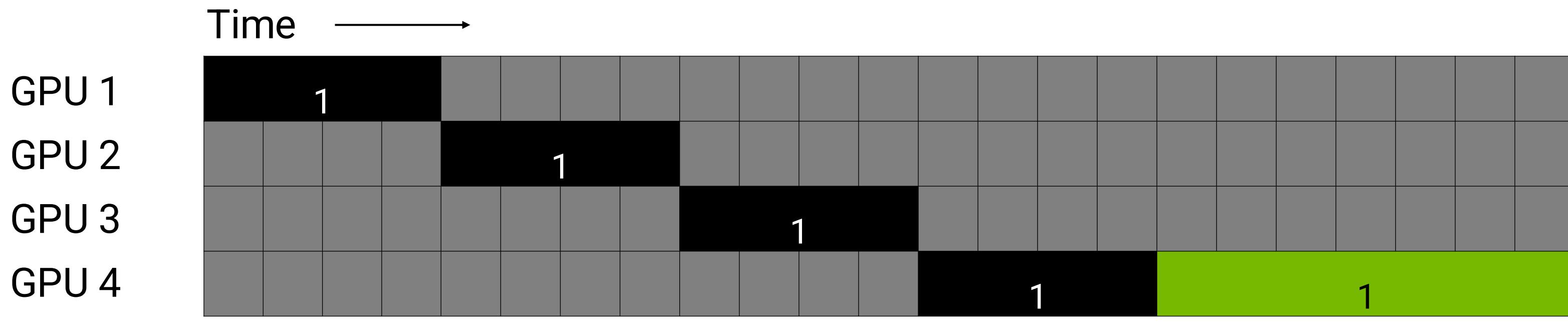
# PIPELINING



Split batch into microbatches  
and pipeline execution

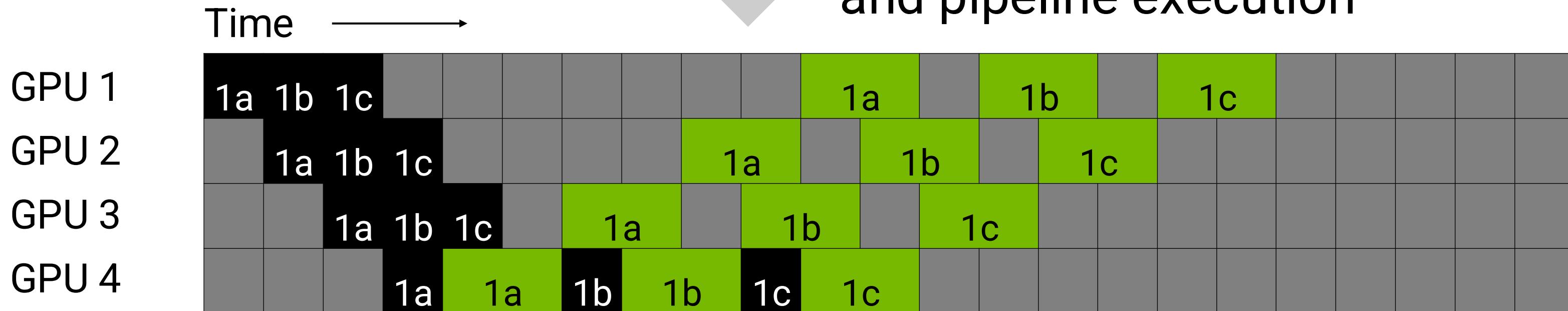


# PIPELINING

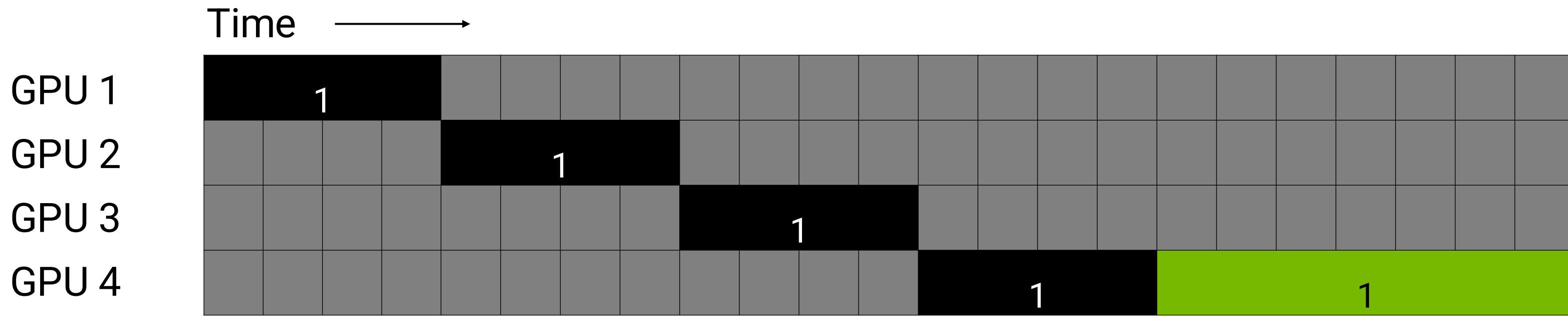


↓

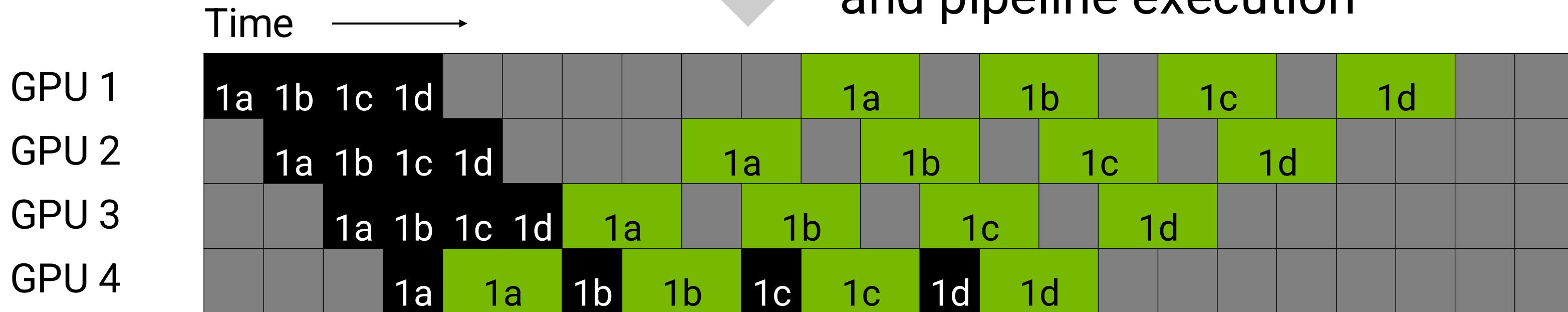
Split batch into microbatches  
and pipeline execution



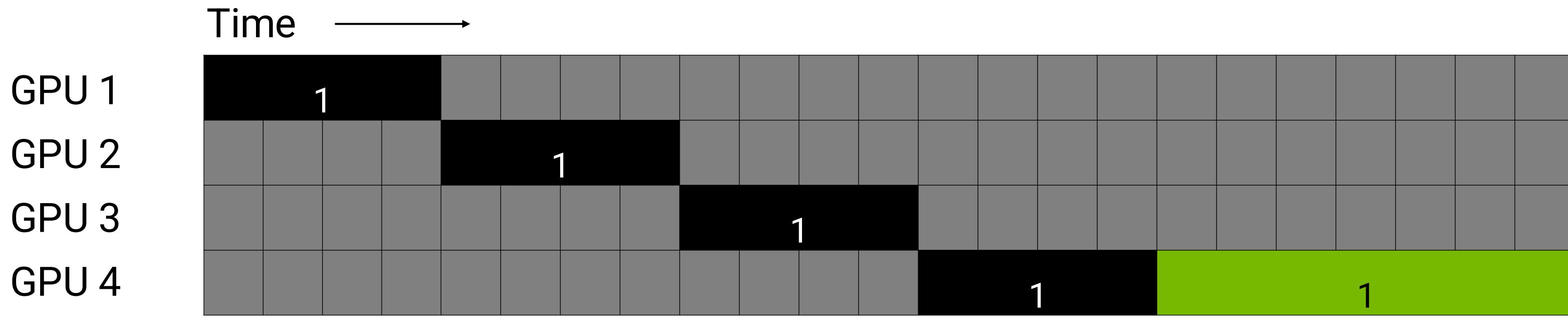
# PIPELINING



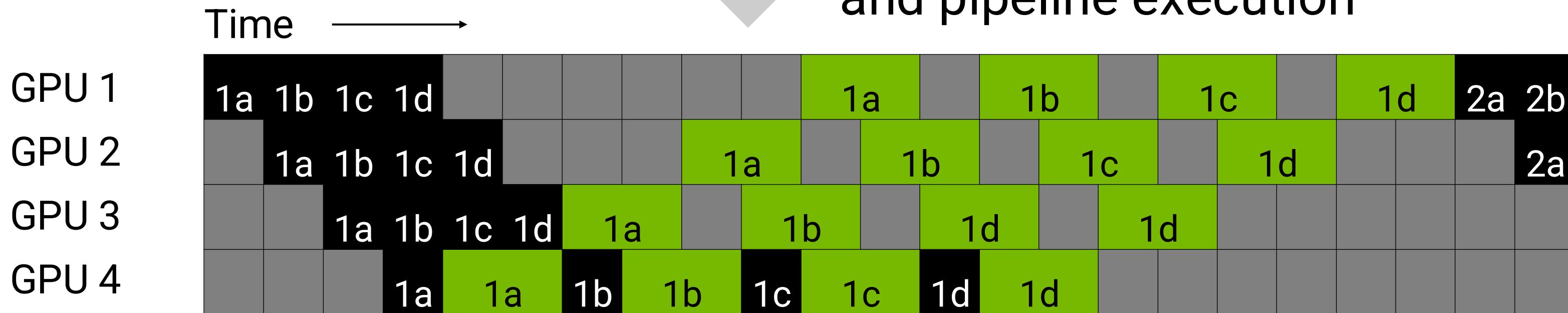
Split batch into microbatches  
and pipeline execution



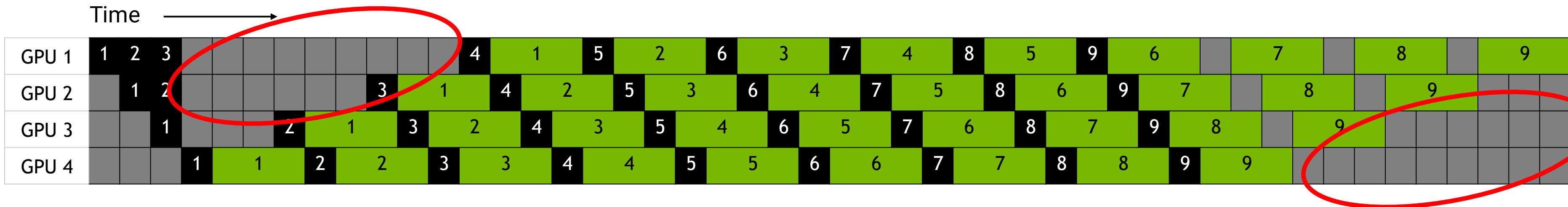
# PIPELINING



Split batch into microbatches  
and pipeline execution



# PIPELINE BUBBLES



$p$  : number of pipeline stages

$m$  : number of micro batches

$t_f$  : forward step time

$t_b$  : backward step time



$$\text{total time} = (m + p - 1) \times (t_f + t_b)$$

$$\text{ideal time} = m \times (t_f + t_b)$$

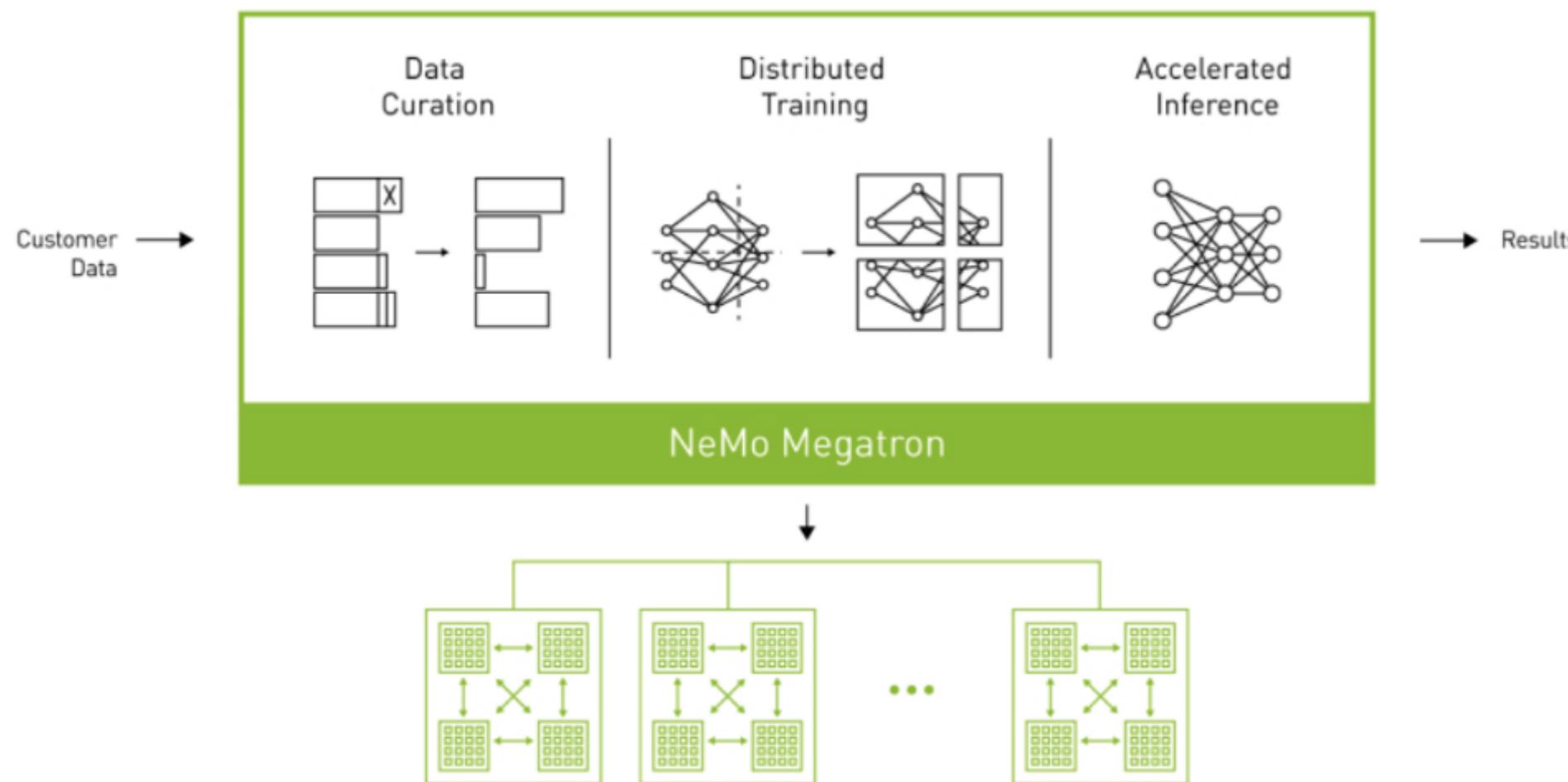
$$\text{bubble time} = (p - 1) \times (t_f + t_b)$$

$$\text{bubble time overhead} = \frac{\text{bubble time}}{\text{ideal time}} = \frac{p - 1}{m}$$

# NVIDIA NeMo Megatron

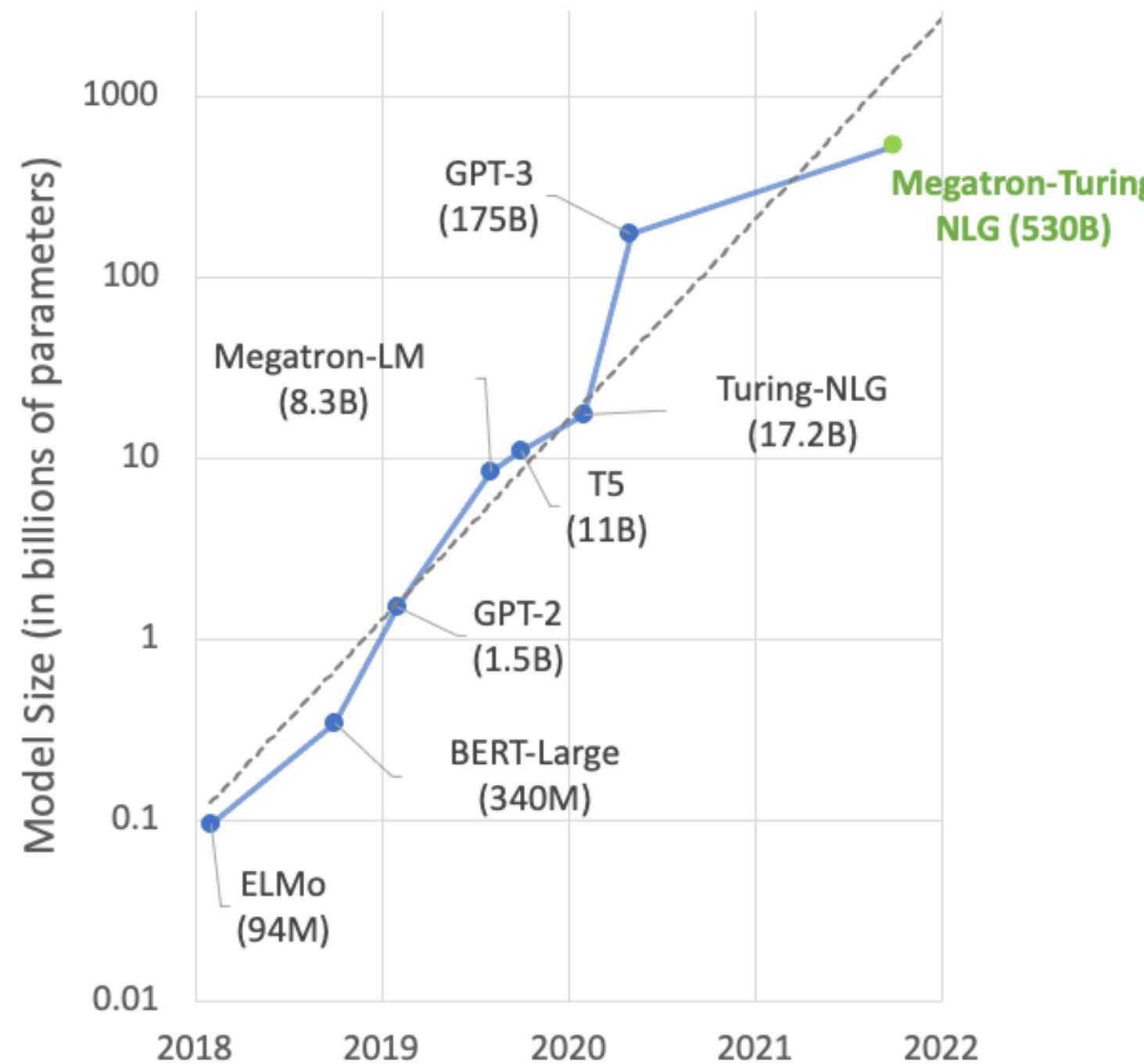
NVIDIA NeMo Megatron is an end-to-end framework for training and deploying LLMs with billions and trillions of parameters.

[Download Now](#)



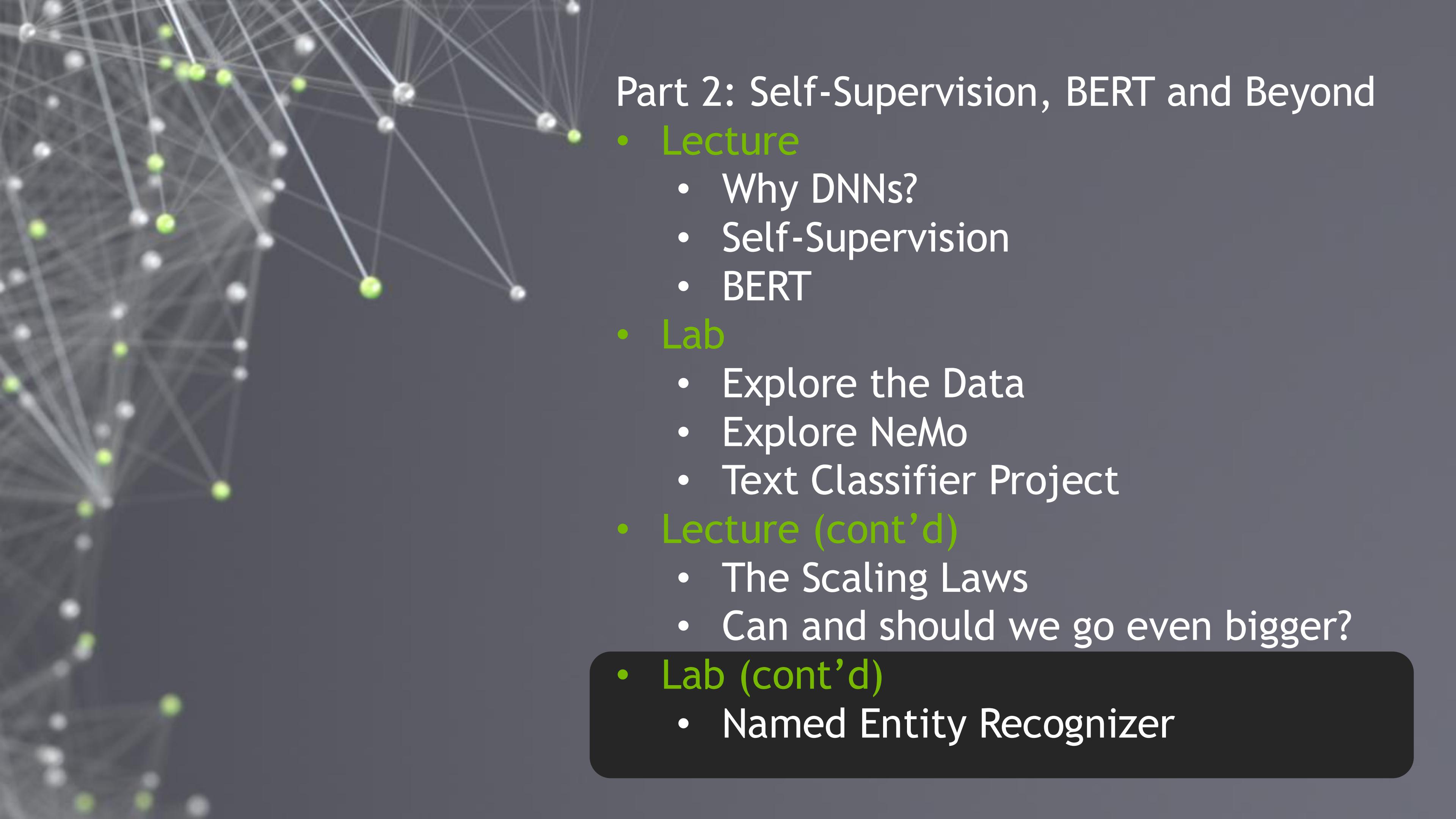
# MEGATRON-TURING NLG 530B

## Enabling the biggest of NLP models





THE LAB



## Part 2: Self-Supervision, BERT and Beyond

- **Lecture**
  - Why DNNs?
  - Self-Supervision
  - BERT
- **Lab**
  - Explore the Data
  - Explore NeMo
  - Text Classifier Project
- **Lecture (cont'd)**
  - The Scaling Laws
  - Can and should we go even bigger?
- **Lab (cont'd)**
  - Named Entity Recognizer

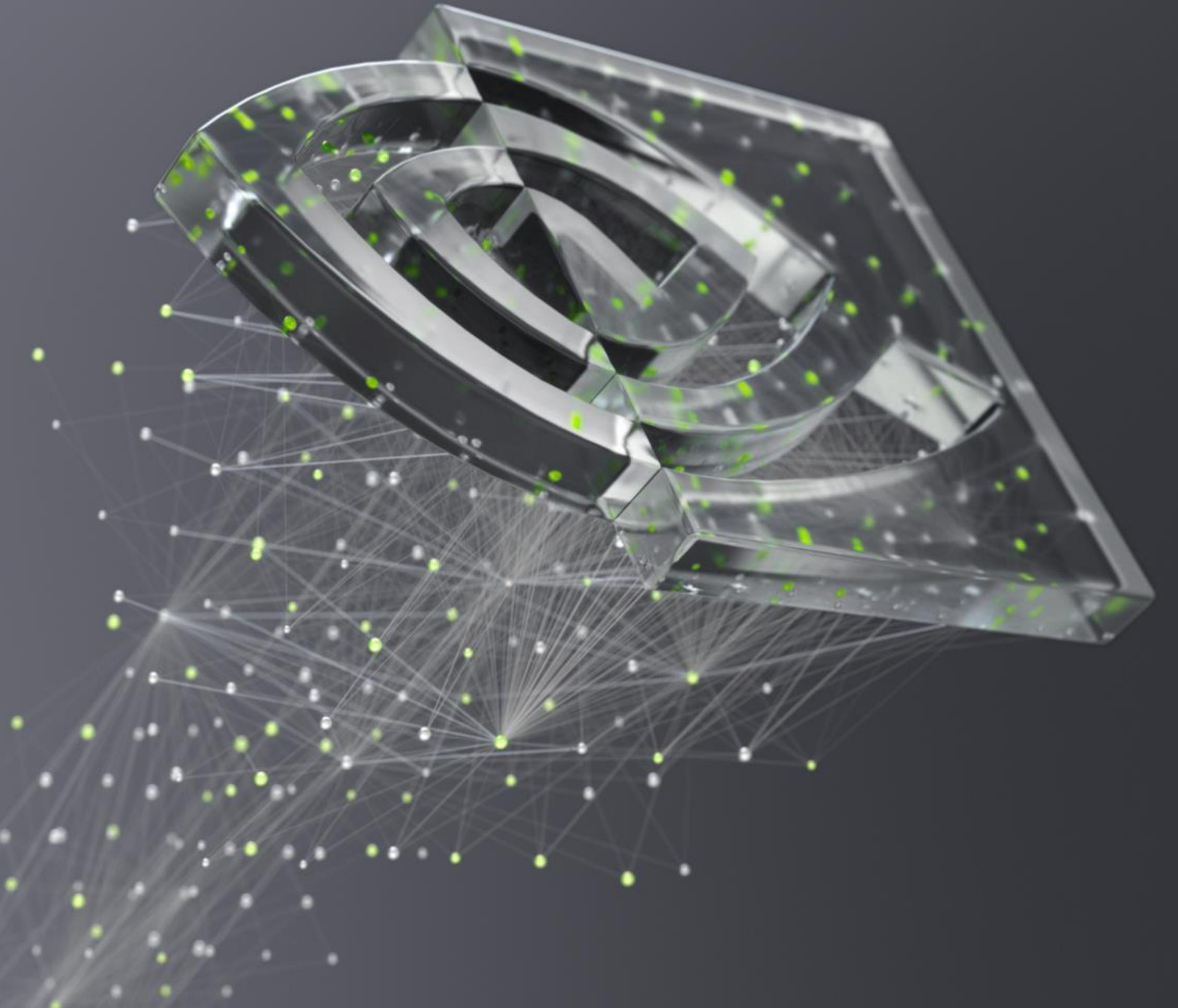
A network graph visualization on a dark background. It consists of numerous small, semi-transparent circular nodes. Some nodes are white, while others are a vibrant lime green. These nodes are interconnected by a dense web of thin, gray lines representing edges. The overall effect is one of a complex, decentralized system or a social network.

IN THE NEXT CLASS...

# NEXT CLASS

## Overview

1. Discuss how to design your model for efficient inference
2. Discuss how to optimise your model for efficient execution
3. Discuss how to efficiently host a largely Conversational AI application



DEEP  
LEARNING  
INSTITUTE