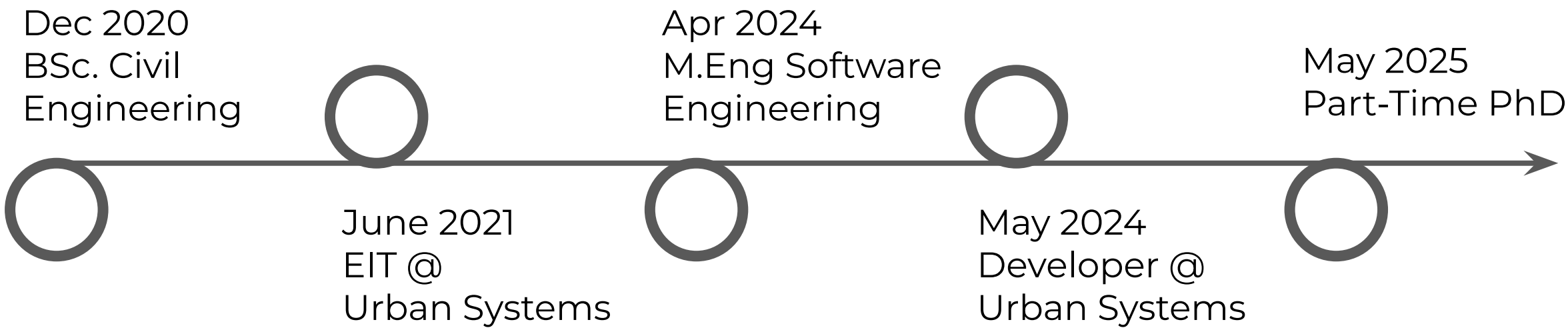
Intro to Large Language Models

UC Workshop

INTRODUCTION, OVERVIEW, & FINE-TUNING

URBAN SYSTEMS

Who is Corey Yang-Smith?



URBAN SYSTEMS

Who is Corey Yang-Smith?

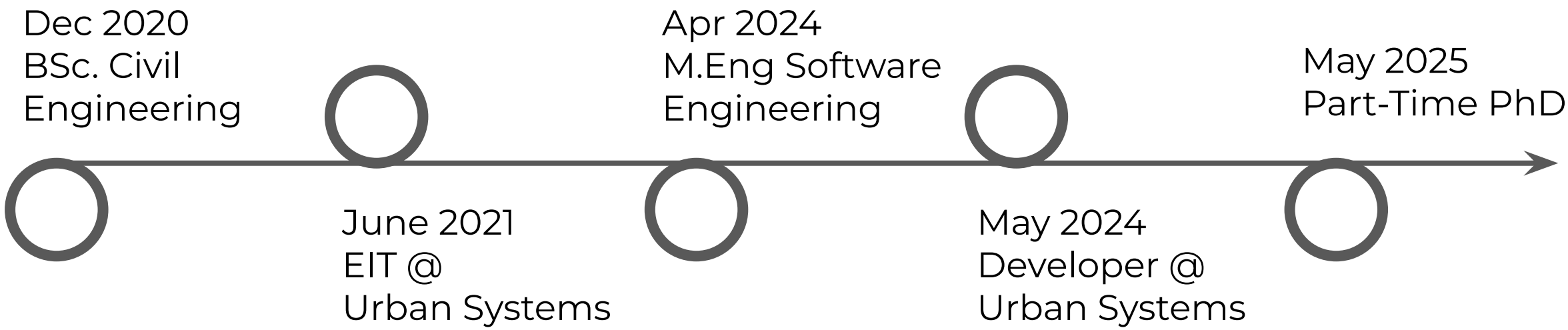
Who is Corey Yang-Smith?

Dec 2020
BSc. Civil
Engineering

June 2021
EIT @
Urban Systems

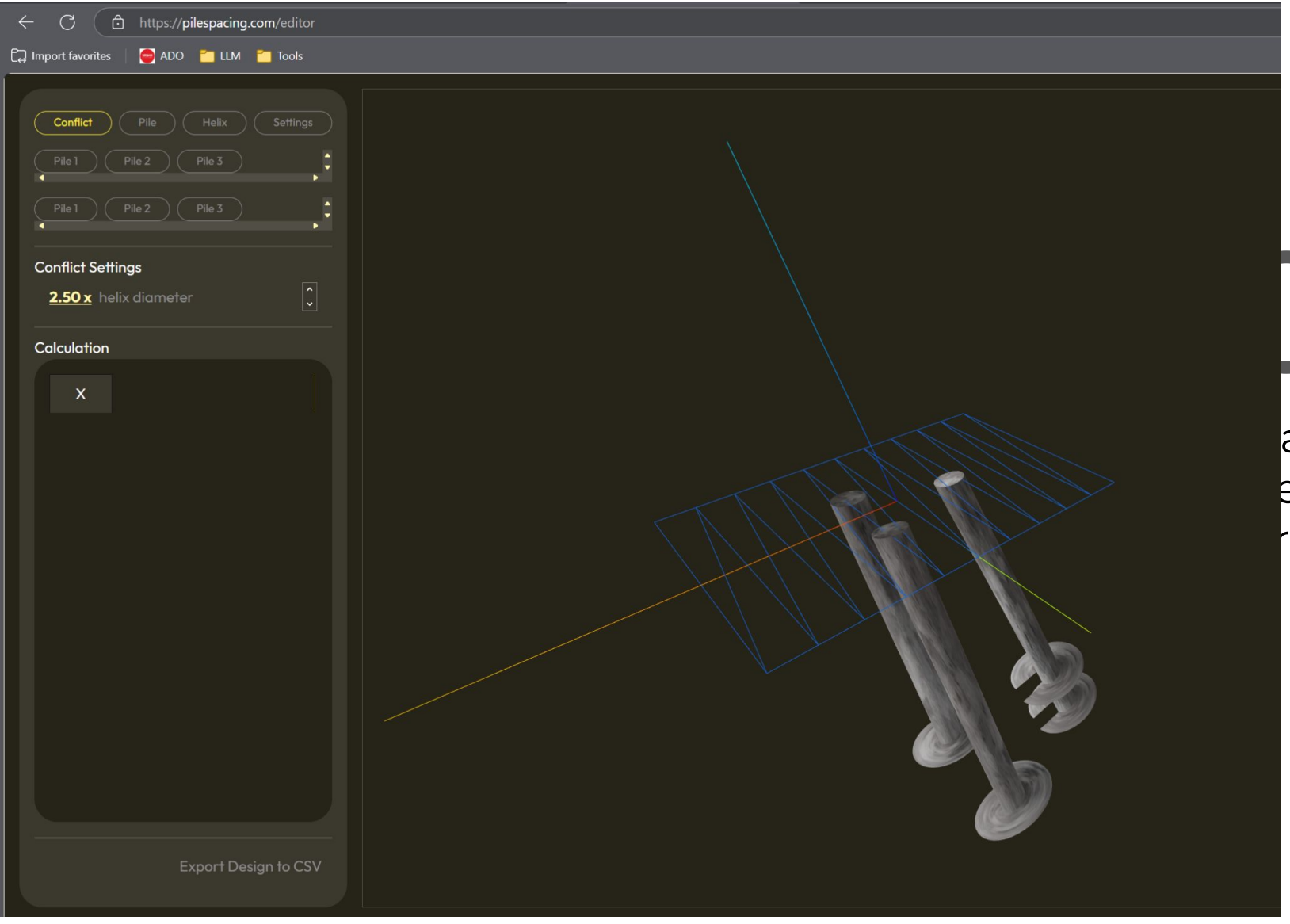
URBAN SYSTEMS

Who is Corey Yang-Smith?



URBAN SYSTEMS

Who is Corey Yang-Smith?

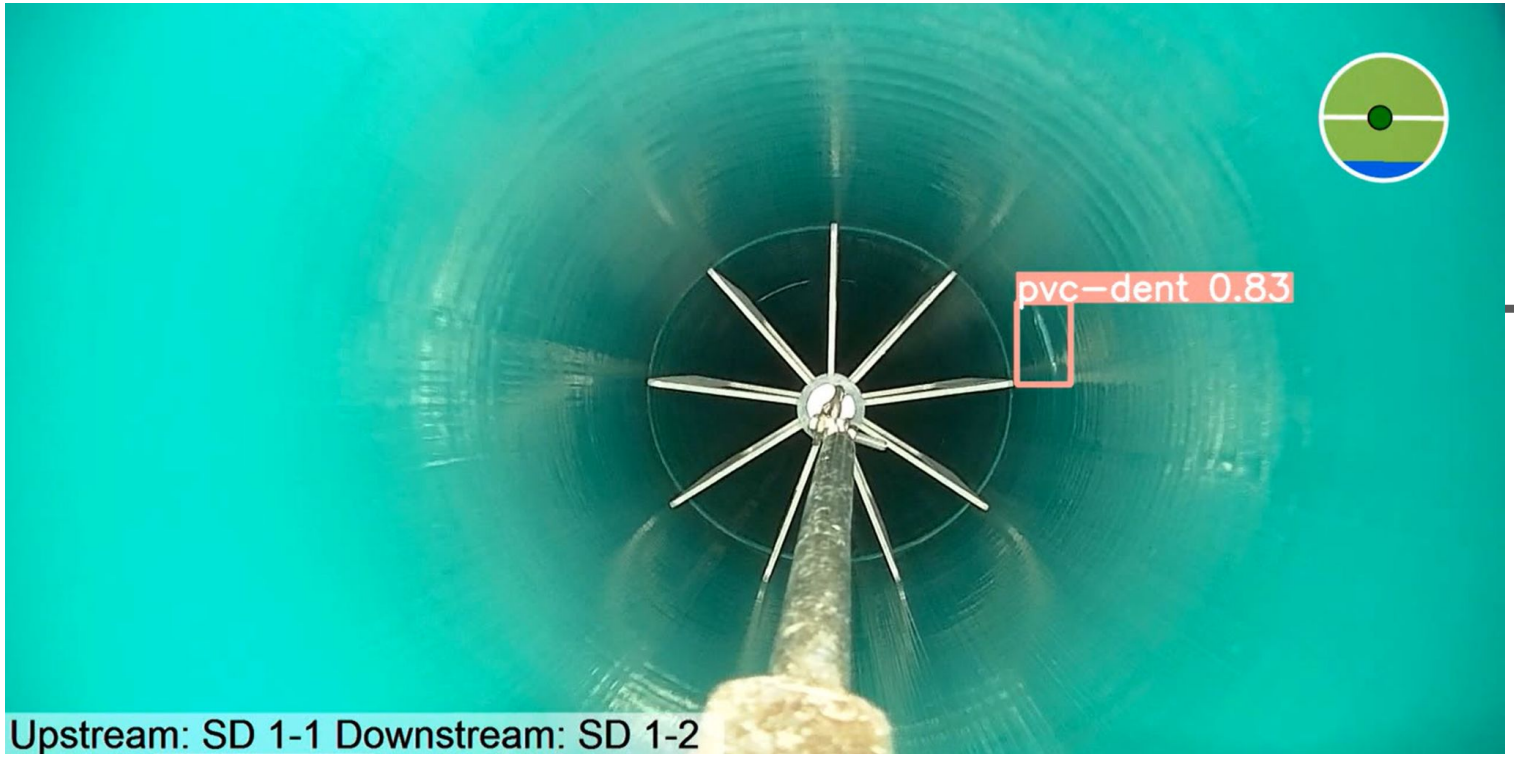


May 2025
Part-Time PhD

ay 2024
eveloper @
rban Systems

URBAN SYSTEMS

Who is Corey Yang-Smith?

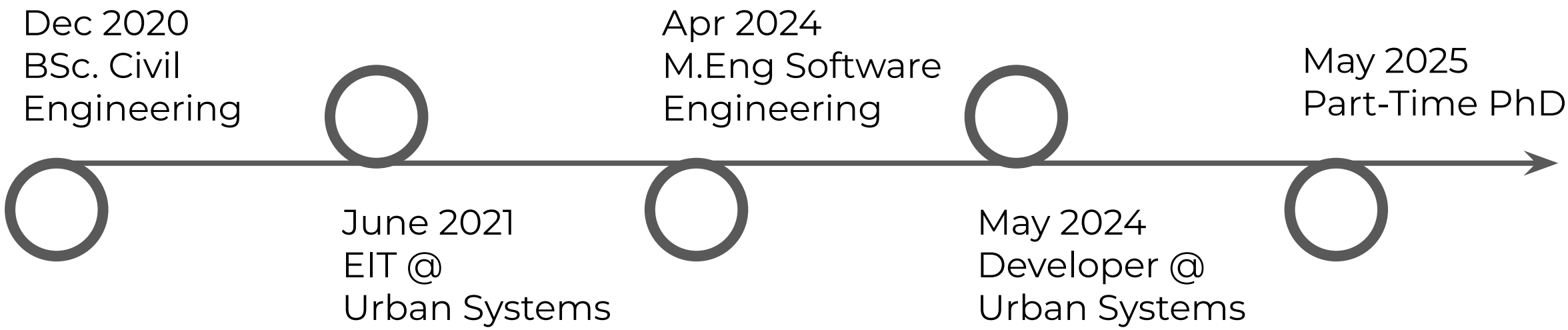


May 2024
Developer @
Urban Systems

May 2025
Part-Time PhD

URBAN SYSTEMS

Who is Corey Yang-Smith?



URBAN SYSTEMS

OBJECTIVES

LECTURE

- Cover fundamentals of large language models (LLMs)
- Explore design considerations and trade-offs
- Explore model training (pre-training, fine-tuning)
- Explore high-impact prompting techniques

WORKSHOP

- Gain hands-on experience interfacing with LLMs through API and libraries
- Gain experience with different prompting techniques
- Fine-tune a local model

URBAN SYSTEMS

AGENDA

#1

COMPONENTS OF AN LLM

- Model Construction
- Model Design
- Model Sizing

#2

PROMPTING STRUCTURE

- Prompting Structure
- Parameters
- Model Interface

#3

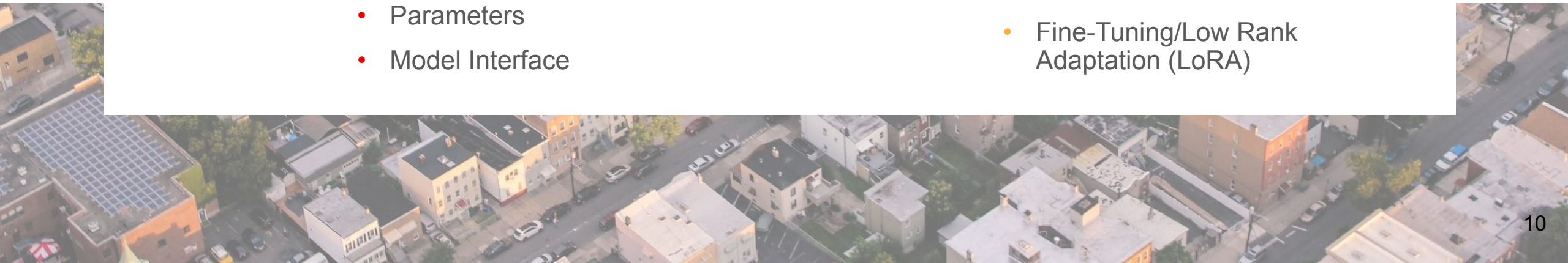
PROMPTING TECHNIQUES

- Few-Shot
- Chain-of-Thought

#4

MODEL TRAINING

- Pre-Training
- Fine-Tuning/Low Rank Adaptation (LoRA)



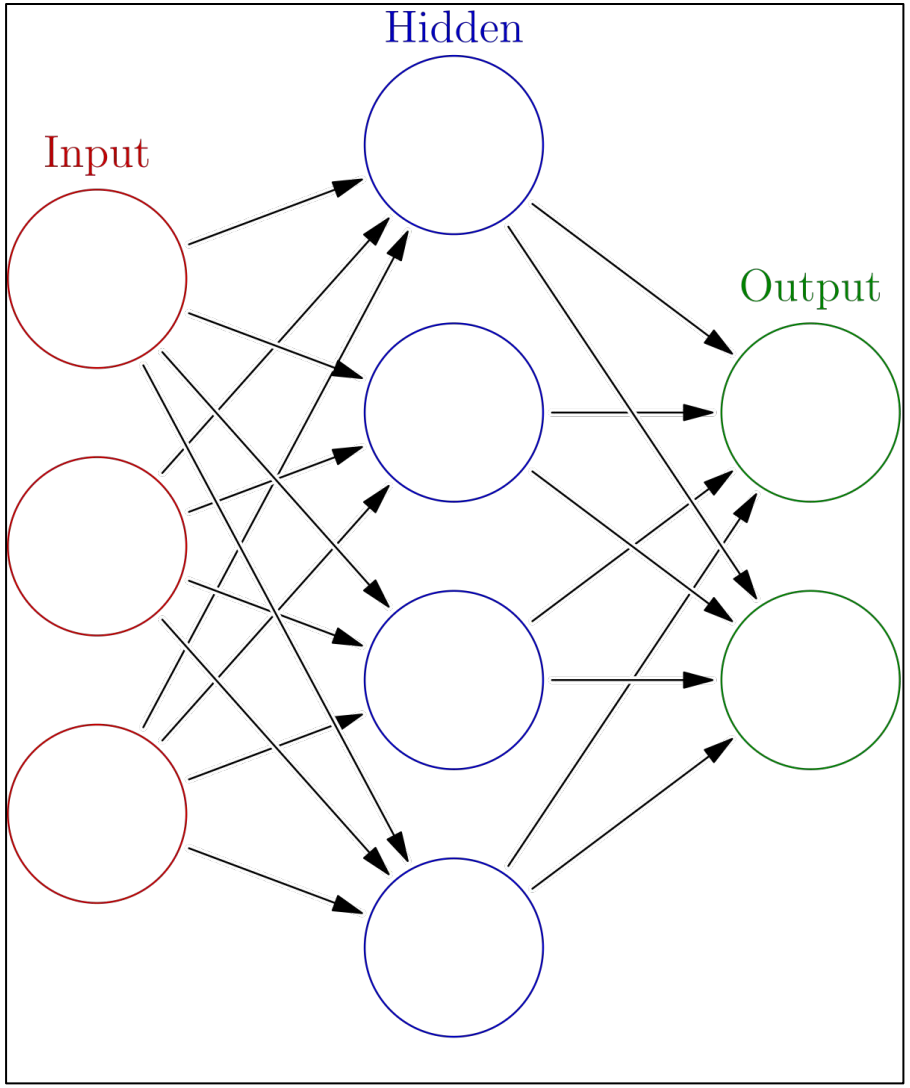
WHAT IS A LARGE LANGUAGE MODEL?

WHAT IS A ~~LARGE~~ LANGUAGE MODEL?

URBAN SYSTEMS

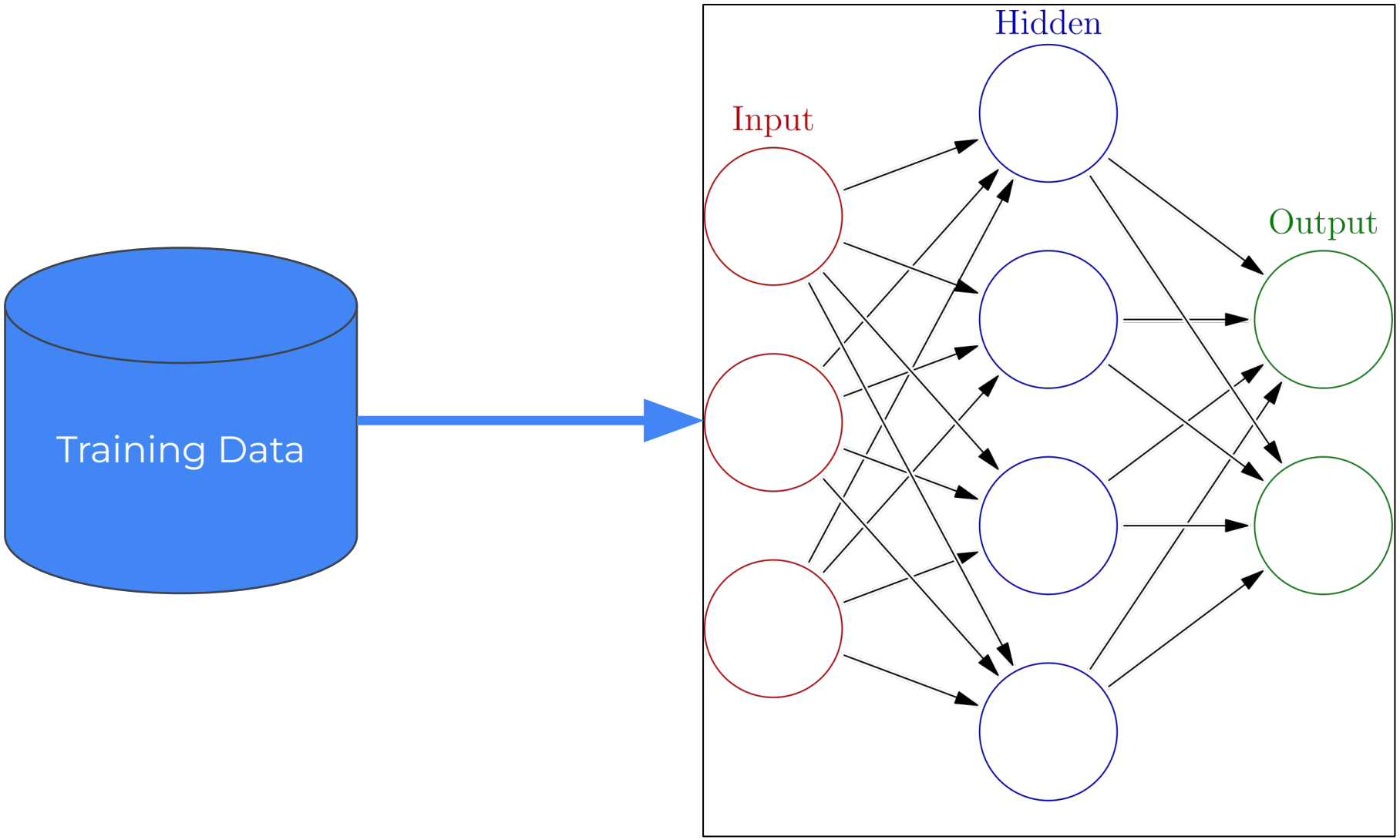
WHAT IS A ~~LARGE LANGUAGE~~ MODEL?

WHAT IS A ~~LARGE LANGUAGE~~ MODEL?



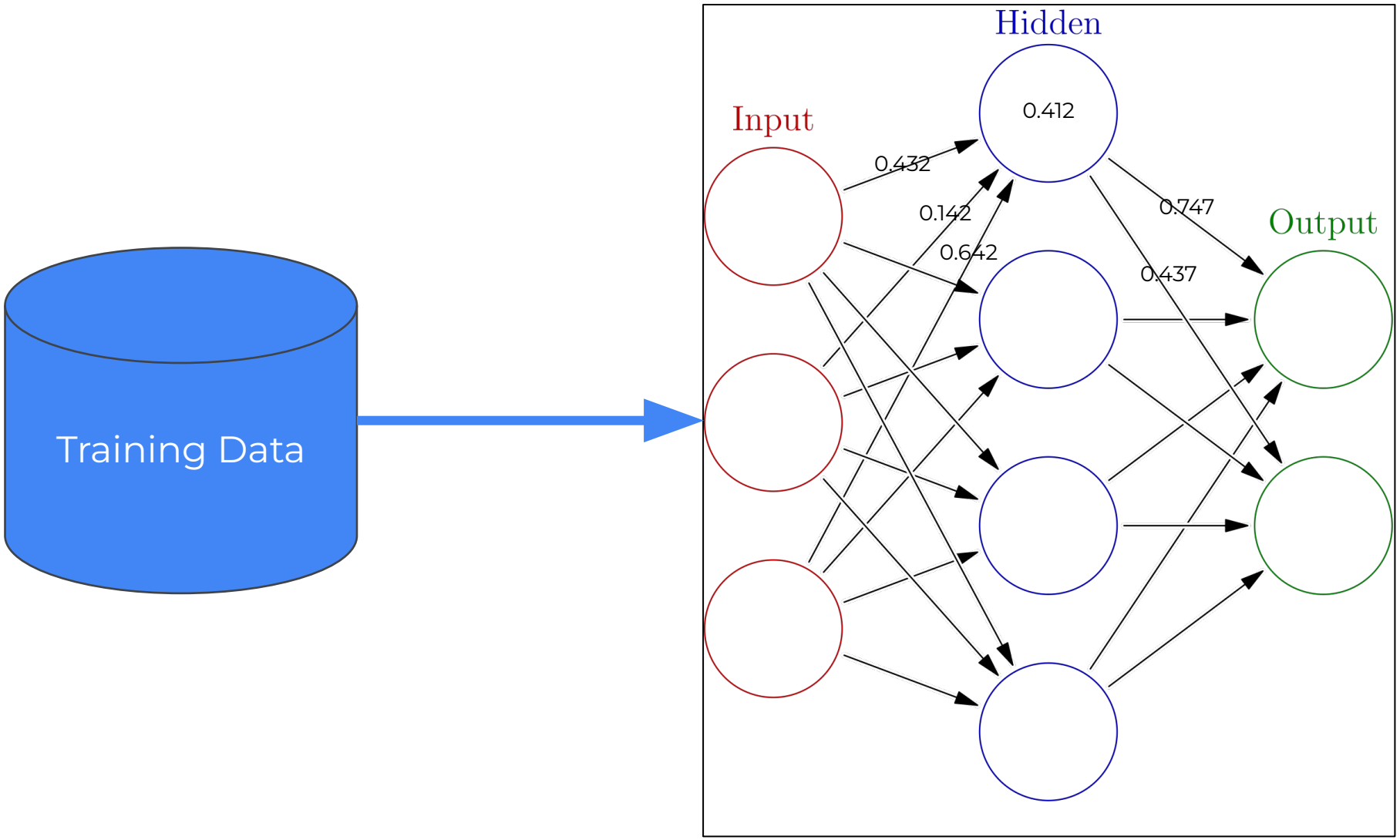
URBAN SYSTEMS

WHAT IS A ~~LARGE LANGUAGE~~ MODEL?



URBAN SYSTEMS

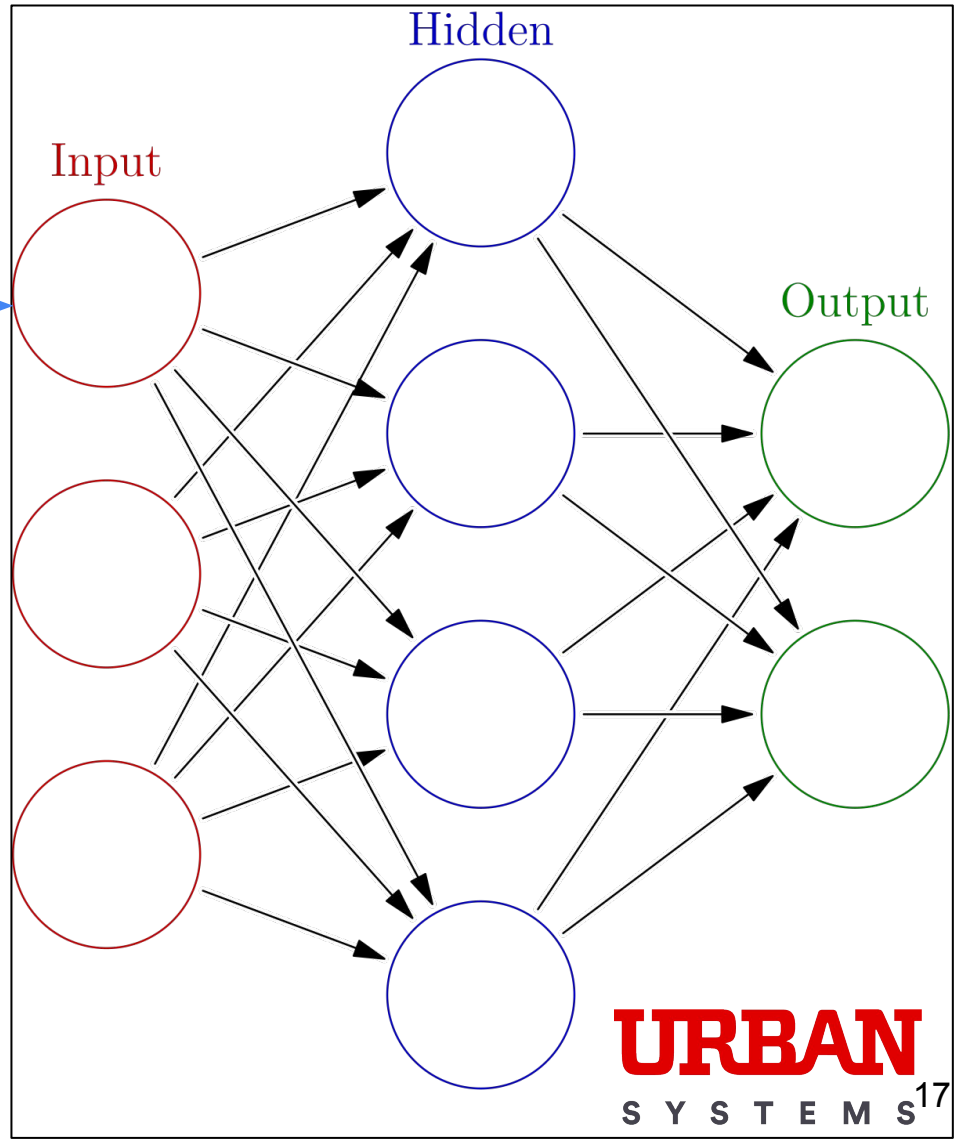
WHAT IS A ~~LARGE LANGUAGE~~ MODEL?



WHAT IS A ~~LARGE LANGUAGE~~ MODEL?

Speed
40km/hr

40



URBAN SYSTEMS

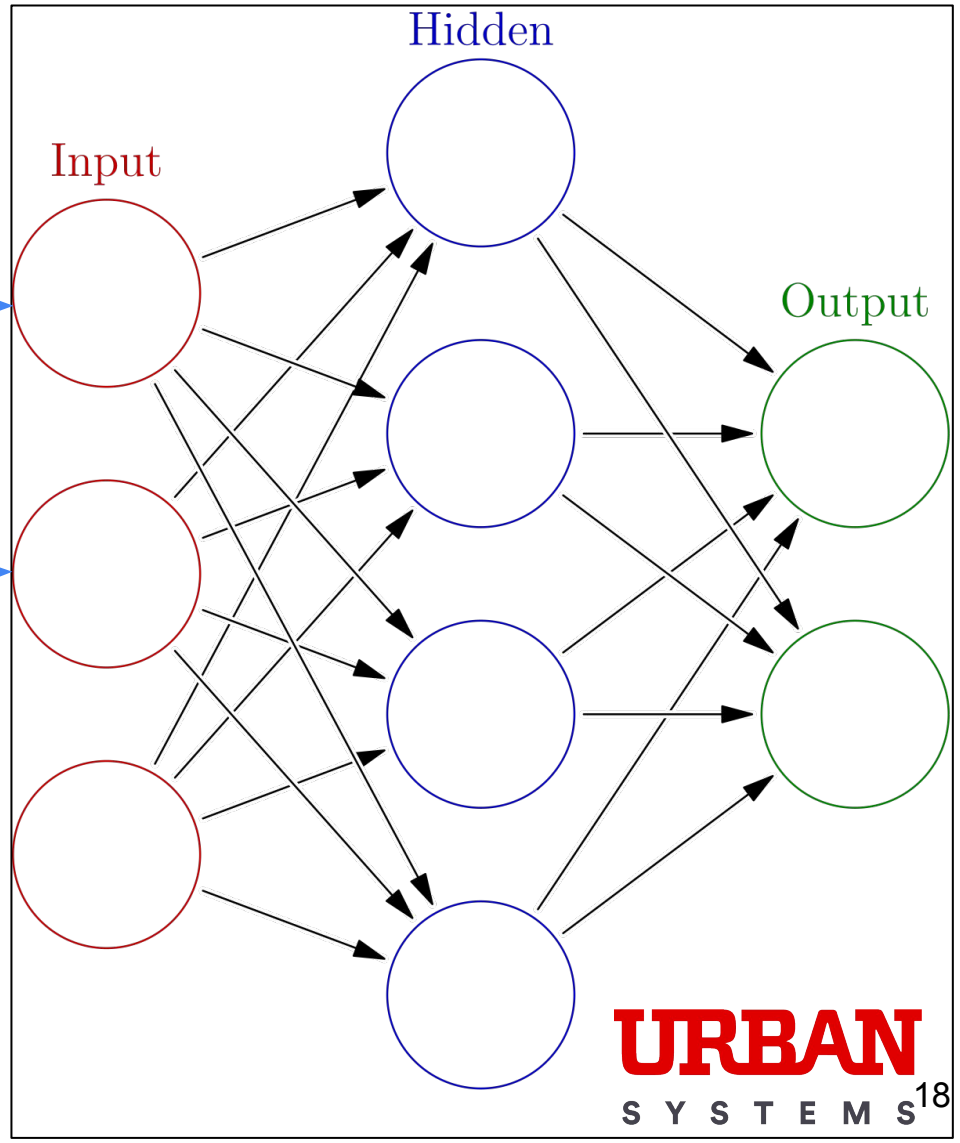
WHAT IS A ~~LARGE LANGUAGE~~ MODEL?

Speed
40km/hr

40

Time of Day
Midday

2



URBAN SYSTEMS

WHAT IS A ~~LARGE LANGUAGE~~ MODEL?

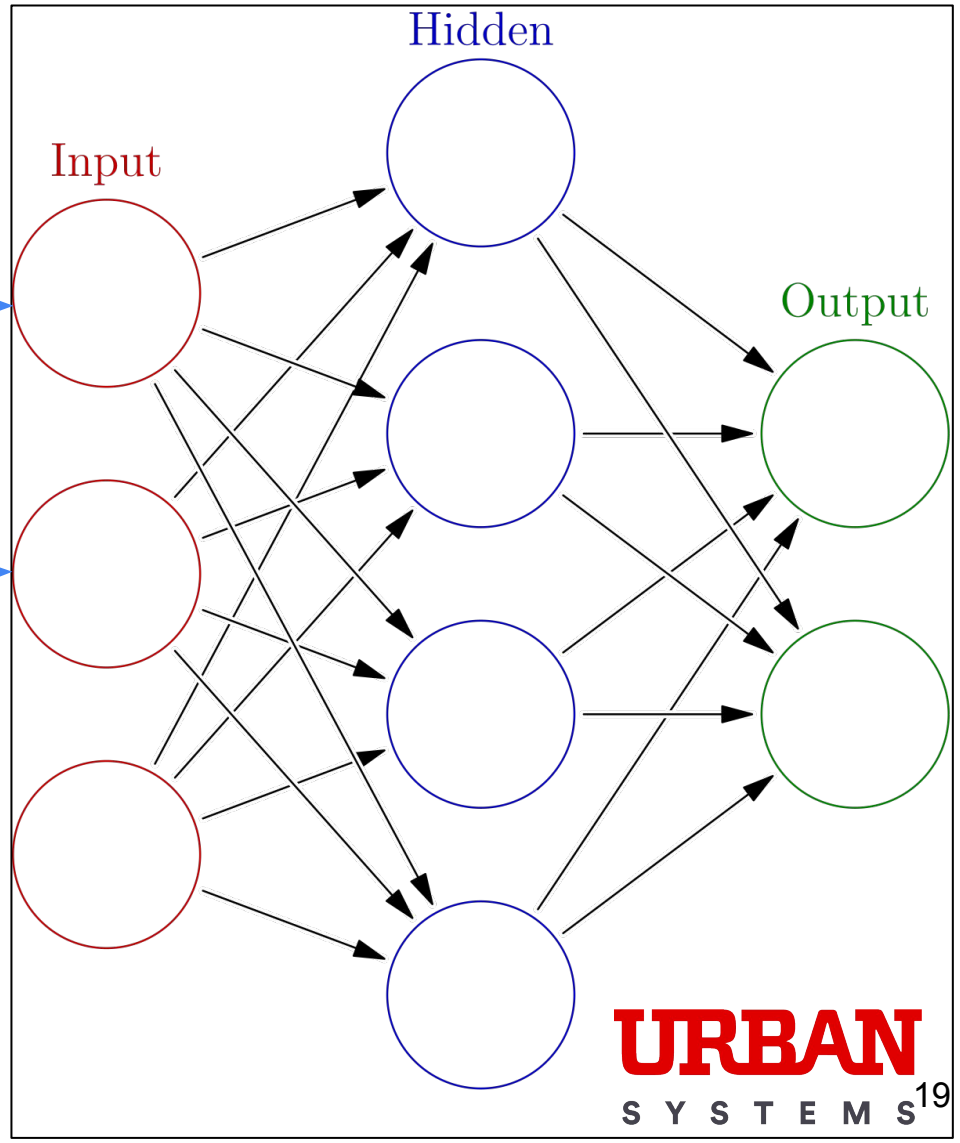
Speed
40km/hr

40

Time of Day
Midday

Ordinal Encoding

2



URBAN SYSTEMS

WHAT IS A ~~LARGE LANGUAGE~~ MODEL?

Speed
40km/hr

40

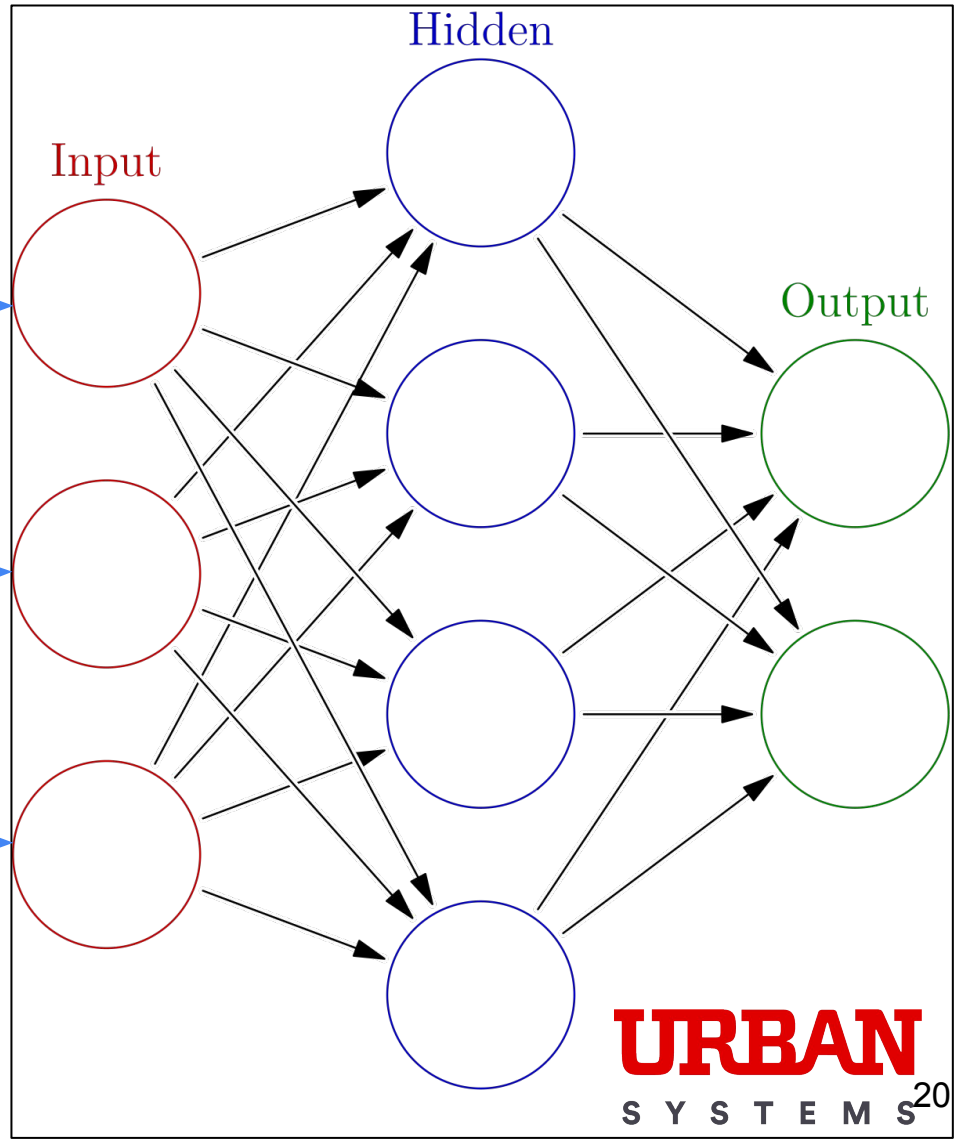
Time of Day
Midday

Ordinal Encoding

2

Image Preprocessing

[[232,12,123],
[231,11,143],
...]



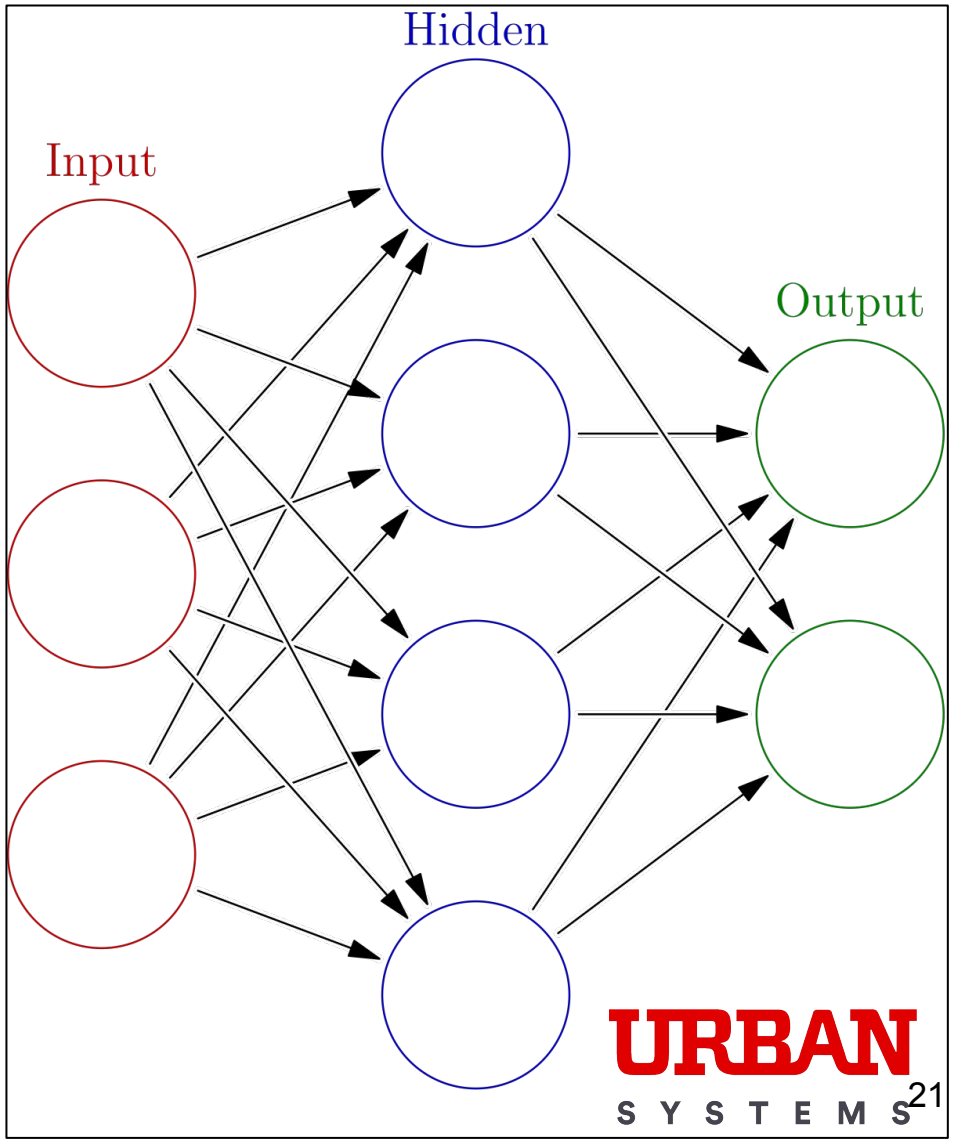
URBAN SYSTEMS

WHAT IS A ~~LARGE~~ LANGUAGE MODEL?

Text Description

Busy, multi-lane collector road near the University of Calgary, connecting commuters and nearby amenities.

?



URBAN SYSTEMS

WHAT IS A LARGE LANGUAGE MODEL?

Text Description

Busy, multi-lane collector road near the University of Calgary, connecting commuters and nearby amenities.

amenities.

amenities.

amenities.

amenities.

amenities.

amenities.

amenities.

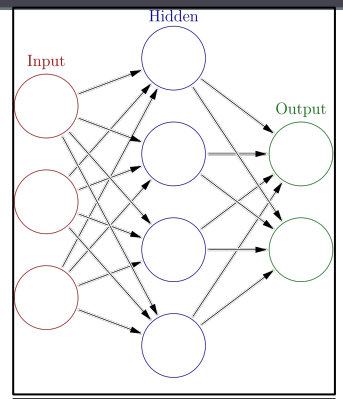
amenities.

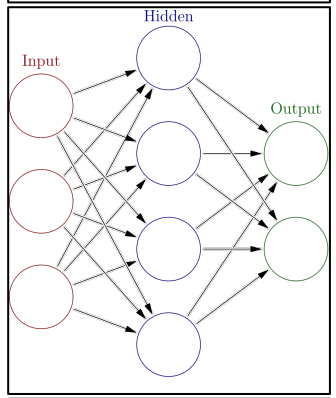
amenities.

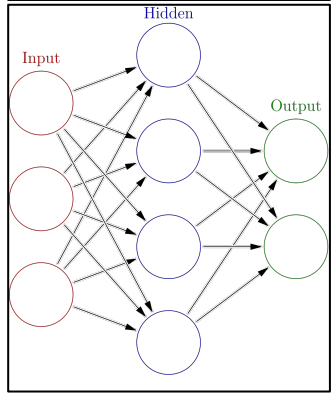
amenities.

amenities.

?







URBAN SYSTEMS

COMPONENTS OF AN LLM

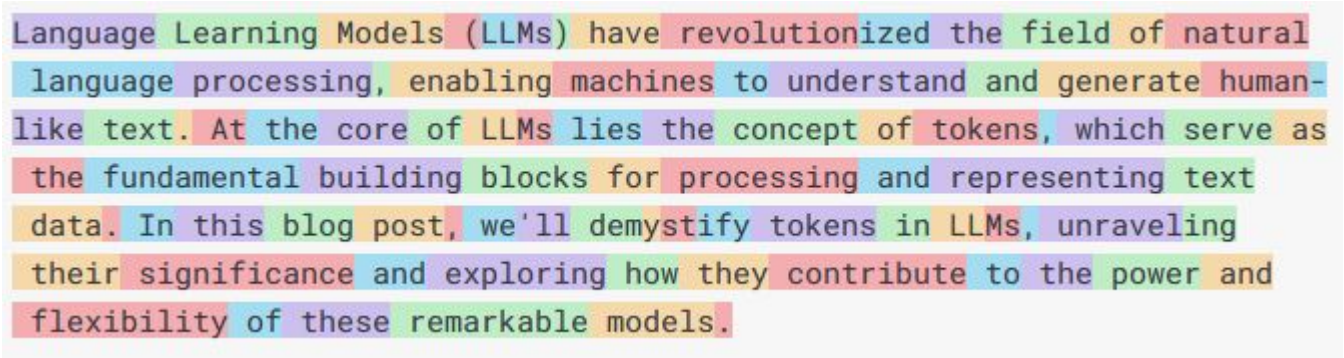
From Paper to Practice

URBAN SYSTEMS

MODEL CONSTRUCTION

TOKENIZATION

- *“Tokenization is used in NLP to split paragraphs and sentences into smaller units that can be more easily assigned meaning”* [Source]



Useful Resources

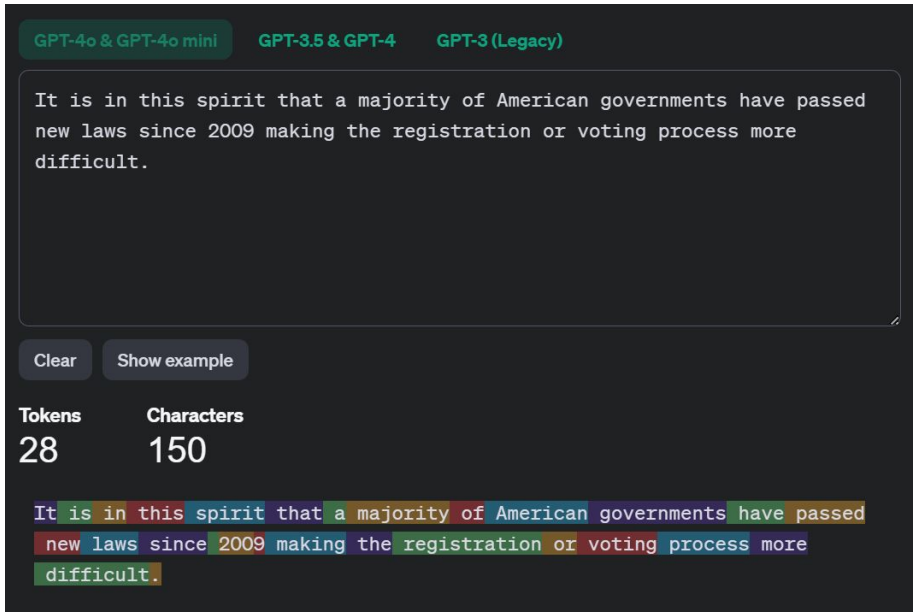
- [tool] OpenAI Tokenizer
- [tool] Token Visualizer

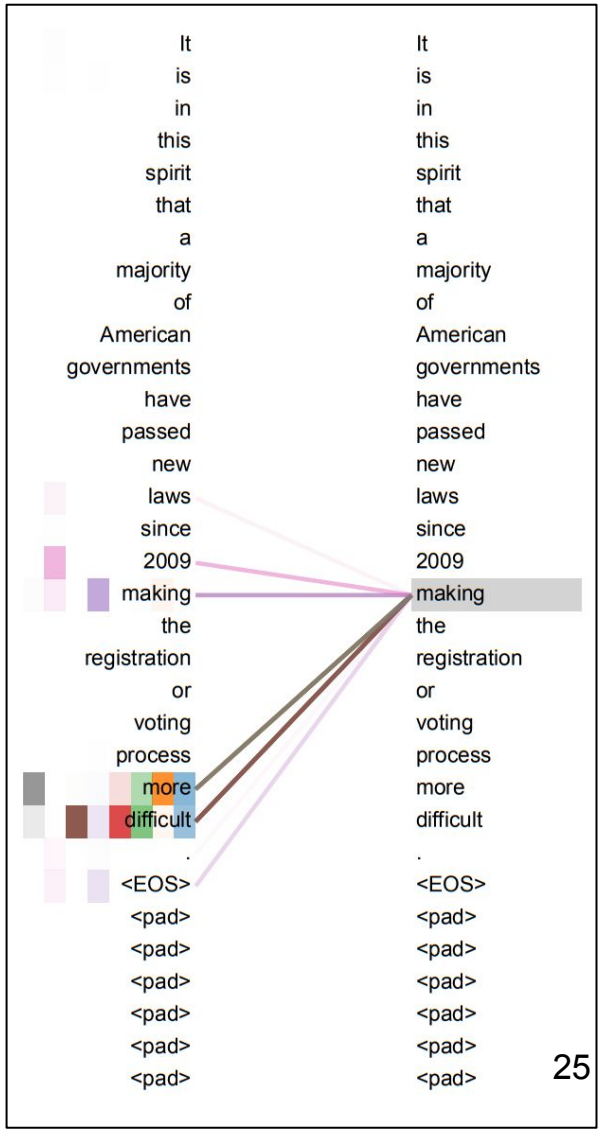
URBAN SYSTEMS

MODEL CONSTRUCTION

TOKENIZATION

- Words
- Subwords
- Characters
- Punctuation
- Special Tokens



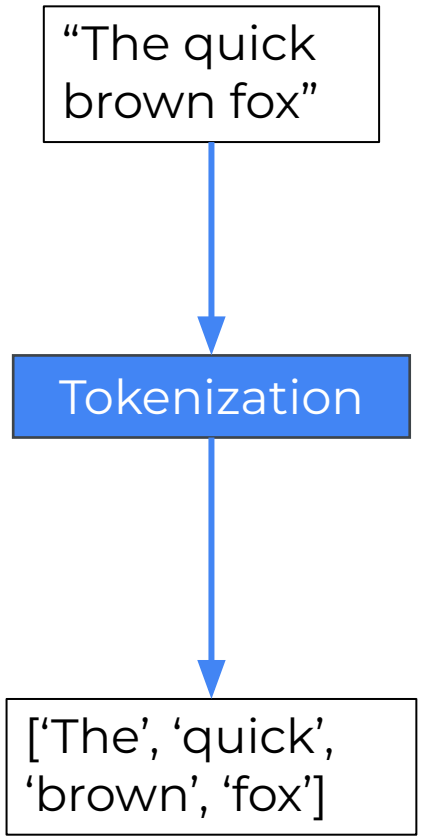


Useful Resources

- [tool] OpenAI Tokenizer
- [tool] Token Visualizer

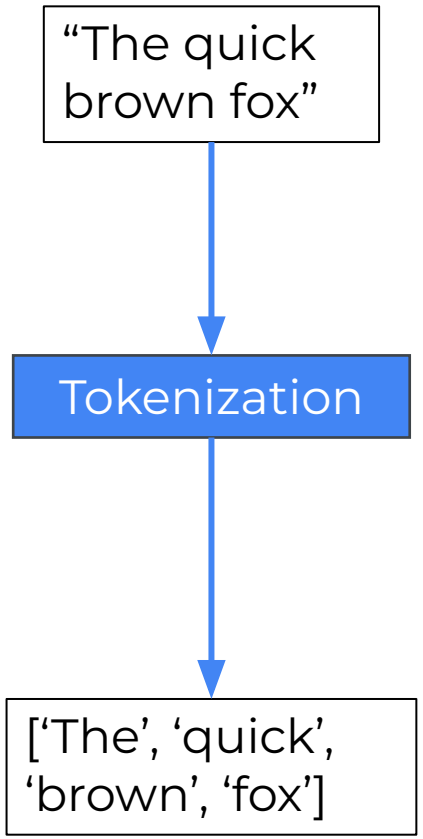
MODEL CONSTRUCTION

Tokenization

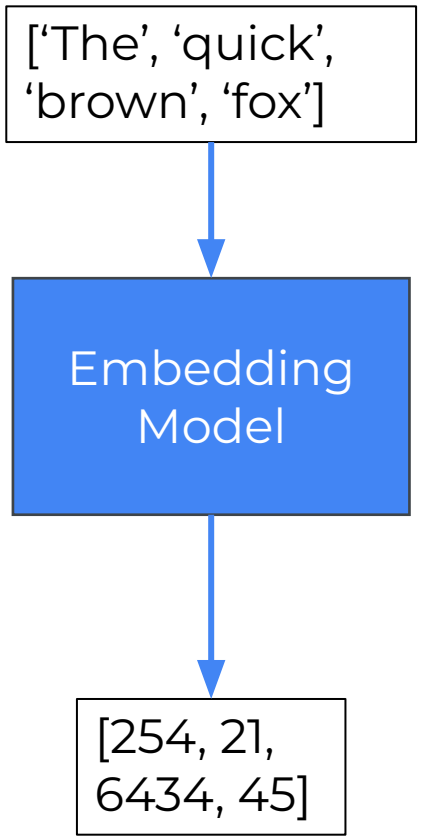


MODEL CONSTRUCTION

Tokenization

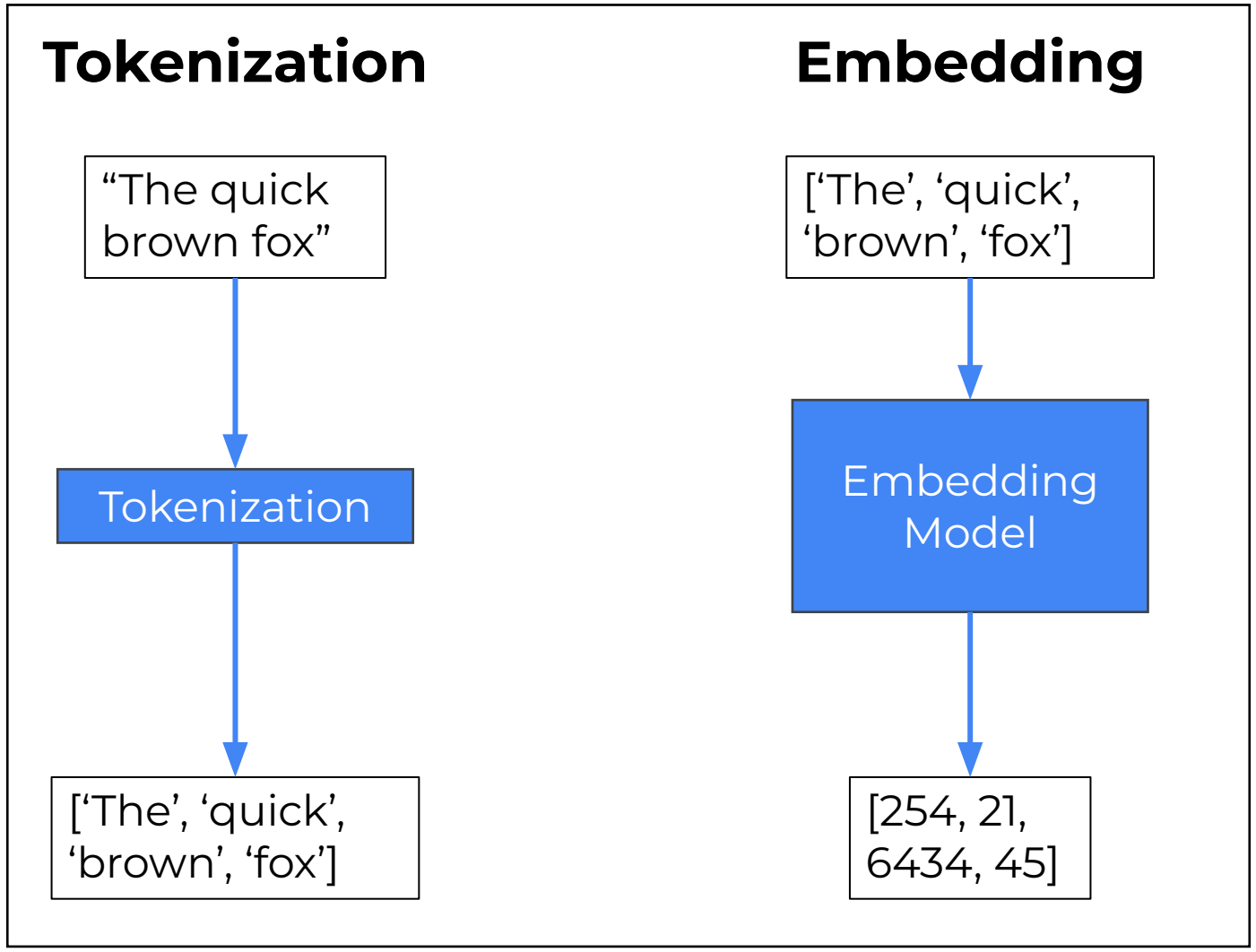


Embedding



URBAN SYSTEMS

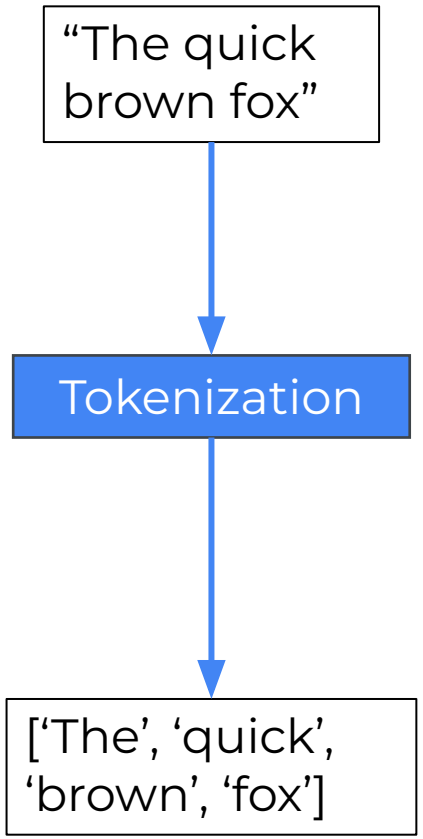
MODEL CONSTRUCTION



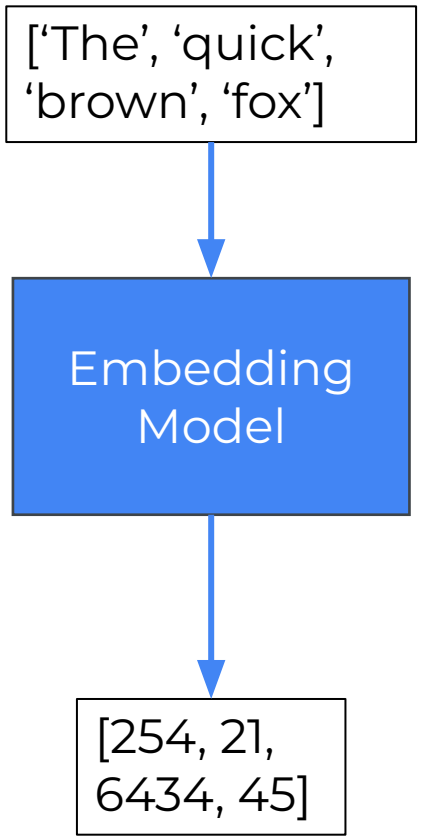
Encoding Process

MODEL CONSTRUCTION

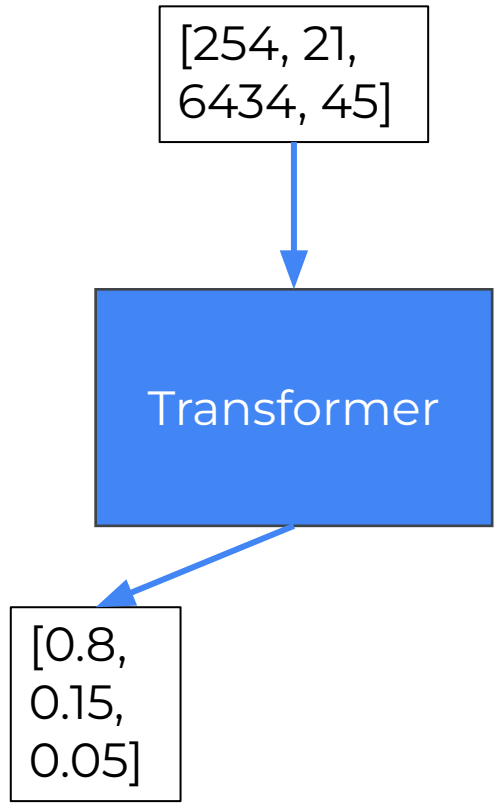
Tokenization



Embedding



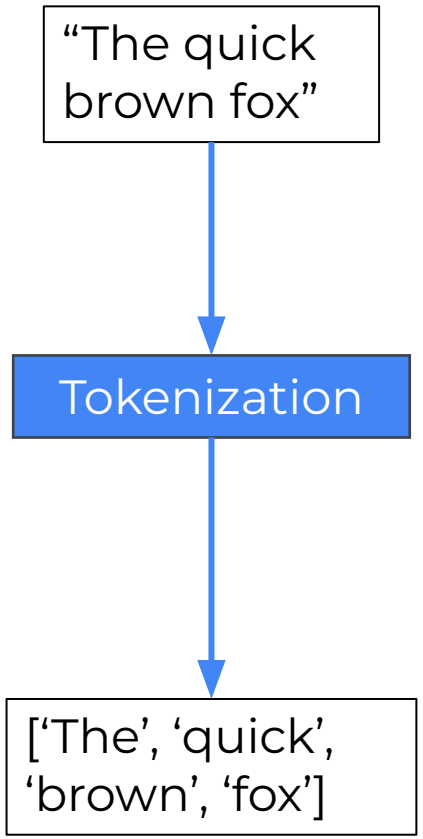
Transformer



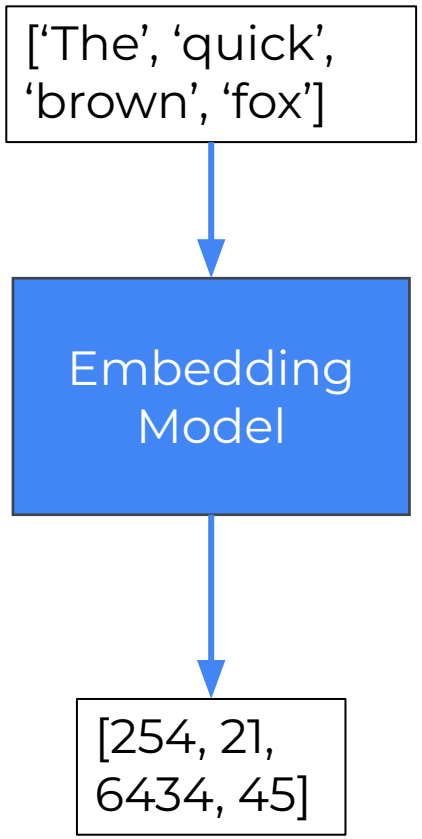
URBAN SYSTEMS

MODEL CONSTRUCTION

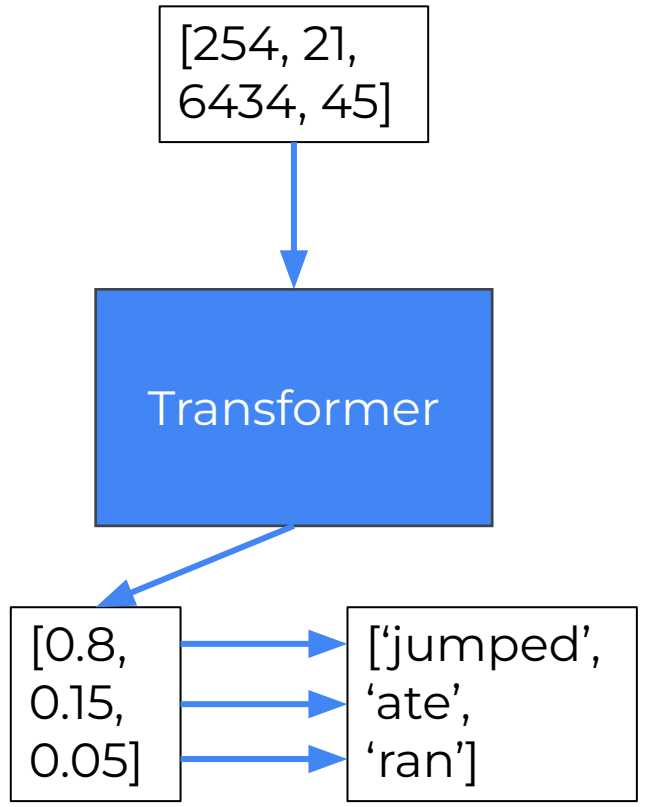
Tokenization



Embedding



Transformer



URBAN SYSTEMS

MODEL CONSTRUCTION

TRANSFORMER ARCHITECTURE

- Compared to historical models (RNN, LSTMs), they're much faster
  - CPU vs GPU

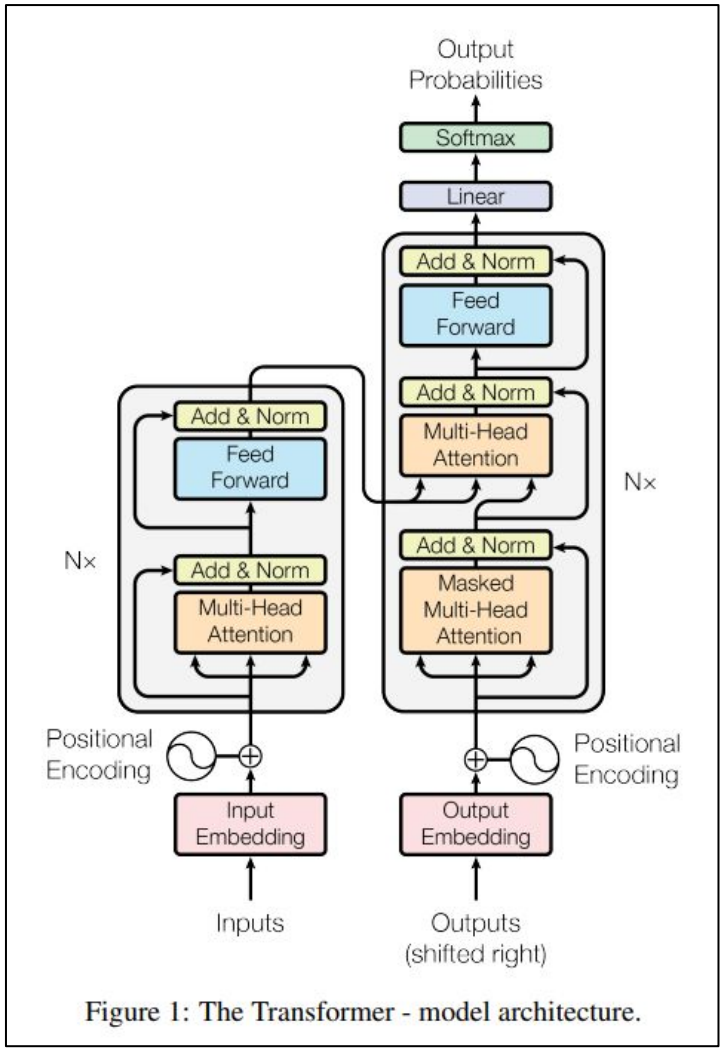


Figure 1: The Transformer - model architecture.

Useful Resources

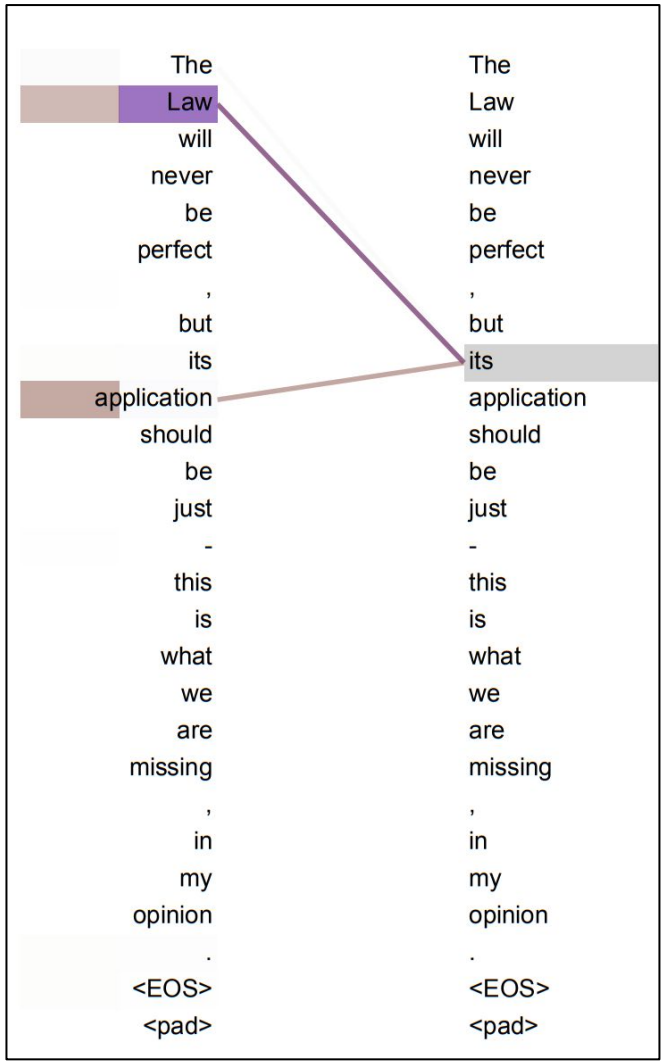
- [arxiv] Attention Is All You Need
- [youtube] Large Language Models explained briefly
- [tool] Transformer Explained

URBAN SYSTEMS

MODEL CONSTRUCTION

TRANSFORMER ARCHITECTURE

- Compared to historical models (RNN, LSTMs), they're much faster
  - CPU vs GPU
- Attention Mechanism: Understands relationships between words



Attention Visualization, self attention layer in layer 5 of 6

Figure 4 from Attention is All You Need

Useful Resources

- [arxiv] Attention Is All You Need
- [youtube] Large Language Models explained briefly
- [tool] Transformer Explained

URBAN SYSTEMS

MODEL CONSTRUCTION

TRANSFORMER ARCHITECTURE

- Compared to historical models (RNN, LSTMs), they're much faster
  - CPU vs GPU
- Attention Mechanism: Understands relationships between words
- Tokenization: Operates on tokens (not words)
  - Numerical representation of a word, character, or sub-word

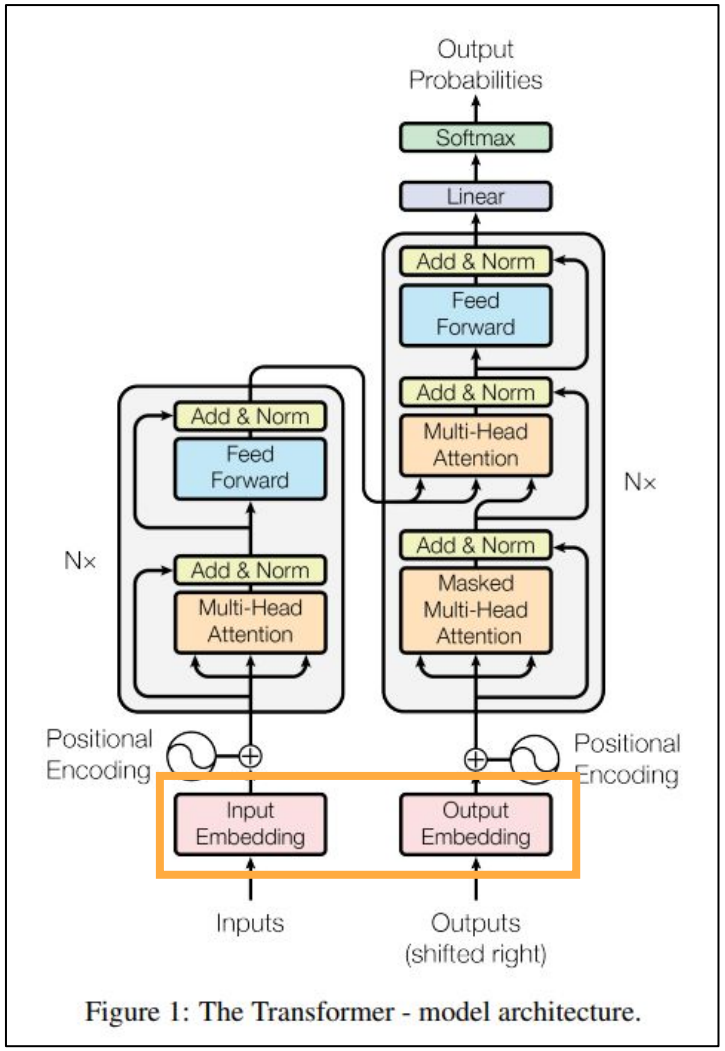


Figure 1: The Transformer - model architecture.

Useful Resources

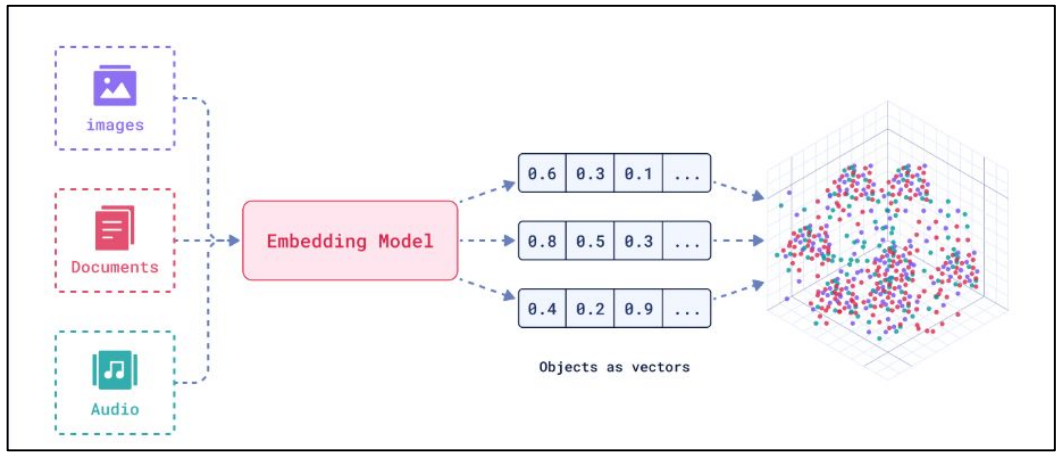
- [arxiv] Attention Is All You Need
- [youtube] Large Language Models explained briefly
- [tool] Transformer Explained

URBAN SYSTEMS

MODEL CONSTRUCTION

TRANSFORMER ARCHITECTURE

- Compared to historical models (RNN, LSTMs), they're much faster
  - CPU vs GPU
- Attention Mechanism: Understands relationships between words
- Tokenization: Operates on tokens (not words)
  - Numerical representation of a word, character, or sub-word



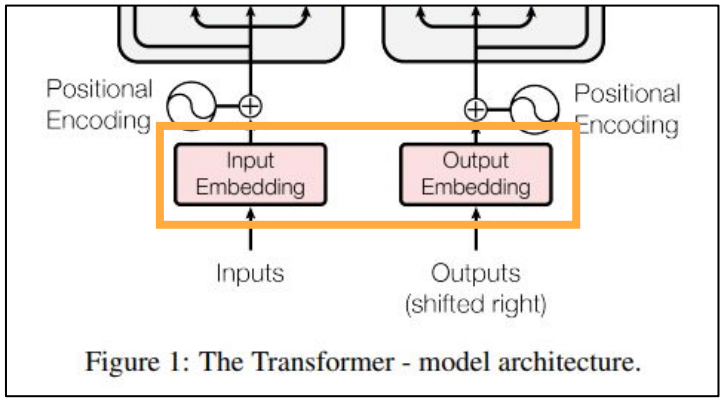


Figure 1: The Transformer - model architecture.

Useful Resources

- [arxiv] Attention Is All You Need
- [youtube] Large Language Models explained briefly
- [tool] Transformer Explained

URBAN SYSTEMS

MODEL DESIGN

MODEL TYPES

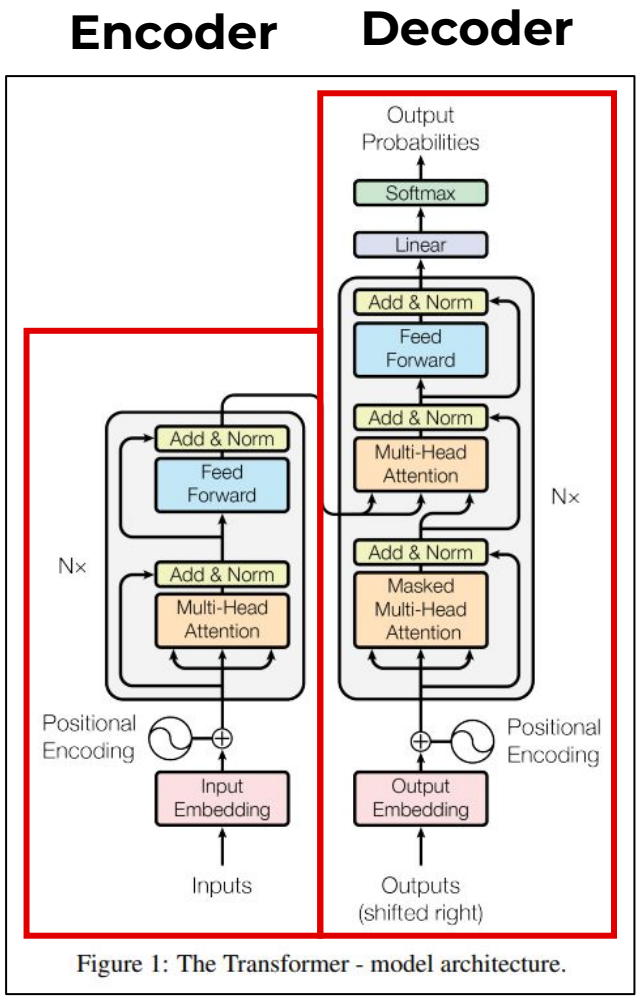


Figure 1: The Transformer - model architecture.

MODEL DESIGN

MODEL TYPES

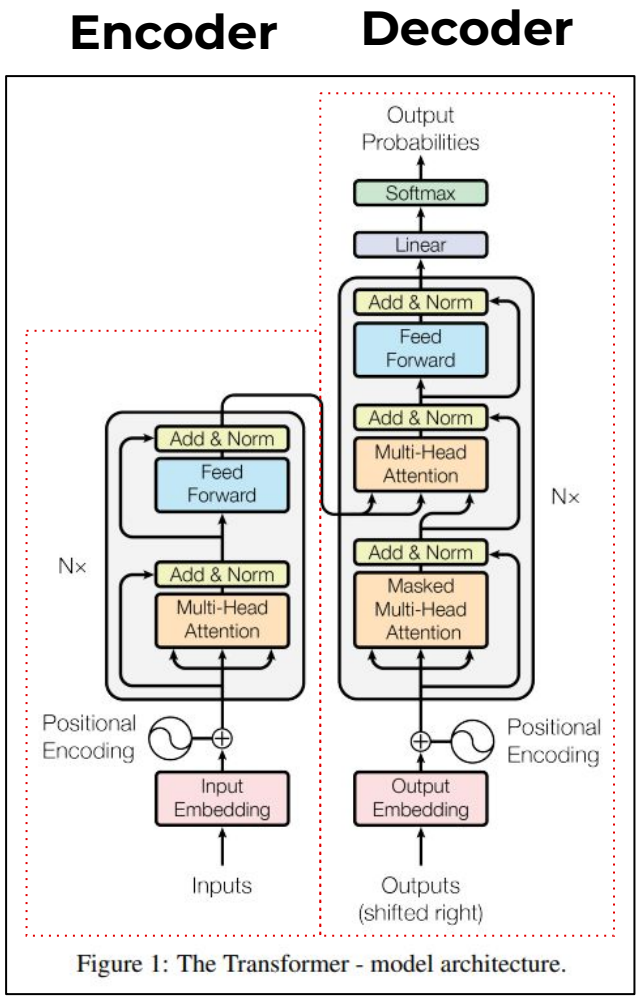


Figure 1: The Transformer - model architecture.

MODEL DESIGN

MODEL TYPES

- Encoder vs Decoder
  - **Encoder-Decoder:** output heavily relies on input, good for translation or summarization tasks. Models: T5, BART

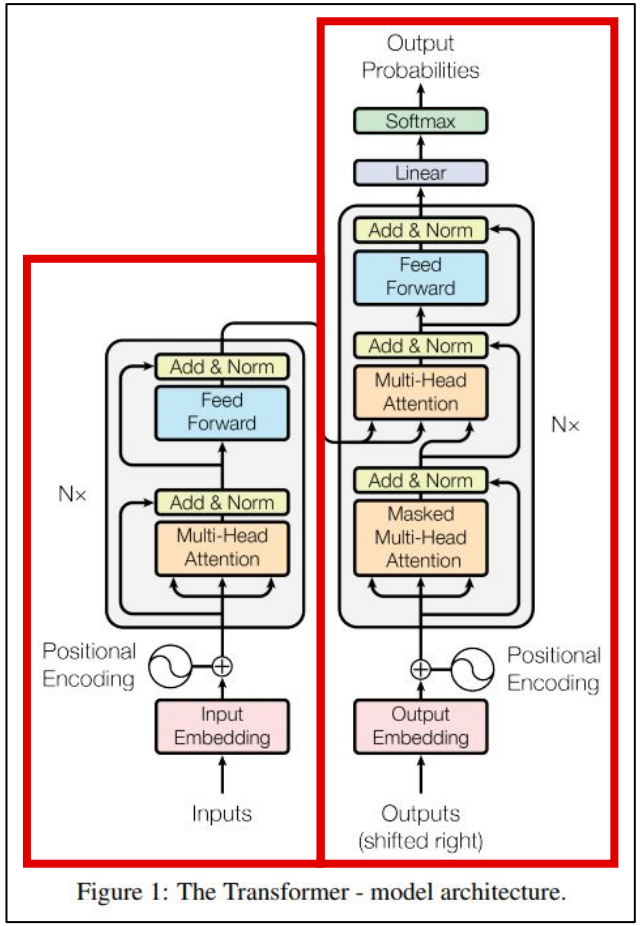


Figure 1: The Transformer - model architecture.

MODEL DESIGN

MODEL TYPES

- Encoder vs Decoder
  - **Encoder-Decoder:** output heavily relies on input, good for translation or summarization tasks. Models: T5, BART
  - **Encoder-only:** Encodes input sequence, optimized for understanding but does not generate output, good for text classification, named entity recognition. Models: BERT

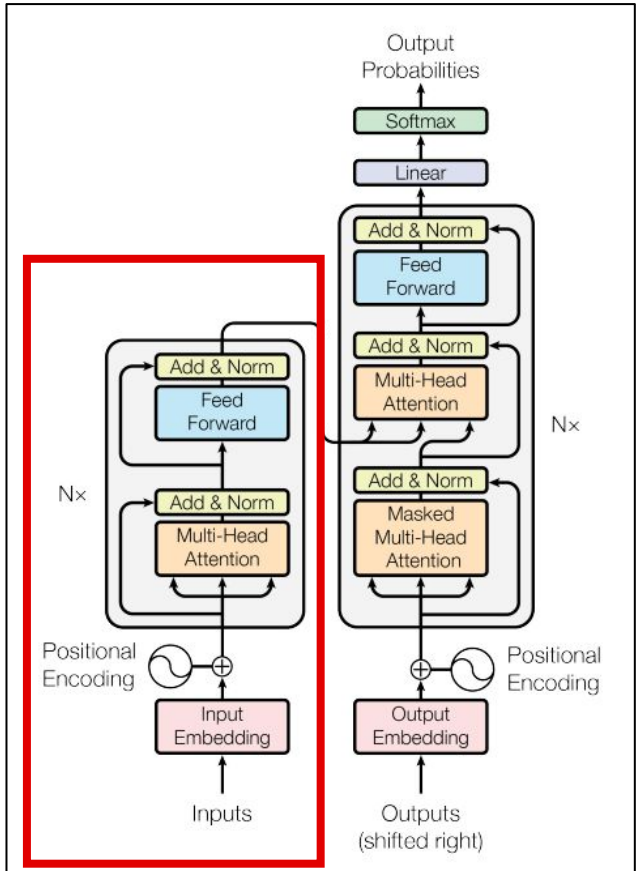


Figure 1: The Transformer - model architecture.

URBAN SYSTEMS

MODEL DESIGN

MODEL TYPES

- Encoder vs Decoder
  - **Encoder-Decoder:** output heavily relies on input, good for translation or summarization tasks. Models: T5, BART
  - **Encoder-only:** Encodes input sequence, optimized for understanding but does not generate output, good for text classification, named entity recognition. Models: BERT
  - **Decoder-only:** Focus on autoregressively generating sequences, good for text generation, open-ended QA. Models: ChatGPT

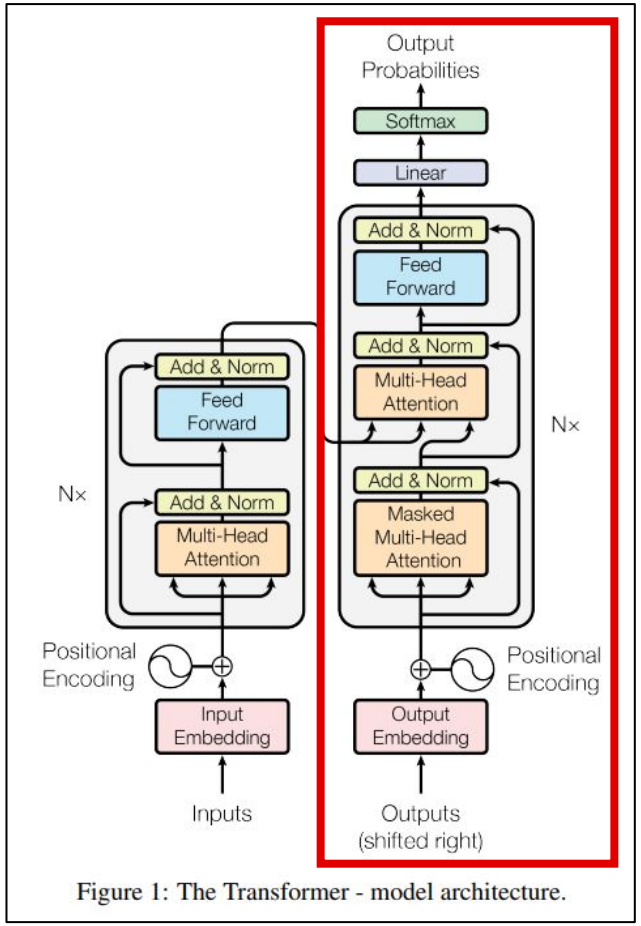
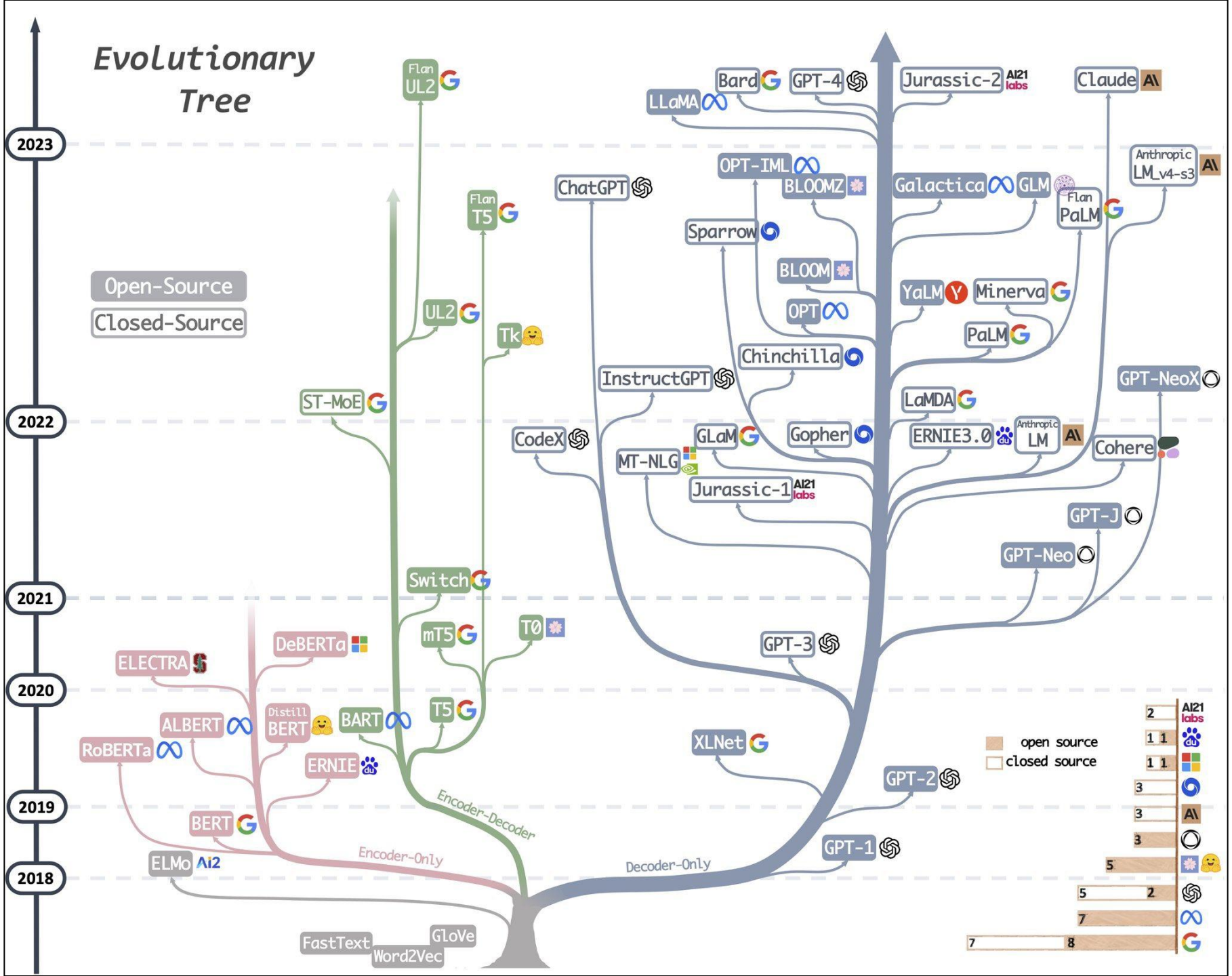


Figure 1: The Transformer - model architecture.


Evolutionary Tree
2023
2022
2021
2020
2019
2018
Open-Source
Closed-Source
Flan UL2
Flan T5
UL2
Tk
ST-MoE
Switch
mT5
T0
DeBERTa
ELECTRA
ALBERT
Distill BERT
BART
T5
RoBERTa
ERNIE
BERT
ELMo Ai2
FastText
GloVe
Word2Vec
Encoder-Only
Encoder-Decoder
Decoder-Only
LLaMA
Bard
GPT-4
Jurassic-2 AI21 labs
Claude A\
ChatGPT
OPT-IML
BLOOMZ
Galactica
GLM
Anthropic LM_v4-s3
Flan PaLM
Sparrow
BLOOM
YaLM
Minerva
OPT
PaLM
Chinchilla
InstructGPT
GPT-NeoX
LaMDA
CodeX
GLaM
Gopher
ERNIE3.0
Anthropic LM A\
Cohere
MT-NLG
Jurassic-1 AI21 labs
GPT-J
GPT-Neo
GPT-3
XLNet
GPT-2
GPT-1
open source
closed source
2
1 1
1 1
3
3
3
5
5 2
7
7 8

MODEL DESIGN

MODEL TYPES

Base Model vs Chat/Instruct

- Base: Simply next-token prediction
- Instruct: Fine-tuned base model to be more helpful for QA and task completion. Often further refined through reinforcement learning with human feedback (RLHF, post-training)

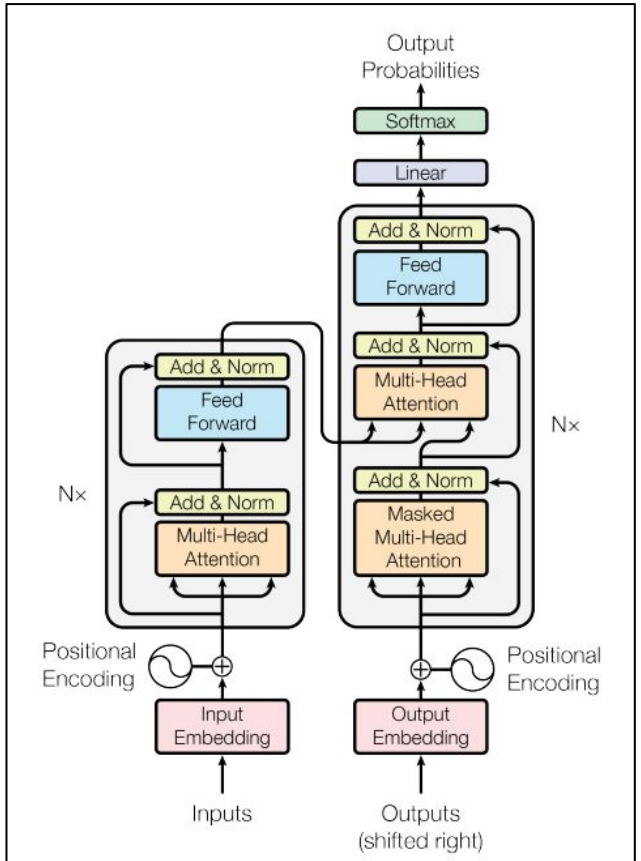


Figure 1: The Transformer - model architecture.

Useful Resources

- [blog] Base LLM vs. Instruction-tuned LLM

URBAN SYSTEMS

MODEL DESIGN

MODEL SIZE

Classification	Model Size	Example
Small	<1B	NanoGPT
Medium	1-10B	Phi-4
Large	10-100B	Llama 3.3
Massive	100B+	GPT 4o

Useful Resources

- [arxiv] Training Compute-Optimal Large Language Models
- [arxiv] A Theory of Emergence of Complex Skills in Language Models
- [arxiv] Skill-Mix: a Flexible and Expandable Family of Evaluations for AI Models

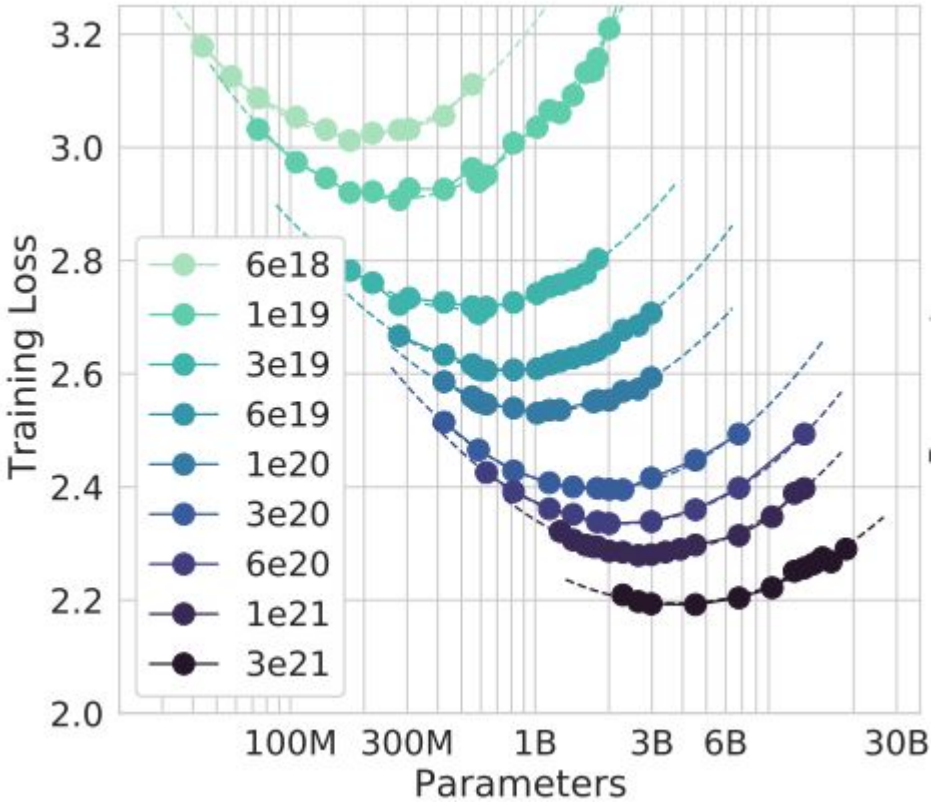
URBAN SYSTEMS

MODEL DESIGN

MODEL SIZE

- Emergent properties from scaling
- There is some relationship between training data and model size
- Scaling laws and diminishing returns?

We can expect more intelligence “for free” by scaling up



Useful Resources

- [arxiv] Training Compute-Optimal Large Language Models
- [arxiv] A Theory of Emergence of Complex Skills in Language Models
- [arxiv] Skill-Mix: a Flexible and Expandable Family of Evaluations for AI Models

URBAN SYSTEMS

PROMPTING STRUCTURE

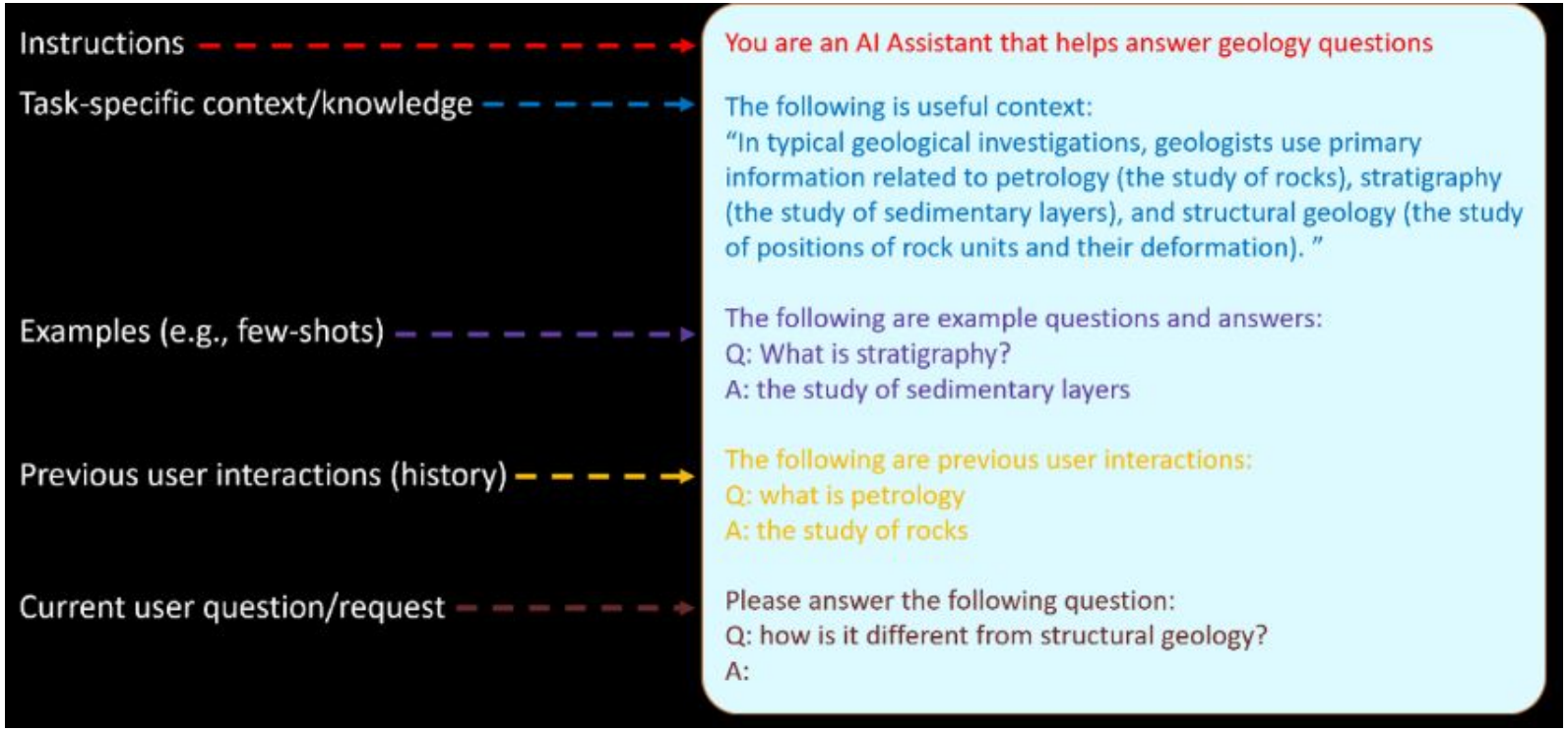
The API of AI

URBAN SYSTEMS

PROMPTING STRUCTURE

PROMPT STRUCTURE

- System
- Human Input
- Assistant/Output



Useful Resources

- [Microsoft] Getting Started with LLM Prompt Engineering
- [huggingface] LLM Prompting Guide

URBAN SYSTEMS

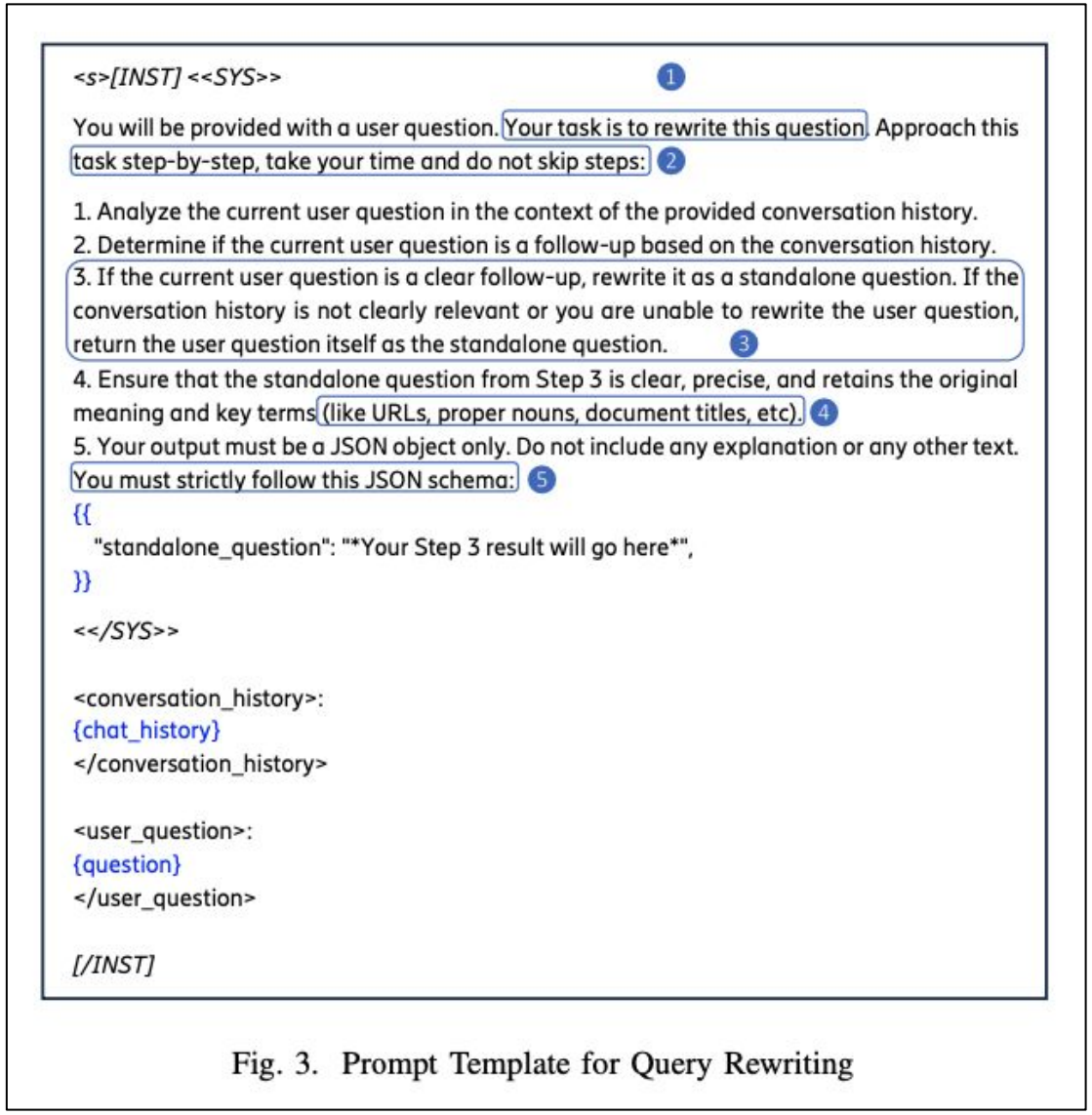
PROMPTING STRUCTURE

PROMPT STRUCTURE

- System
- Human Input
- Assistant/Output

Useful Resources

- [Microsoft] Getting Started with LLM Prompt Engineering
- [huggingface] LLM Prompting Guide



```
<s>[INST] <<SYS>>                                                    1

You will be provided with a user question. Your task is to rewrite this question. Approach this
task step-by-step, take your time and do not skip steps: 2

1. Analyze the current user question in the context of the provided conversation history.
2. Determine if the current user question is a follow-up based on the conversation history.
3. If the current user question is a clear follow-up, rewrite it as a standalone question. If the
conversation history is not clearly relevant or you are unable to rewrite the user question,
return the user question itself as the standalone question.                                    3
4. Ensure that the standalone question from Step 3 is clear, precise, and retains the original
meaning and key terms (like URLs, proper nouns, document titles, etc). 4
5. Your output must be a JSON object only. Do not include any explanation or any other text.
You must strictly follow this JSON schema: 5
{{
  "standalone_question": "*Your Step 3 result will go here*",
}}

<</SYS>>

<conversation_history>:
{chat_history}
</conversation_history>

<user_question>:
{question}
</user_question>

[/INST]
```

Fig. 3. Prompt Template for Query Rewriting

URBAN SYSTEMS

PROMPTING STRUCTURE

PROMPT FORMATS

- If prompting through an API or Web App, this is handled for you.
- If you're interfacing with models locally and directly, you will have to structure this yourself

```
<|begin_of_text|><|start_header_id|>system<|end_header_id|>

Cutting Knowledge Date: December 2023
Today Date: 23 July 2024

You are a helpful assistant<|eot_id|><|start_header_id|>user<|end_header_id|>

What is the capital of France?<|eot_id|><|start_header_id|>assistant<|end_header_id|>
```

Useful Resources

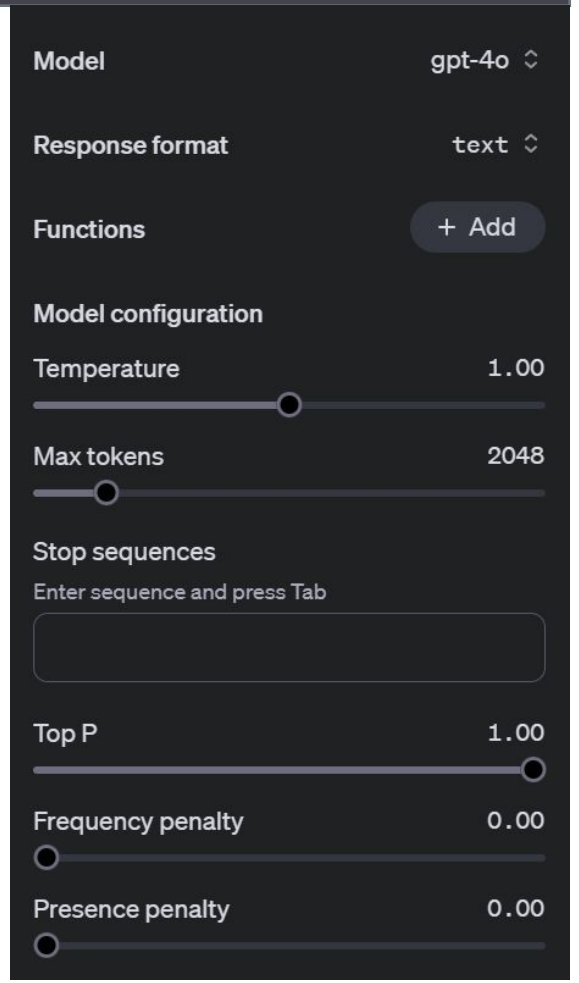
- [docs] Meta Llama 3.1 Prompt Format
- [docs] Meta Llama 2 Prompt Format
- [huggingface] LLM Prompting Guide

URBAN SYSTEMS

PROMPTING STRUCTURE

MODEL PARAMETERS

- **Temperature:** A control of **creativity** or randomness in text generation.
  - Higher = more creative
  - Lower = more deterministic
- **Top K:** LLM selects from top K most likely tokens
  - Higher = more diversity
  - Lower = more fluency
- **Top P:** retains tokens with a total cumulative probability above P
  - Higher = more diversity
  - Lower = less randomness
- **Max Tokens:** max tokens in model output



Useful Resources

- [blog] What is Temperature in NLP?
- [blog] Tuning LLM Parameters for Optimal Performance
- [youtube] Top-P vs Top-K

URBAN SYSTEMS

EXERCISE

https://tinyurl.com/uc-llm-workshop

Prompting Notebook
https://colab.research.google.com/drive/1YH_Cr6N6qgYGUu26McQa086xvern61p0?usp=sharing

Fine-Tuning Notebook
https://colab.research.google.com/drive/1Nz1t1LnCPiXrkAyQcPG61VTPziFJcuLM?usp=sharing

URBAN SYSTEMS

PROMPTING TECHNIQUES

FEW-SHOT PROMPTING/IN-CONTEXT LEARNING

- “Shot” is synonymous with examples
- Can tweak for style/tone of writing by providing examples
- Can improve performance for outputting specific formats
- Can be used for creative writing, formatting, data transformations, reasoning tasks, etc…
- Order of examples maybe matters
- Negative examples likely confuse models

Useful Resources

- [arxiv] Language Models are Few-Shot Learners
- [arxiv] Lost in the Middle: How Language Models Use Long Contexts

URBAN SYSTEMS

PROMPTING TECHNIQUES

FEW-SHOT PROMPTING/IN-CONTEXT LEARNING

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
```

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
```

Useful Resources

- [arxiv] Language Models are Few-Shot Learners
- [arxiv] Lost in the Middle: How Language Models Use Long Contexts

URBAN SYSTEMS

PROMPTING TECHNIQUES

CHAIN-OF-THOUGHT

- Slow down silly AI!
- Helps break down complex tasks, improves performance for reasoning/logic based tasks
- Incited through the phrase, “Lets think step by step”, or similar
- LLMs are sensitive to semantics, each model is different

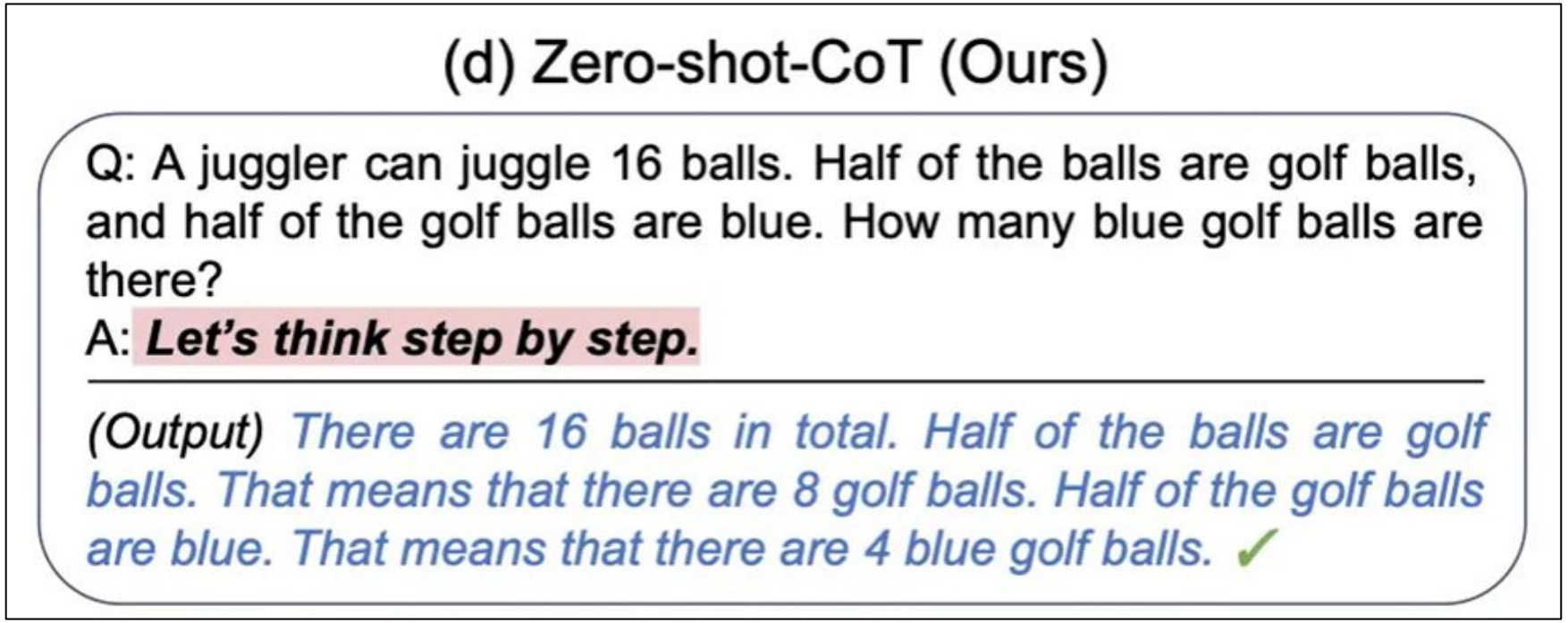
Useful Resources

- [arxiv] Large Language Models are Zero-Shot Reasoners
- [arxiv] Large Language Models as Optimizers

URBAN SYSTEMS

PROMPTING TECHNIQUES

CHAIN-OF-THOUGHT



Useful Resources

- [arxiv] Large Language Models are Zero-Shot Reasoners
- [arxiv] Large Language Models as Optimizers

URBAN SYSTEMS

PROMPTING TECHNIQUES

CHAIN-OF-THOUGHT

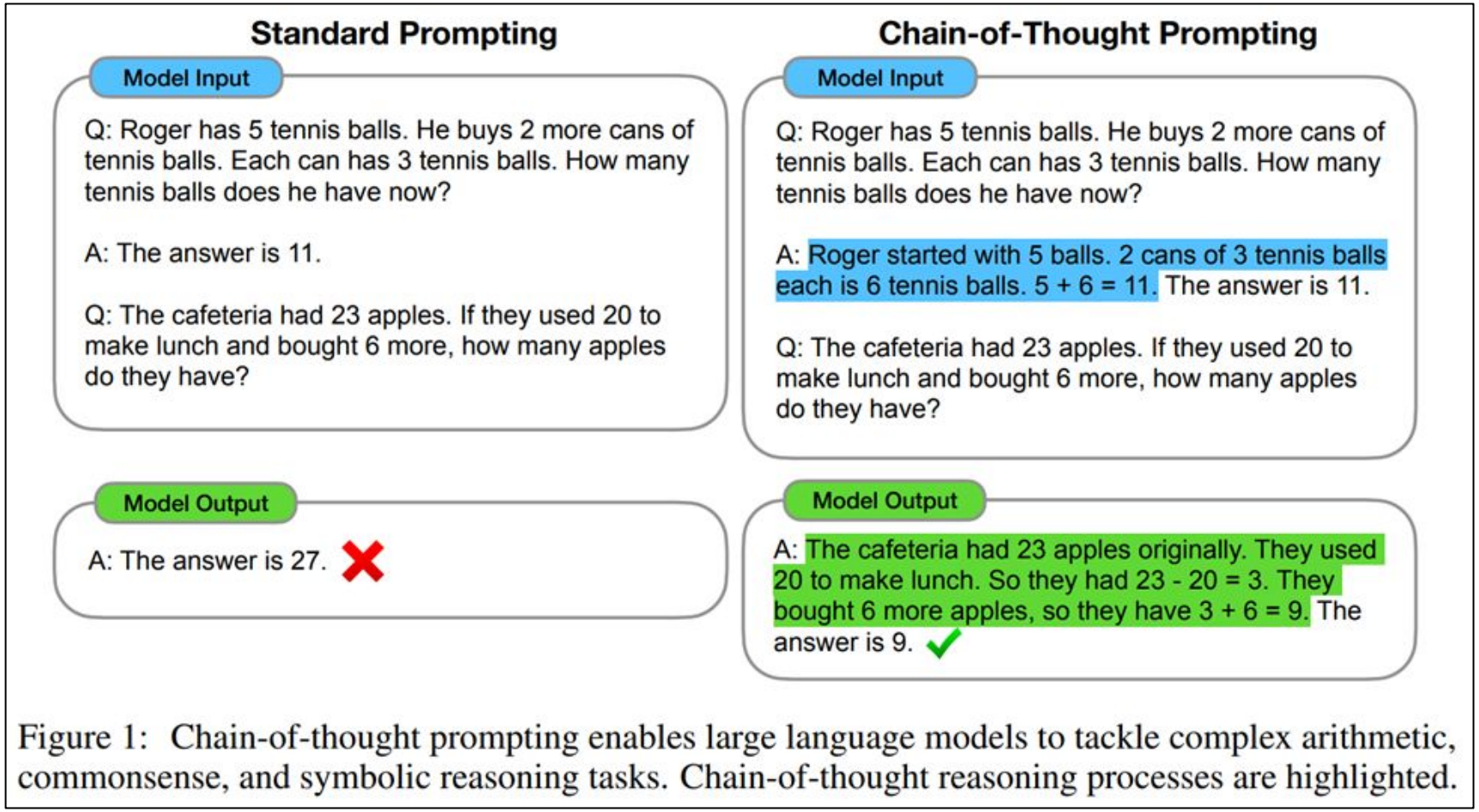


Figure 1: Chain-of-thought prompting enables large language models to tackle complex arithmetic, commonsense, and symbolic reasoning tasks. Chain-of-thought reasoning processes are highlighted.

URBAN SYSTEMS

PROMPTING TECHNIQUES

CHAIN-OF-THOUGHT

Scorer	Optimizer / Source	Instruction position	Top instruction	Acc
Baselines				
PaLM 2-L	(Kojima et al., 2022)	A_begin	Let's think step by step.	71.8
PaLM 2-L	(Zhou et al., 2022b)	A_begin	Let's work this out in a step by step way to be sure we have the right answer.	58.8
PaLM 2-L		A_begin	Let's solve the problem.	60.8
PaLM 2-L		A_begin	(empty string)	34.0
text-bison	(Kojima et al., 2022)	Q_begin	Let's think step by step.	64.4
text-bison	(Zhou et al., 2022b)	Q_begin	Let's work this out in a step by step way to be sure we have the right answer.	65.6
text-bison		Q_begin	Let's solve the problem.	59.1
text-bison		Q_begin	(empty string)	56.8
Ours				
PaLM 2-L	PaLM 2-L-IT	A_begin	Take a deep breath and work on this problem step-by-step.	**80.2**
PaLM 2-L	PaLM 2-L	A_begin	Break this down.	79.9
PaLM 2-L	gpt-3.5-turbo	A_begin	A little bit of arithmetic and a logical approach will help us quickly arrive at the solution to this problem.	78.5

URBAN SYSTEMS

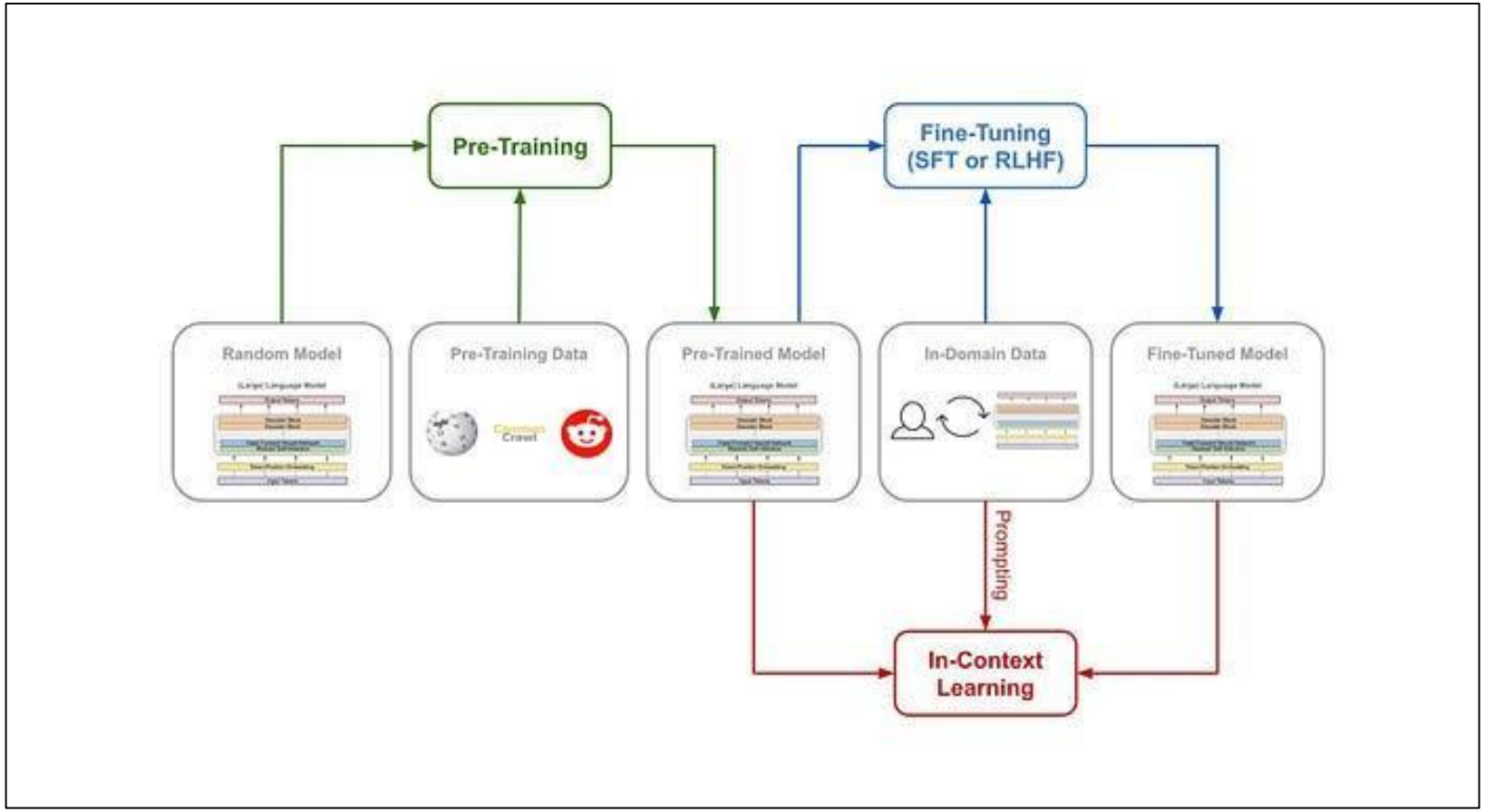
MODEL TRAINING

Getting the most from a query

MODEL TRAINING

PRE-TRAINING

- What can data look like?
- Variety of sources and domains
  - Books
  - Wikipedia
  - News Article
  - Scientific Papers
  - Code Repositories
  - Reddit
  - Dialogues/Conversations
  - Etc…



Useful Resources

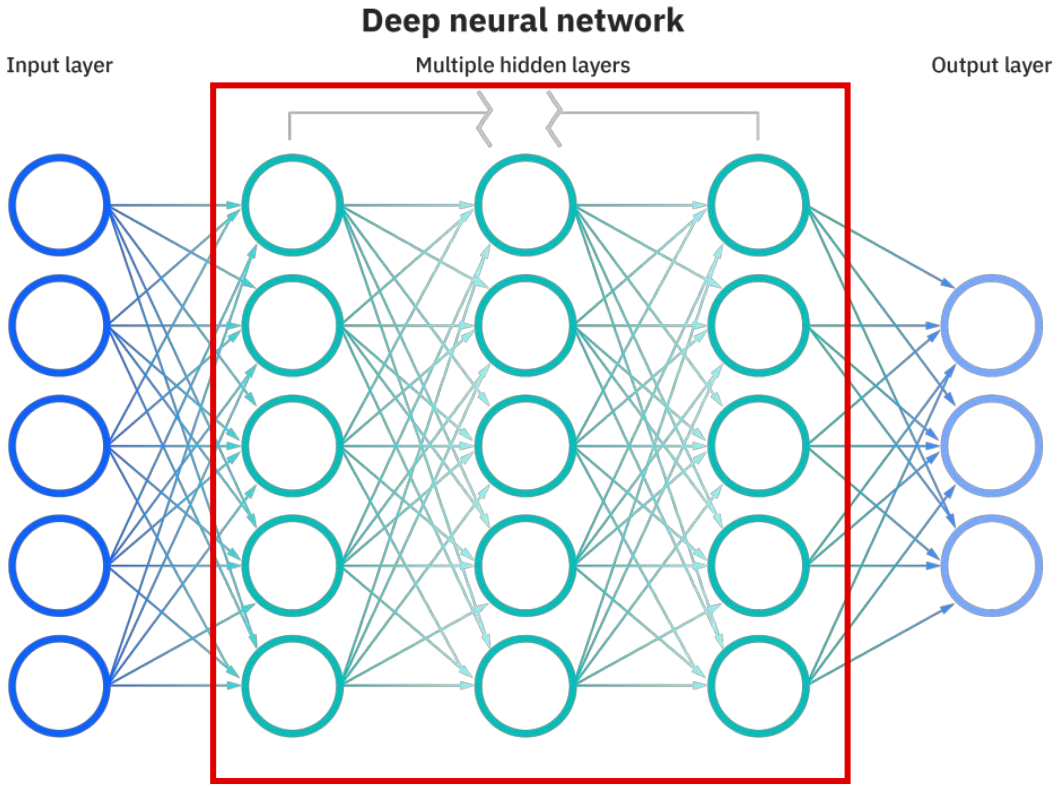
- [blog] Empowering Language Models: Pre-training, Fine-tuning, and In-Context Learning
- [youtube] Intro to Large Language Models – Andrej Karpathy

URBAN SYSTEMS

MODEL TRAINING

FINE-TUNING / TRANSFER LEARNING

- Can fine-tune all the weights



Useful Resources

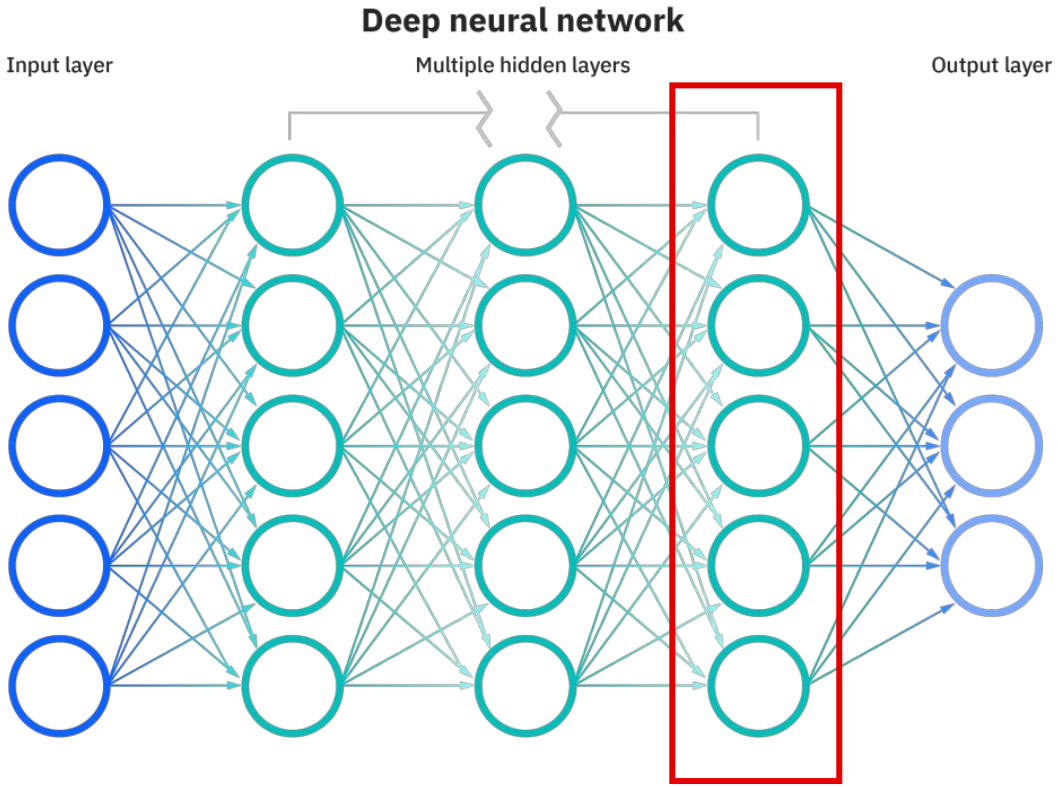
- [youtube] LoRA for fine-tuning LLM models
- [arxiv] LoRA: Low-Rank Adaptation of Large Language Models

URBAN SYSTEMS

MODEL TRAINING

FINE-TUNING / TRANSFER LEARNING

- Can fine-tune all the weights
- Can freeze different sections



Useful Resources

- [youtube] LoRA for fine-tuning LLM models
- [arxiv] LoRA: Low-Rank Adaptation of Large Language Models

URBAN SYSTEMS

MODEL TRAINING

FINE-TUNING / TRANSFER LEARNING

- Can fine-tune all the weights
- Can freeze different sections
- Can add adapter layers

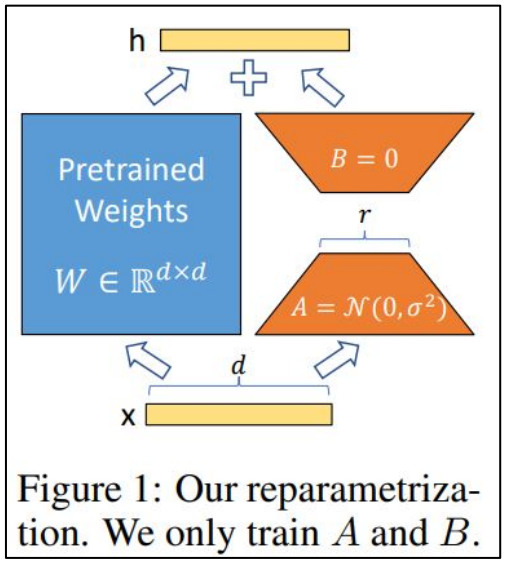
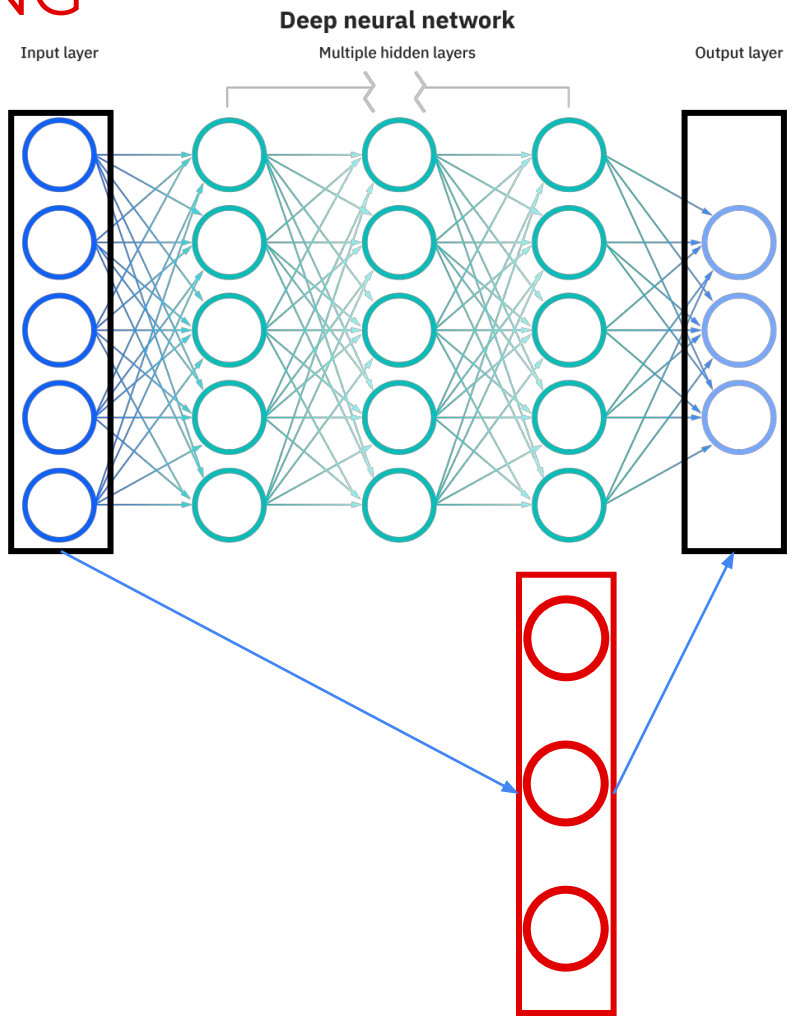


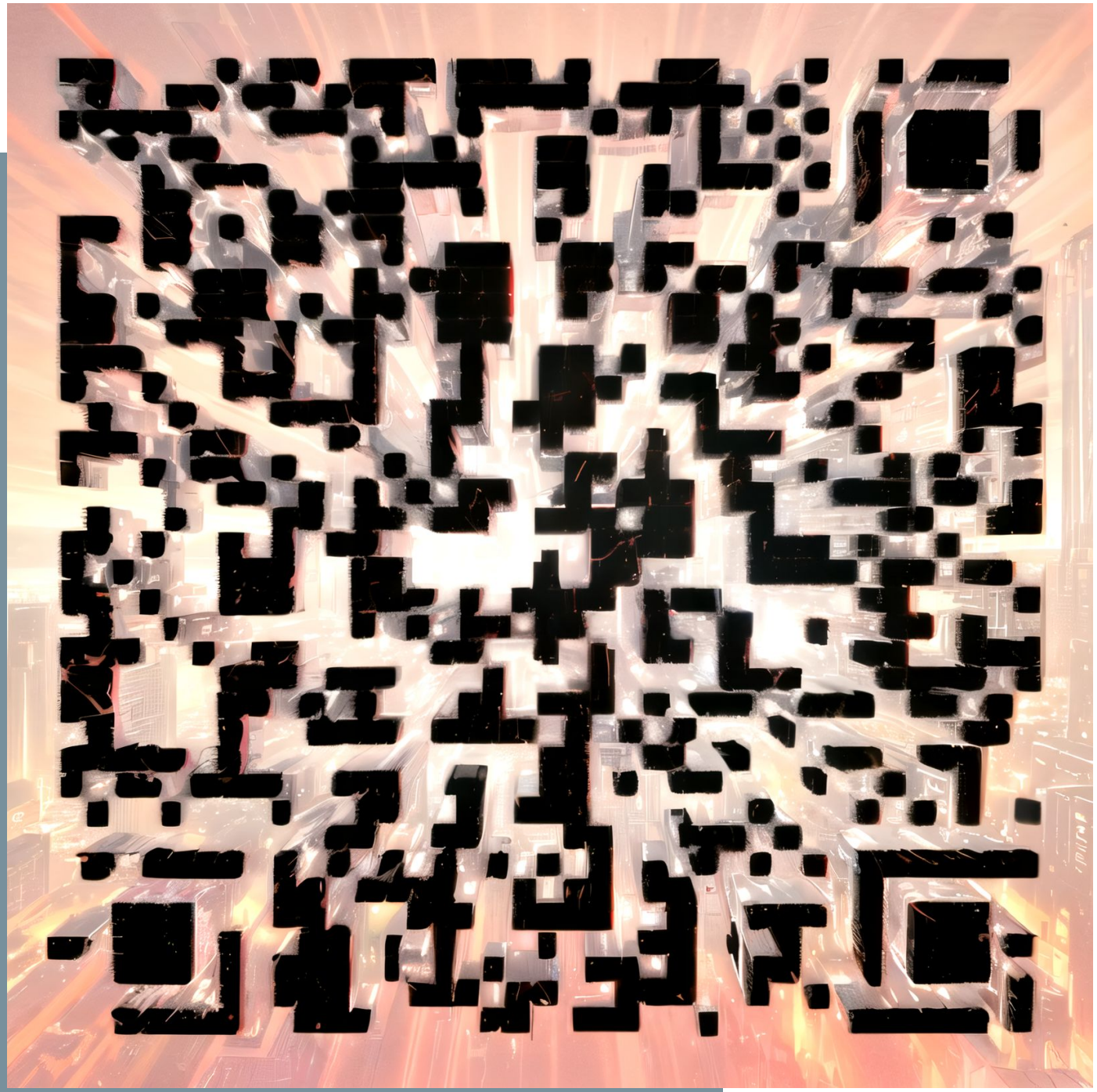
Figure 1: Our reparametrization. We only train A and B.



Useful Resources

- [youtube] LoRA for fine-tuning LLM models
- [arxiv] LoRA: Low-Rank Adaptation of Large Language Models

URBAN SYSTEMS



COREY YANG-SMITH
SOFTWARE AND MACHINE LEARNING DEVELOPER
Email: cyangsmith@urbansystems.ca
LinkedIn: in/yangsmithcorey

MEHREEN AKMAL
SOFTWARE AND MACHINE LEARNING DEVELOPER
Email: makmal@urbansystems.ca
LinkedIn: in/mehreen-akmal

URBAN SYSTEMS