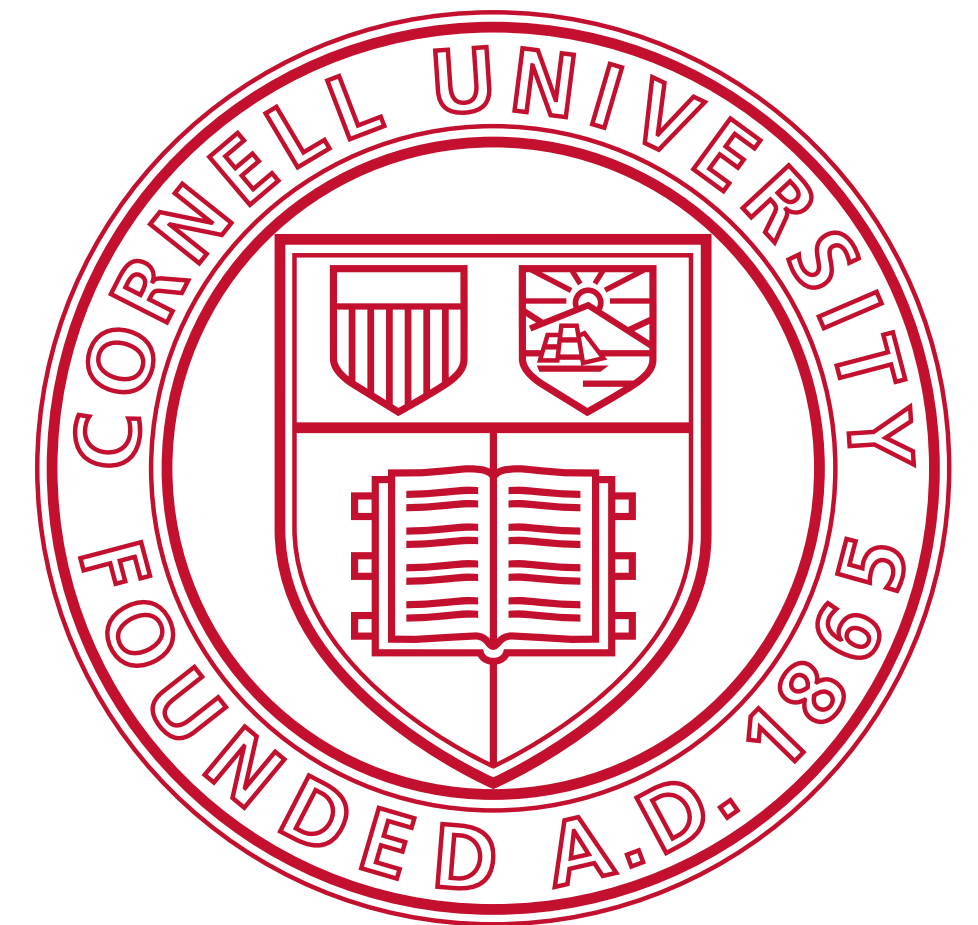


Large Language Models: Principles and Practice

Immanuel Trummer



How do You Know a Human Wrote This?
The New York Times, 2020

Meet GPT-3. It Has Learned to Code (and Blog and Argue).
The New York Times, 2020

An A.I. bot answers 10 burning questions about the 2020 NFL season
USA Today, 2020

The Jessica Simulation: Love and loss in the age of A.I.
The San Francisco Chronicle, 2021

Bringing People Back to Life With the Power of AI Chatbots.
Forbes, 2021

Meet GPT-3, the natural-language system that generates tweets, pens poetry, summarizes emails, answers trivia, translates languages and even writes its own computer programs
Chicago Tribune, 2021

‘Grassroots’ bot campaigns are coming. Governments don’t have a plan to stop them.
Washington Post, 2021

ChatGPT Could be AI's iPhone Moment
Bloomberg, 2022

The New Chatbots Could Change the World.
The New York Times, 2022

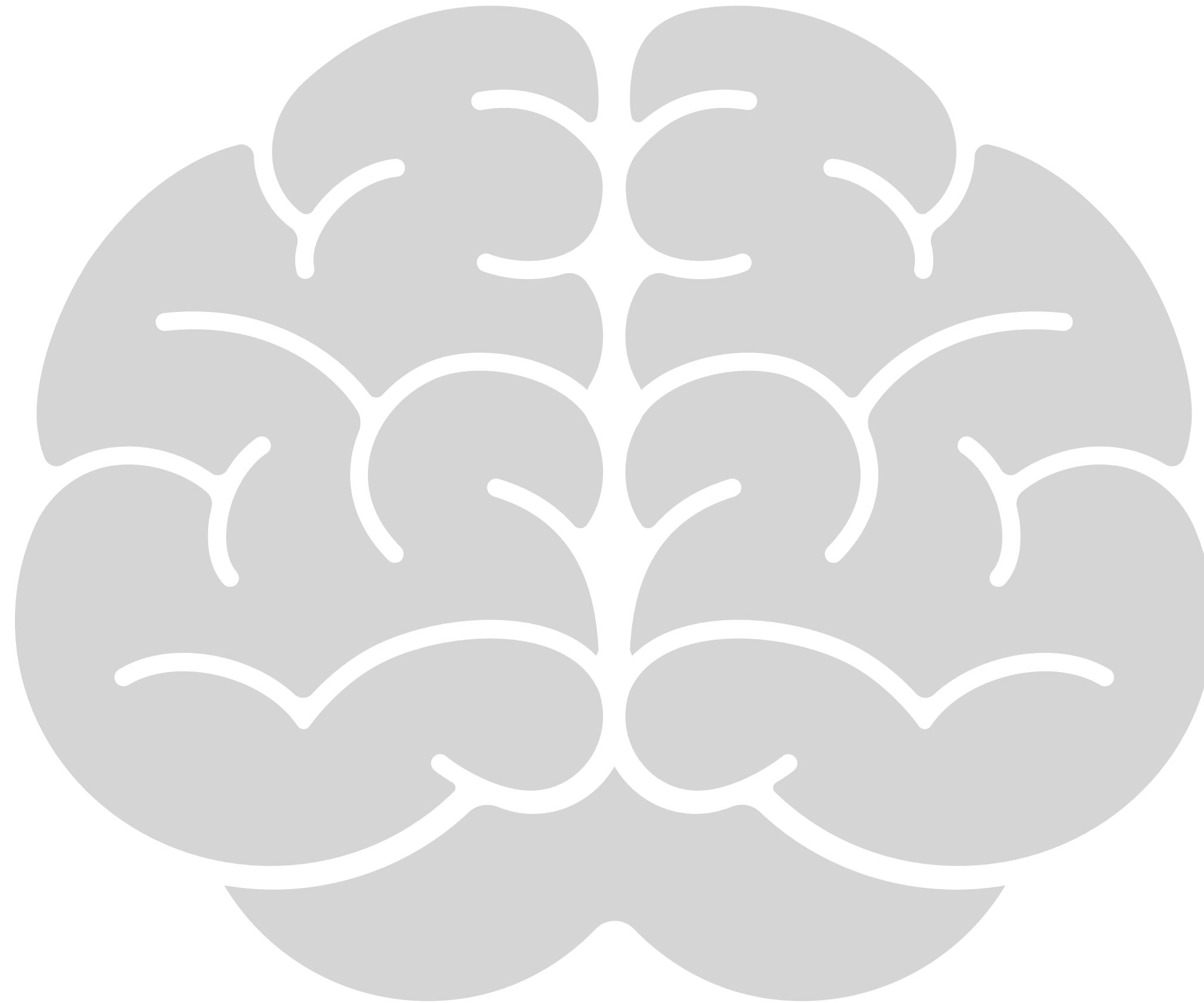
ChatGPT Gained One Million Users in Under a Week. Here's Why the Chatbot is Primed to Disrupt Search as We Know It.
Fortune, 2022

ChatGPT and How AI Disrupts Industries.
Harvard Business Review, 2022

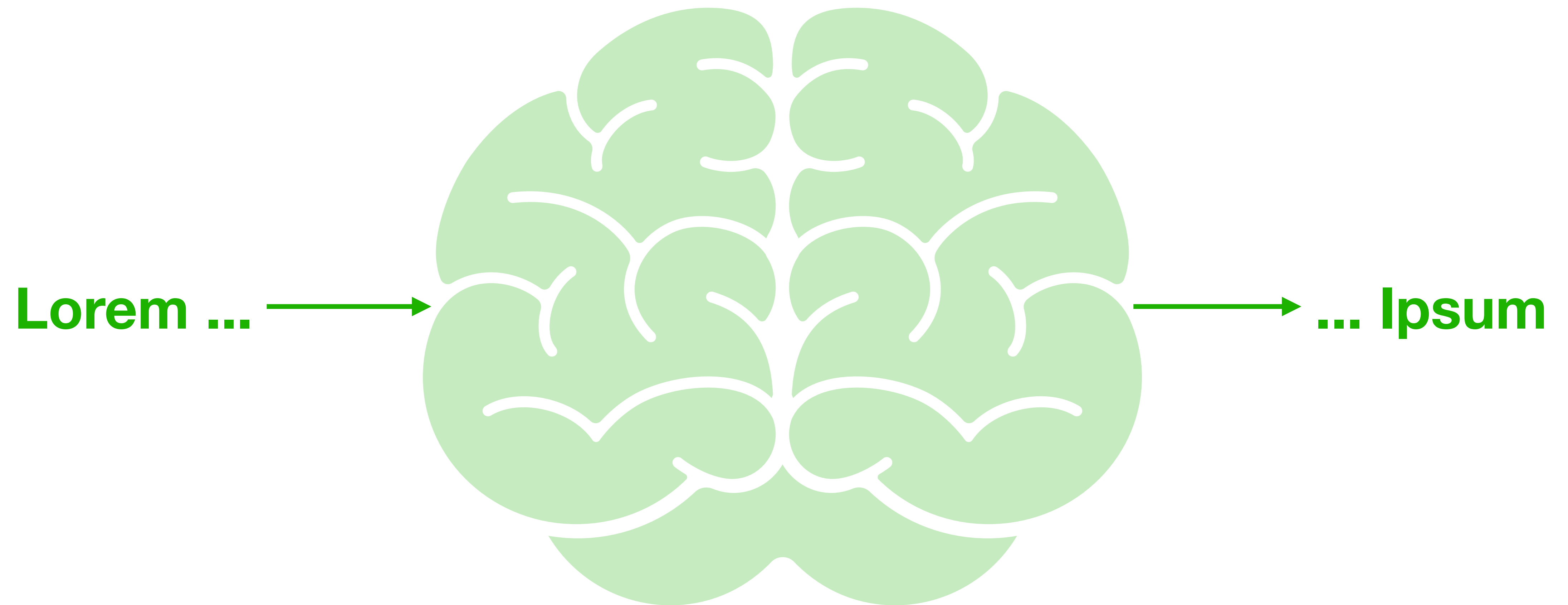
Recent NLP Advances

- **Transformer** architecture
 - More scalable than prior approaches
- **Transfer** learning
 - From data-rich to data-sparse problems

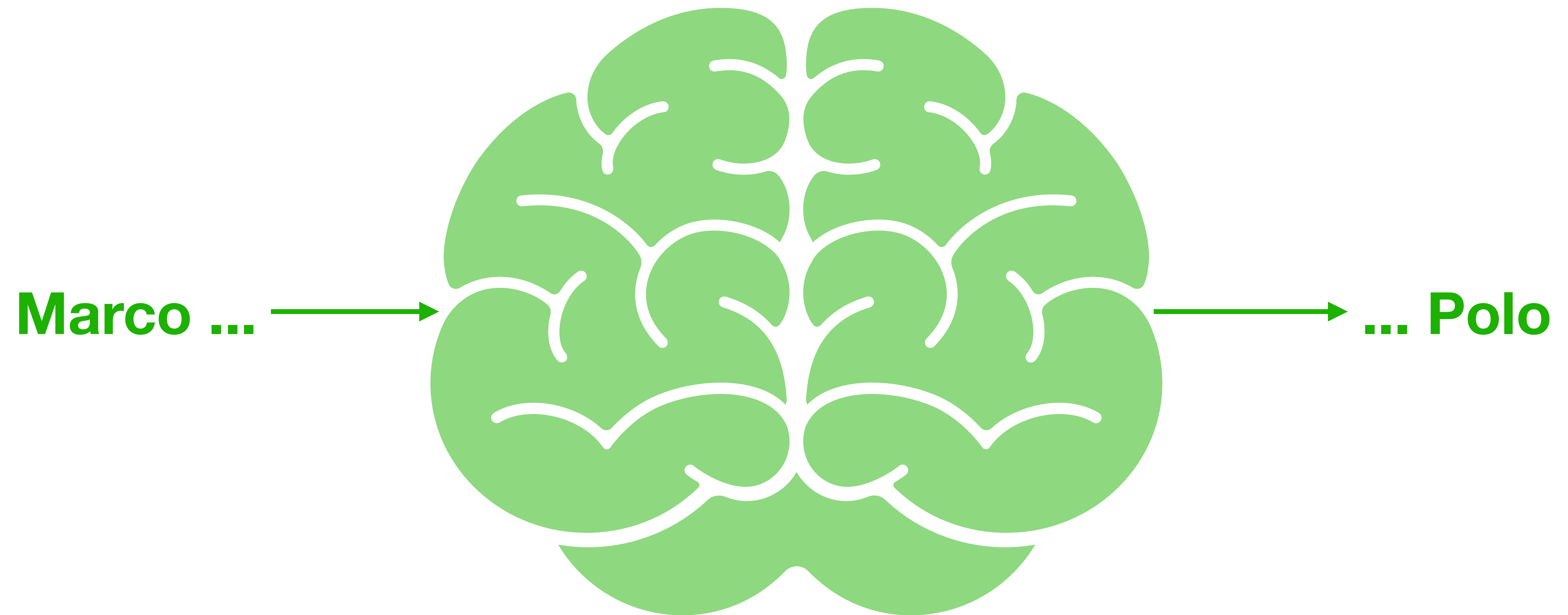
Pre-Training



Pre-Training



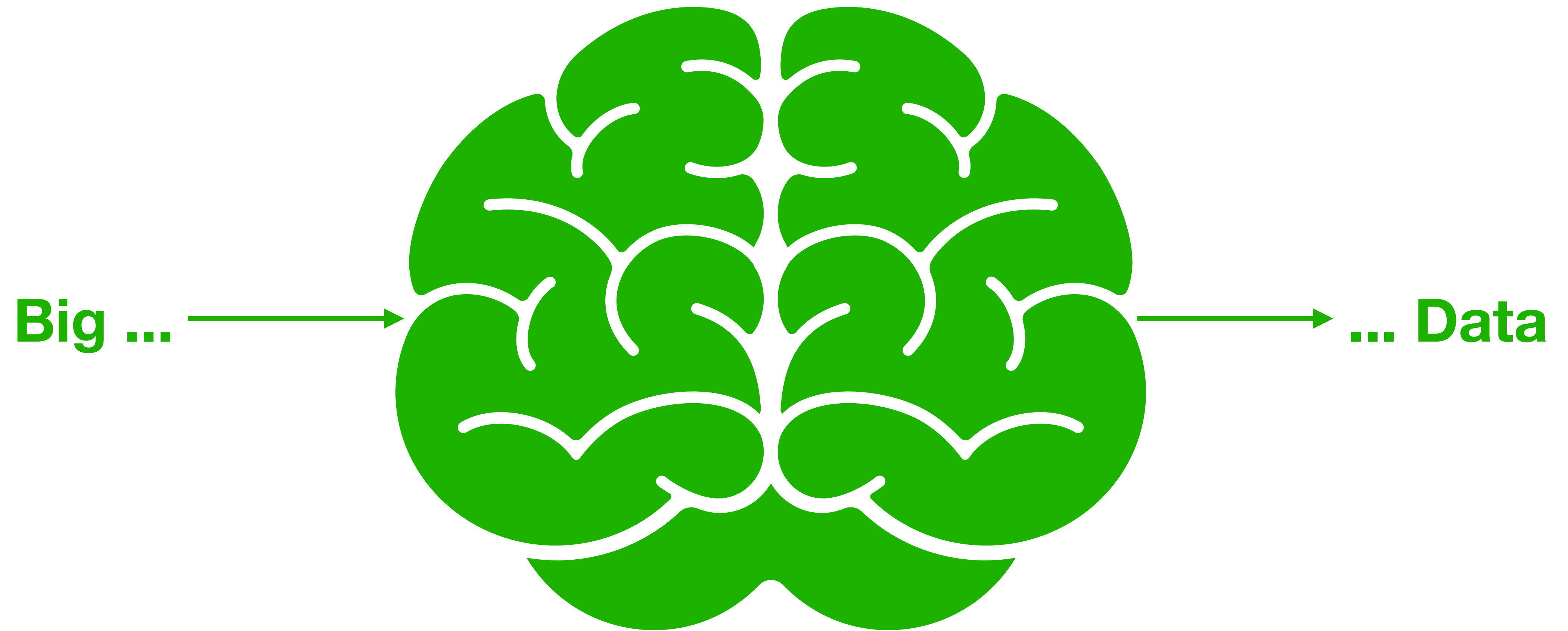
Pre-Training



Pre-Training



Pre-Training



Pre-Trained Model





ChatGPT 3.5 ▾



How can I help you today?

Write a message

that goes with a kitten gif for a friend on a rough day

Suggest fun activities

to help me make friends in a new city

Plan a mental health day

to help me relax

Help me pick

an outfit that will look good on camera

Message ChatGPT



ChatGPT can make mistakes. Check important info.

?

Recent Work on LLMs @ Cornell

- **NEAT**: using text about data to guide data profiling [**CIDR'22 VLDB'22;24**]
- **DB-BERT**: mining tuning hints from text documents [**SIGMOD'22 VLDBJ'23**]
- **λ -Tune**: using language models as database administrators [**SIGMOD'24**]
- **Scrutinizer**: verifying text claims from relational data [**VLDB'20 BDA'20**]
- **NaturalMiner**: automating iterative data analysis [**VLDB'22 SIGMOD'23**]
- **CodexDB**: generating customizable code for SQL queries [**VLDB'22;23**]
- **Evaporate**: extracting structured information from multi-modal data [**VLDB'24**]
- **ThalamusDB**: approximate processing for multi-modal data [**SIGMOD'23;24**]

Tutorial Outline

1. **Intro**
2. **OpenAI's** Python API
3. Building a **NLQI** with GPT-4
4. Analyzing **images** with GPT-4
5. Other **providers** and frameworks
6. **Transformer** Architecture
7. **Transfer** Learning
8. **Conclusion** and Outlook

Tutorial Outline

1. **Intro**

2. **OpenAI's** Python API

3. Building a **NLQI** with GPT-4

4. Analyzing **images** with GPT-4

5. Other **providers** and frameworks

6. **Transformer** Architecture

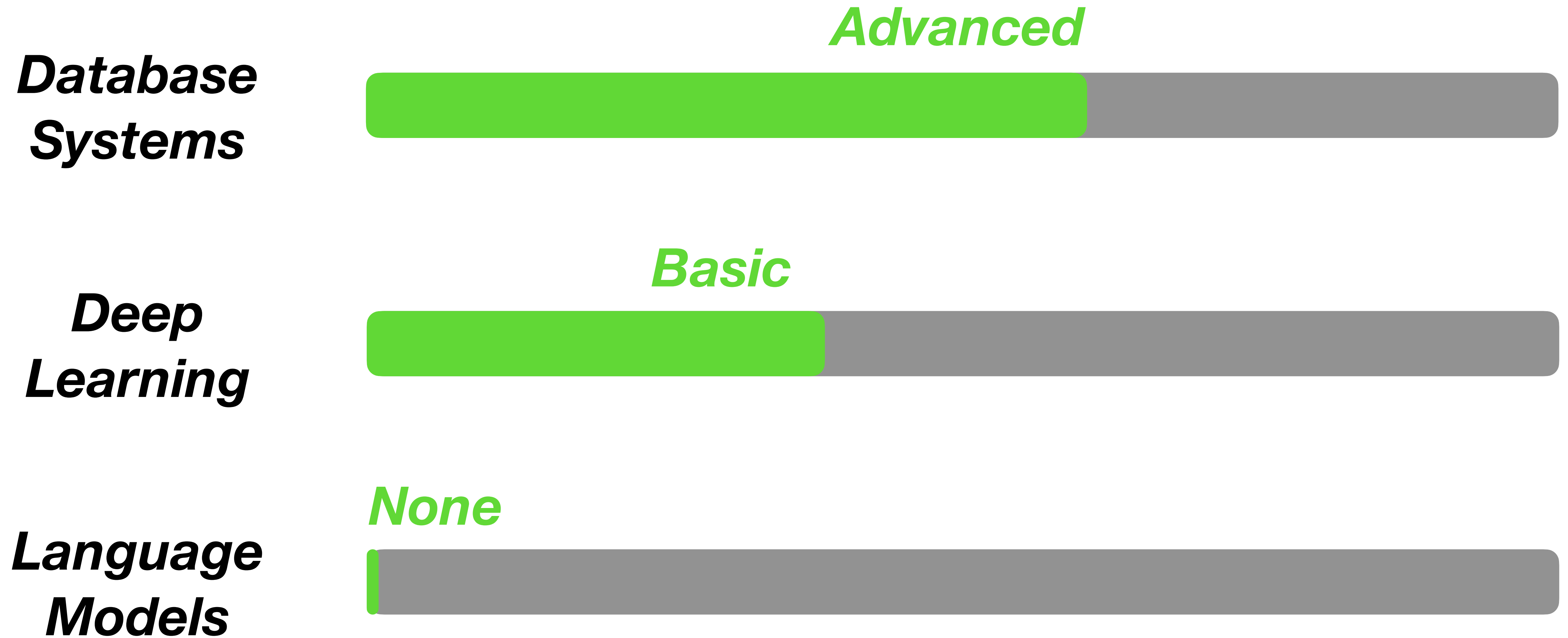
7. **Transfer** Learning

8. **Conclusion** and Outlook

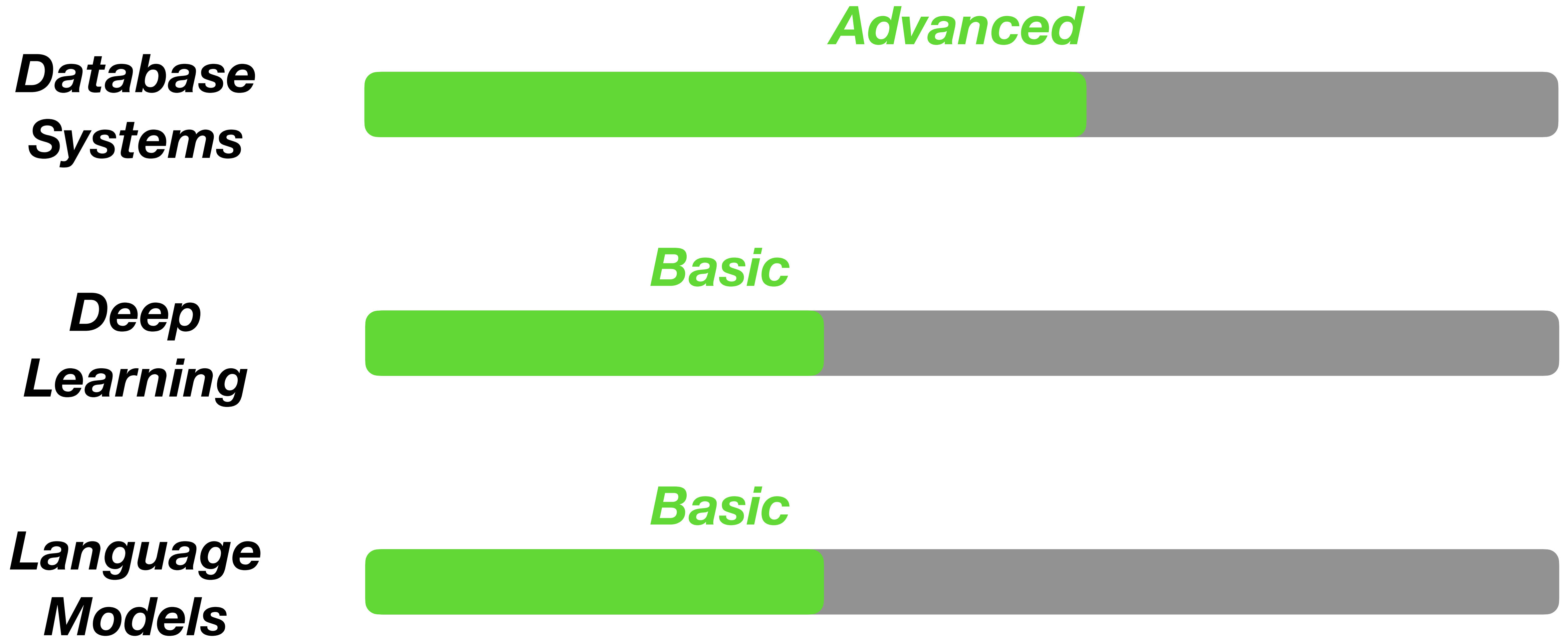
Practice

Principles

Target Audience



Tutorial Goal





OpenAI's Python API

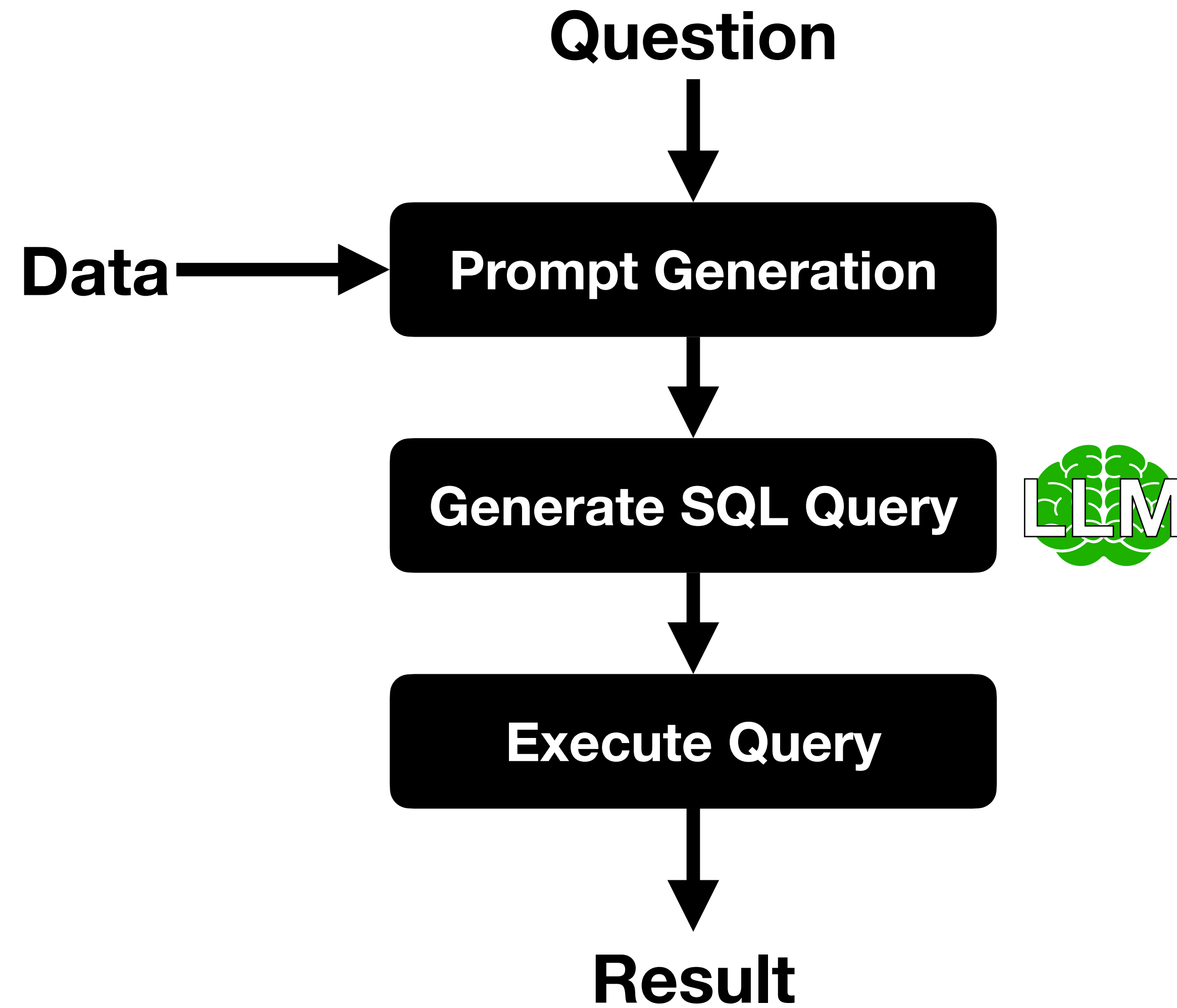
Feature Overview

- Submit **prompts** to OpenAI models
- Retrieve **completions** and meta-data
- Upload files with **training** data
- Create **fine-tuned** model versions
- ...

(Demo)

Building a NLQI with GPT-4

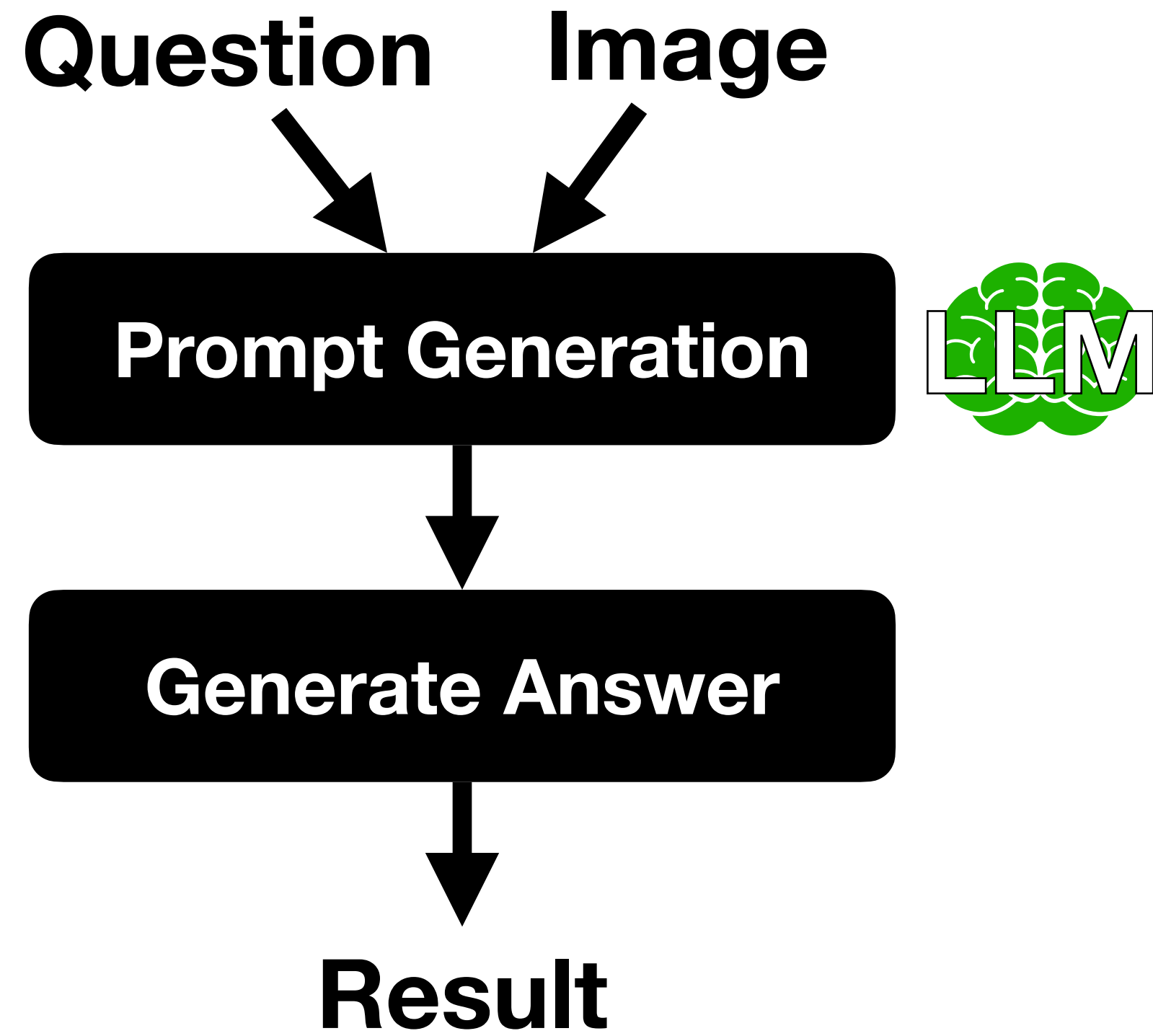
Overview



(Demo)

Analyzing Images with GPT-4

Overview



(Demo)

Other Providers and Frameworks

AI21

- <https://www.ai21.com/>
- **JAMBA** - Joint Attention and Mamba
 - Up to 256K context window size
- Multiple **specialized** models
 - Question answering
 - Summarization
 - ...

Anthropic

- <https://www.anthropic.com/>
- **Constitutional** AI
- Claude 3 Opus
 - Up to **200K/1 Million** tokens

Cohere

- <https://cohere.com/>
- Focus on **integration** of data sources
 - Retrieval augmented generation
 - Lots of connectors out-of-the-box
 - Can define custom connectors
- Partnerships with Oracle, MongoDB, ...

Google

- <https://gemini.google.com/app>
- **Gemini model** series
 - Supports text and images
 - Integration with Google tools

Hugging Face

- <https://huggingface.co/>
- Very **rich collection** of specialized models
 - Various tasks, model types, data modalities
- Can run models **locally** or in the Cloud
- **Resources** for training and fine-tuning

Model Comparisons

Center for Research on Foundation Models

HELM

ModelsScenariosResultsRaw runs

v0.2.0 (last updated 2022-12-29)

Core scenarios

The scenarios where we evaluate all the models.

[Accuracy | Calibration | Robustness | Fairness | Efficiency | General information | Bias | Toxicity | Summarization metrics | JSON]

Accuracy

Model/adaptor	Mean win rate ↑ [sort]	MMLU - EM ↑ [sort]	BoolQ - EM ↑ [sort]	NarrativeQA - F1 ↑ [sort]	NaturalQuestions (closed-book) - F1 ↑ [sort]	NaturalQuestions (open-book) - F1 ↑ [sort]	QuAC - F1 ↑ [sort]	HellaSwag - EM ↑ [sort]	OpenbookQA - EM ↑ [sort]	TruthfulQA - EM ↑ [sort]	MS MARCO (regular) - RR@10 ↑ [sort]	MS MARCO (TREC) - NDCG@10 ↑ [sort]
text-davinci-002	0.952	0.568	0.877	0.727	0.383	0.713	0.445	0.815	0.594	0.61	0.421	0.664
text-davinci-003	0.909	0.569	0.881	0.727	0.406	0.77	0.525	0.822	0.646	0.593	0.368	0.644
TNLG v2 (530B)	0.879	0.469	0.809	0.722	0.384	0.642	0.39	0.799	0.562	0.251	0.377	0.643
Anthropic-LM v4-s3 (52B)	0.864	0.481	0.815	0.728	0.288	0.686	0.431	0.807	0.558	0.368	-	-
J1-Grande v2 beta (17B)	0.806	0.445	0.812	0.725	0.337	0.625	0.392	0.764	0.56	0.306	0.285	0.46
Cohere xlarge v20221108	0.764	0.382	0.762	0.672	0.361	0.628	0.374	0.81	0.588	0.169	0.315	0.55

Slides by Immanuel Trummer, Cornell University

LangChain

LlamaIndex

The Transformer Architecture

Attention Is All You Need

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

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly

Attention is All You Need

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

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly

Attention Mechanism vs. Key-Value Stores

- Shared **vocabulary**:
 - Keys, values, queries
- Similar **semantics**:
 - Find keys matching query
 - Retrieve associated values

Attention: Simplifying Intuition

I

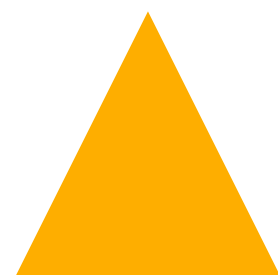
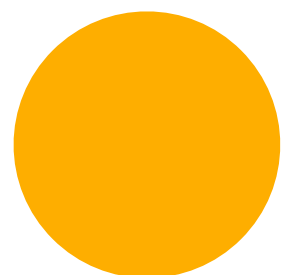
Love

Database

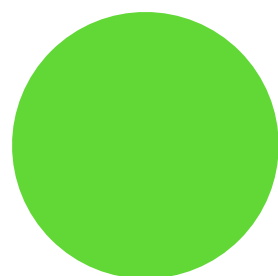
Research

Attention: Simplifying Intuition

Key



Value



I

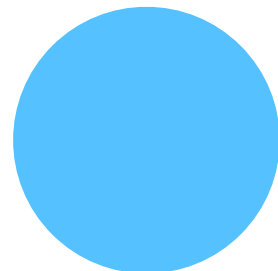
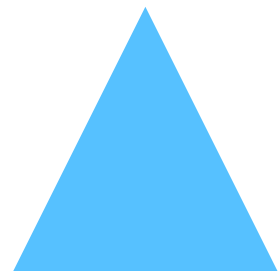
Love

Database

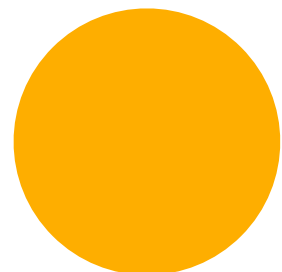
Research

Attention: Simplifying Intuition

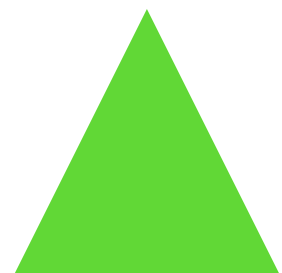
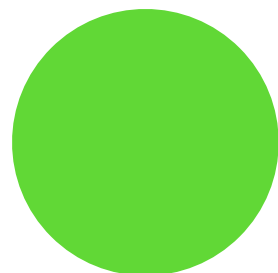
Query



Key



Value



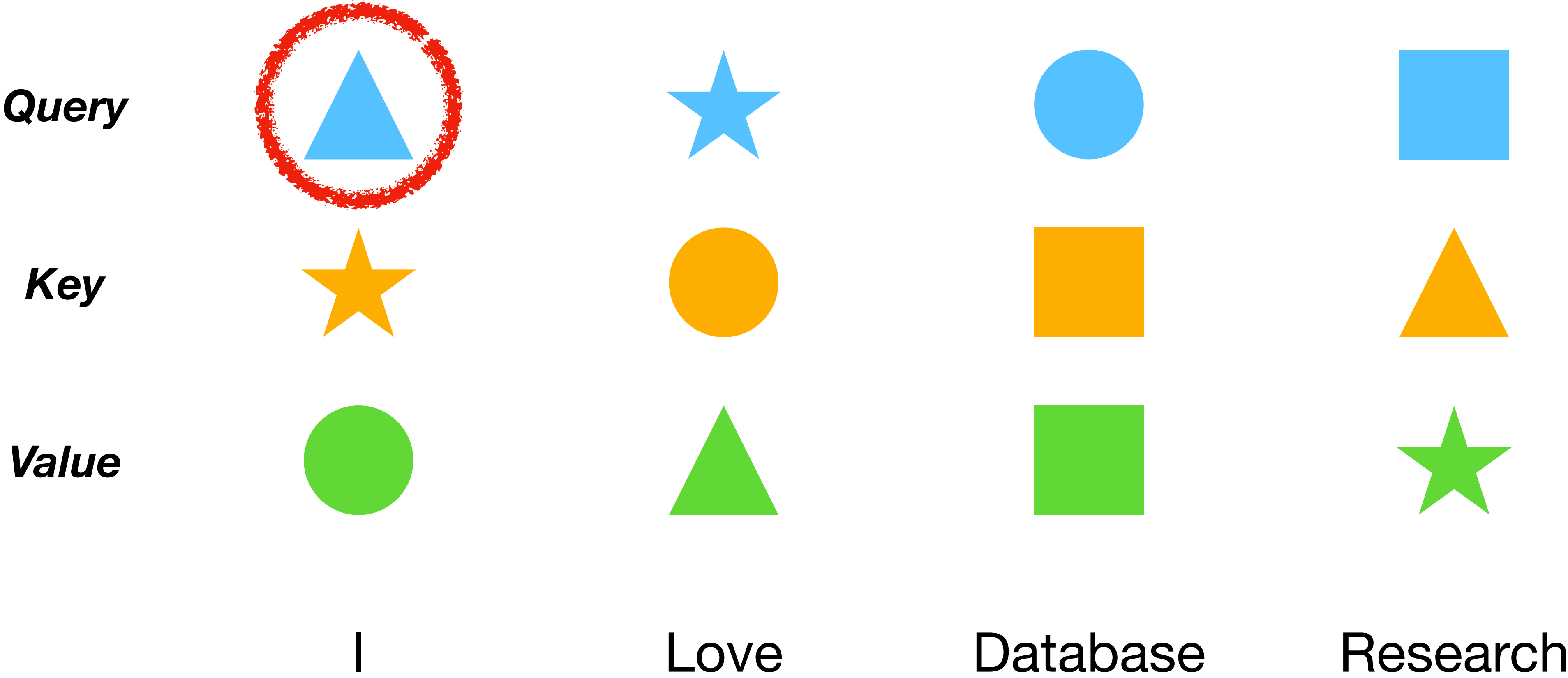
I

Love

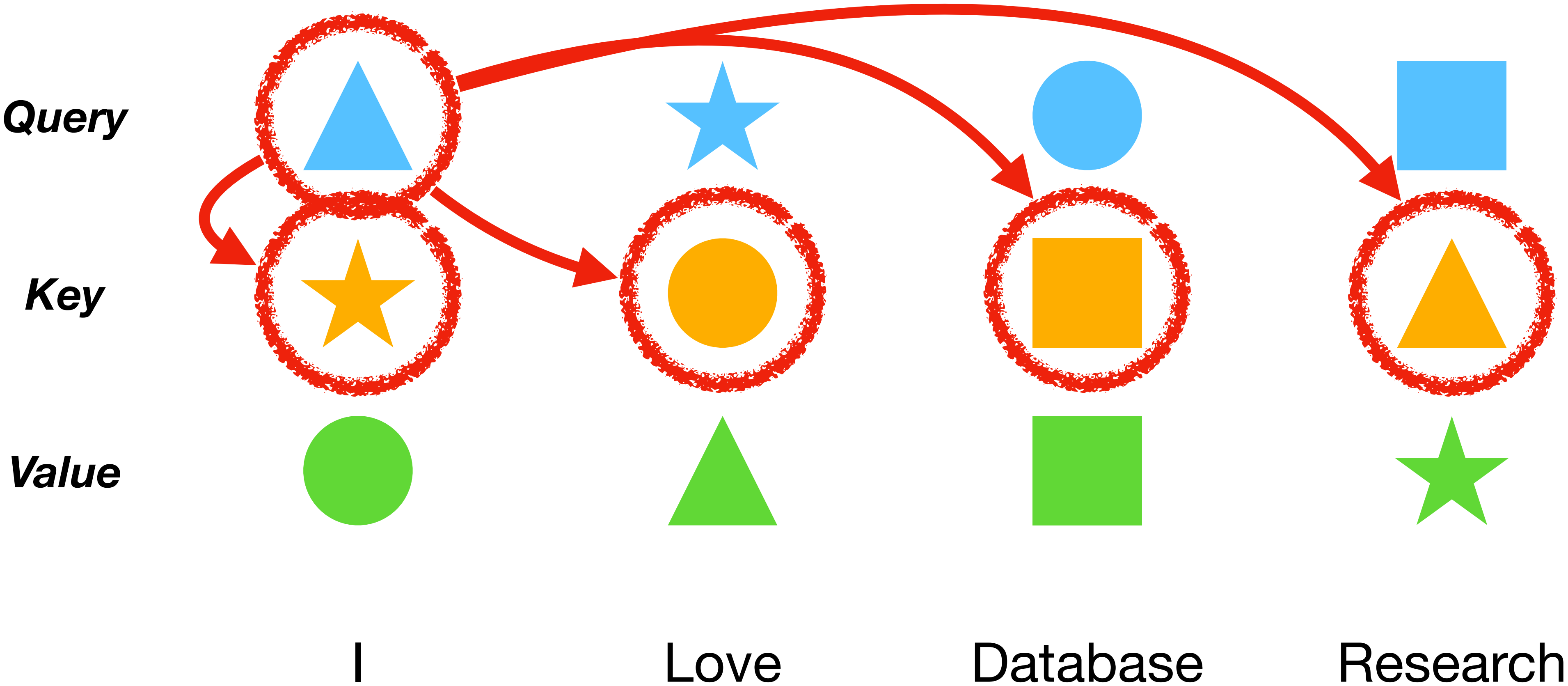
Database

Research

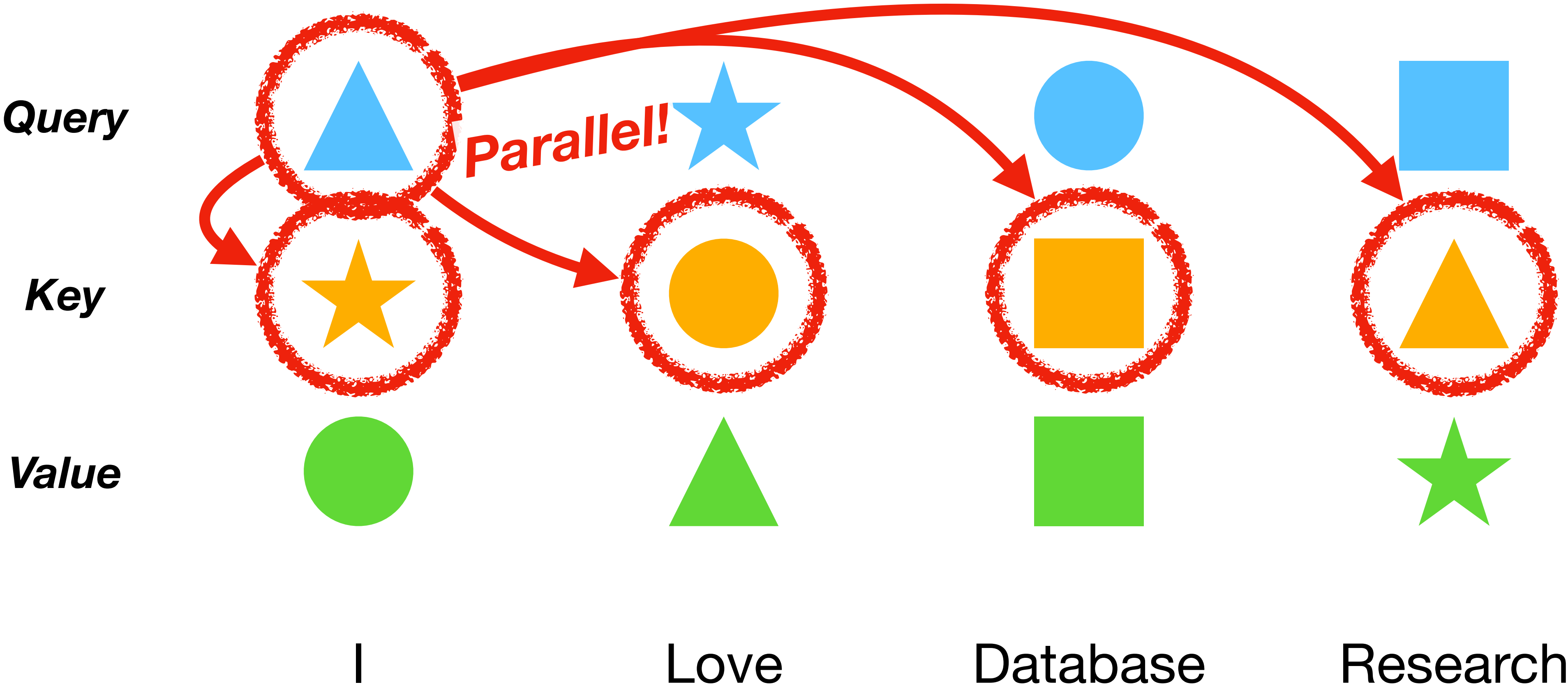
Attention: Simplifying Intuition



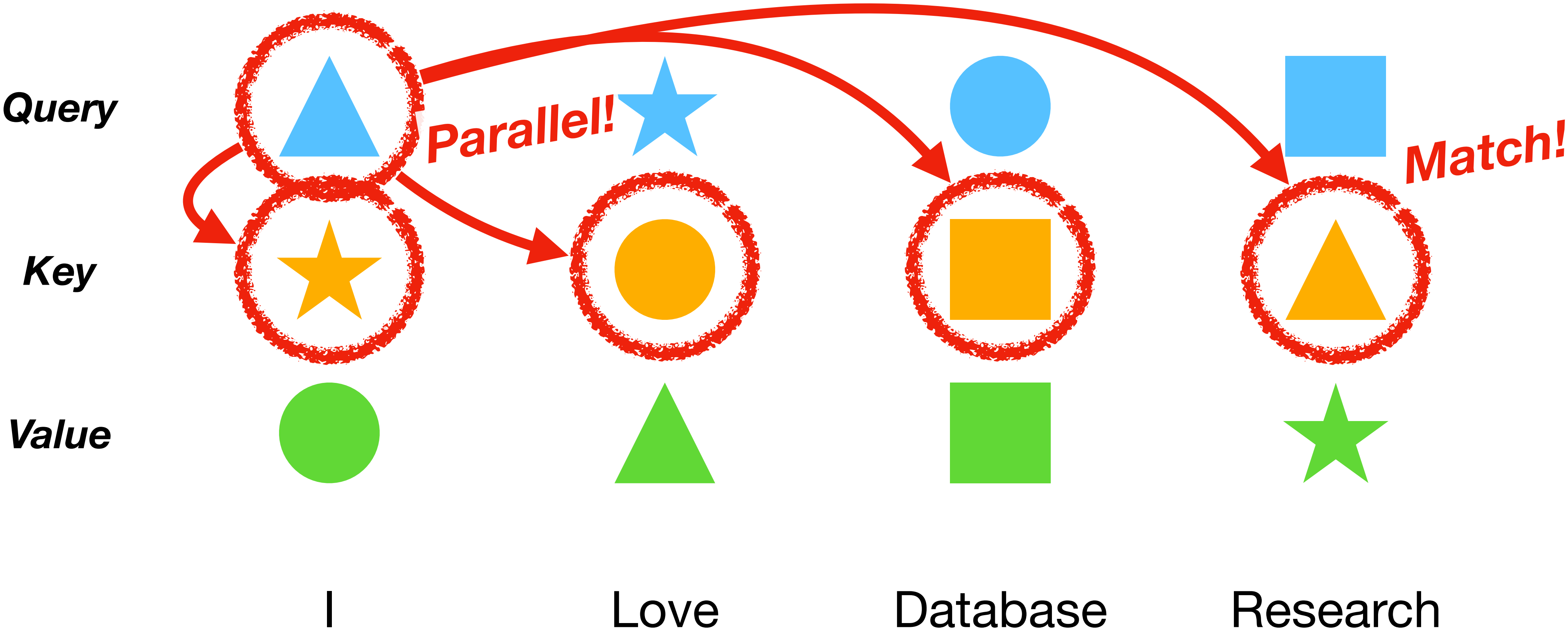
Attention: Simplifying Intuition



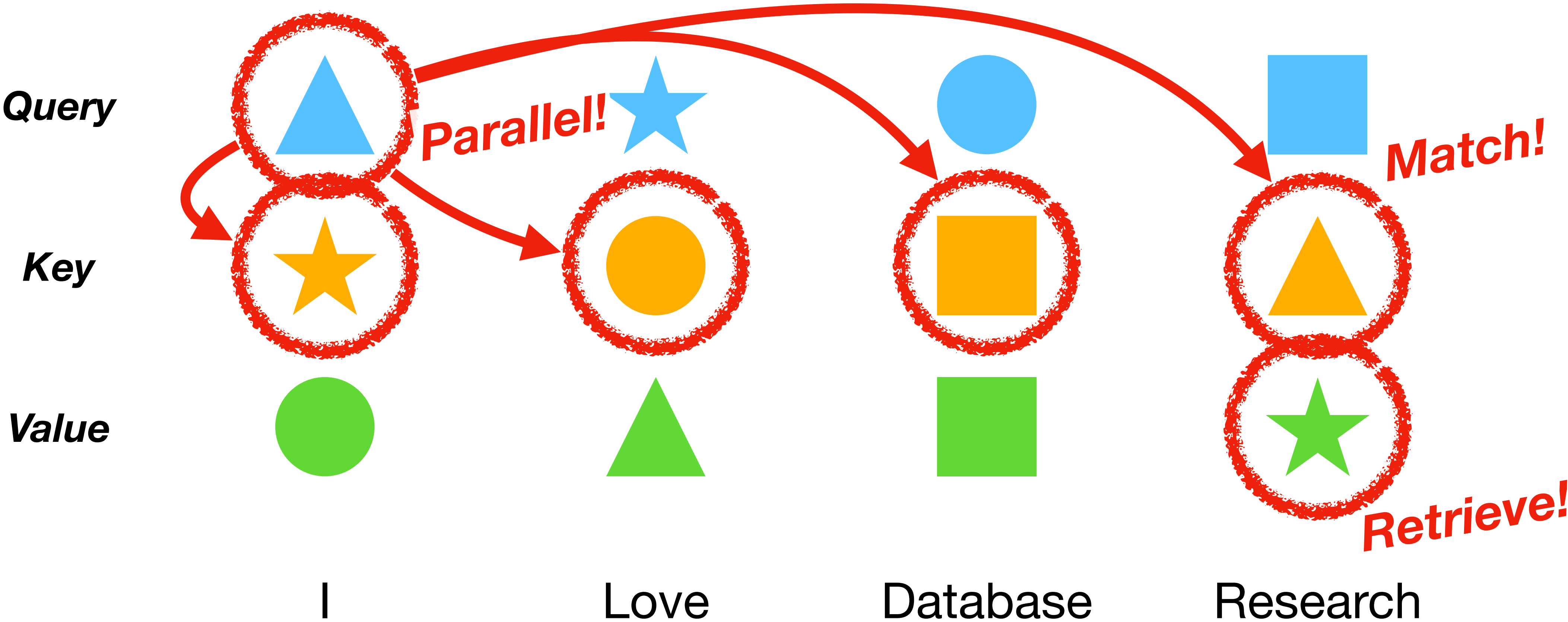
Attention: Simplifying Intuition



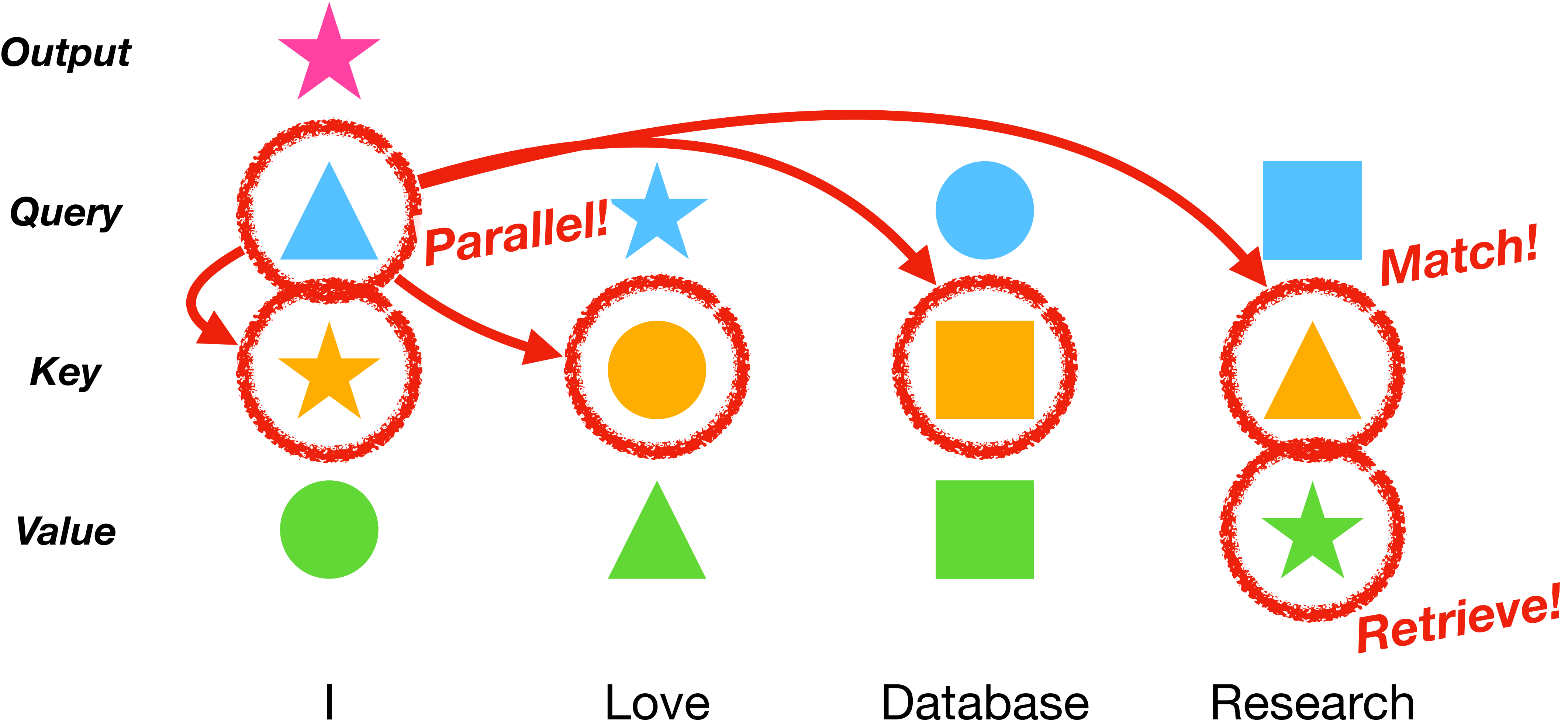
Attention: Simplifying Intuition



Attention: Simplifying Intuition

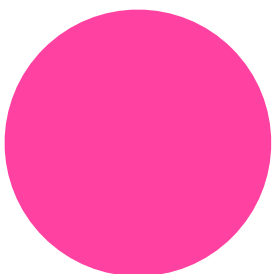


Attention: Simplifying Intuition

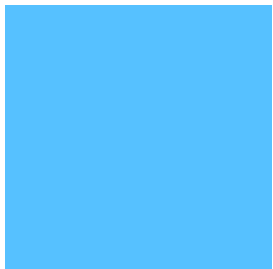
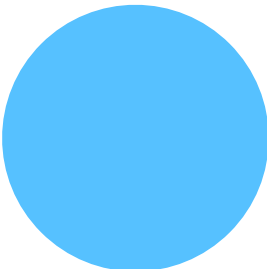
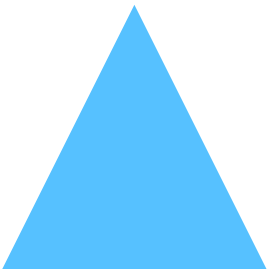


Attention: Simplifying Intuition

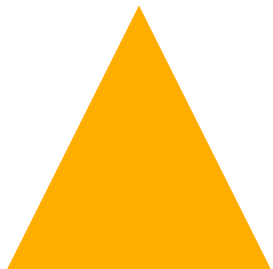
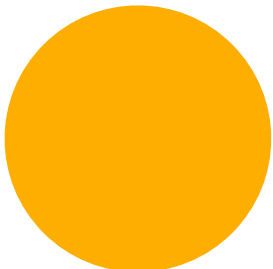
Output



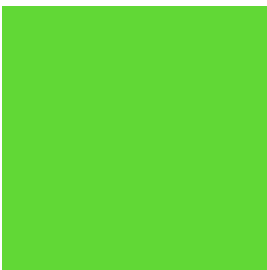
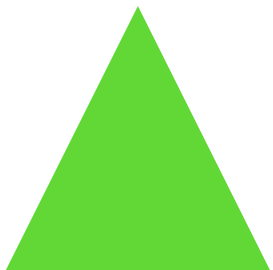
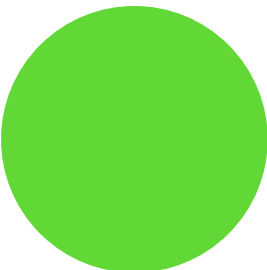
Query



Key



Value



I

Love

Database


Research

Attention vs. Simplifying Intuition

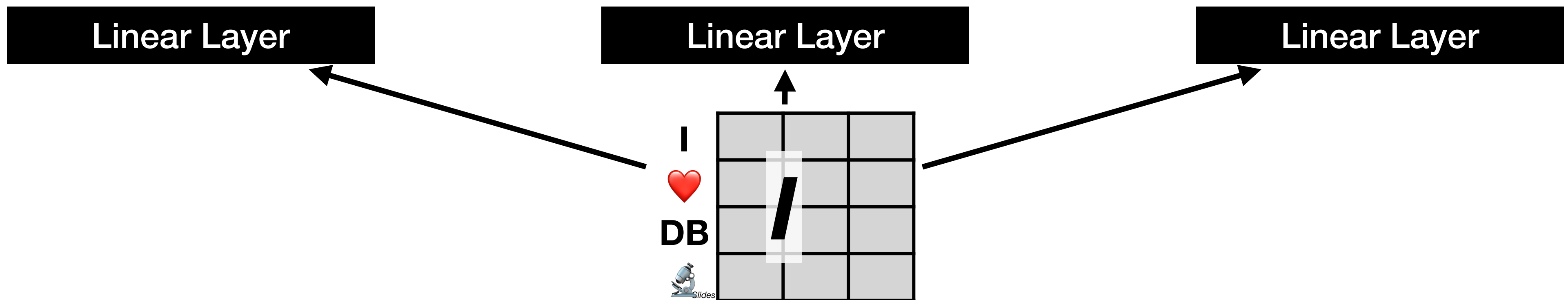
- Real-valued **vectors** instead of discrete symbols
- Continuous **similarity** between queries and keys
- Output value is **sum** of values, weighted by similarity
- Several **normalization** steps

Attention

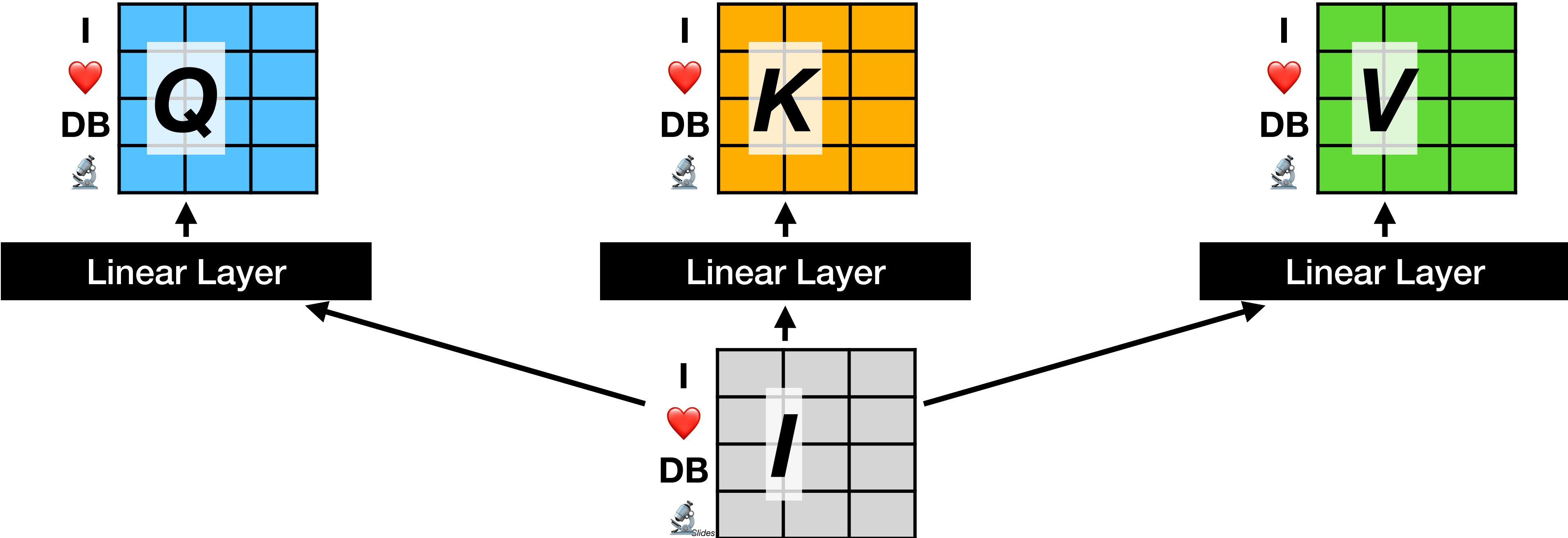
I
❤️
DB


Slides

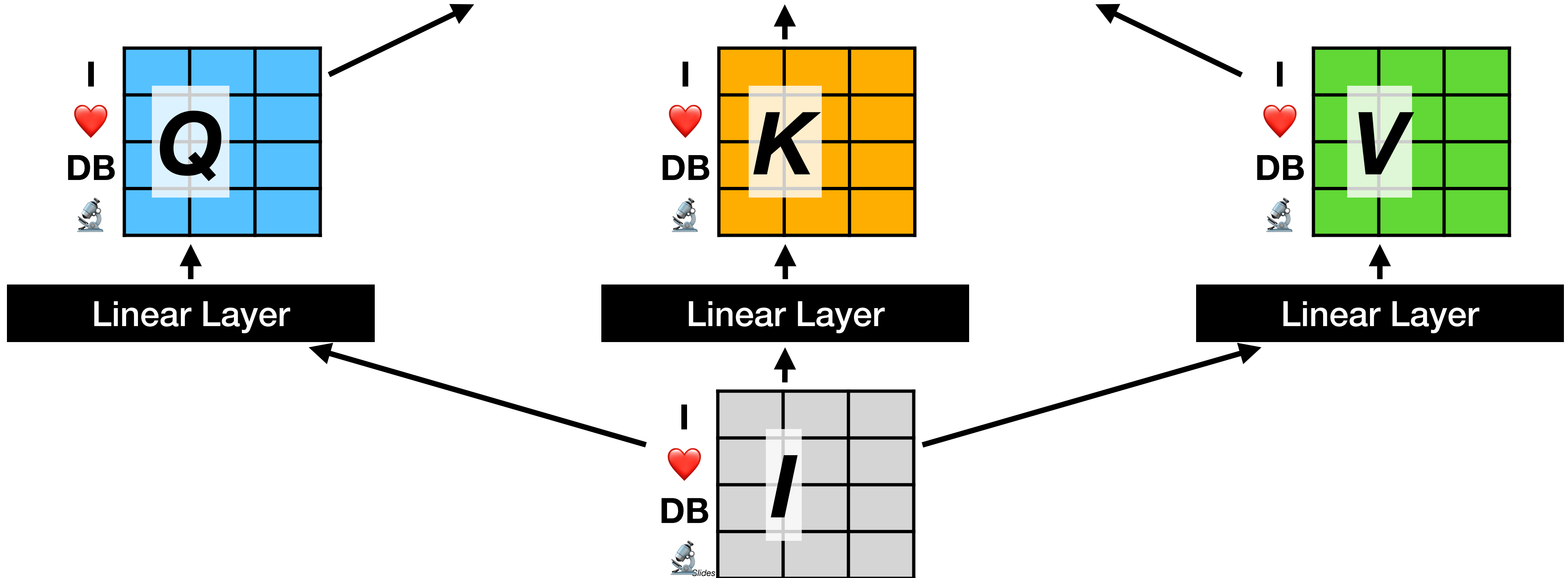
Attention



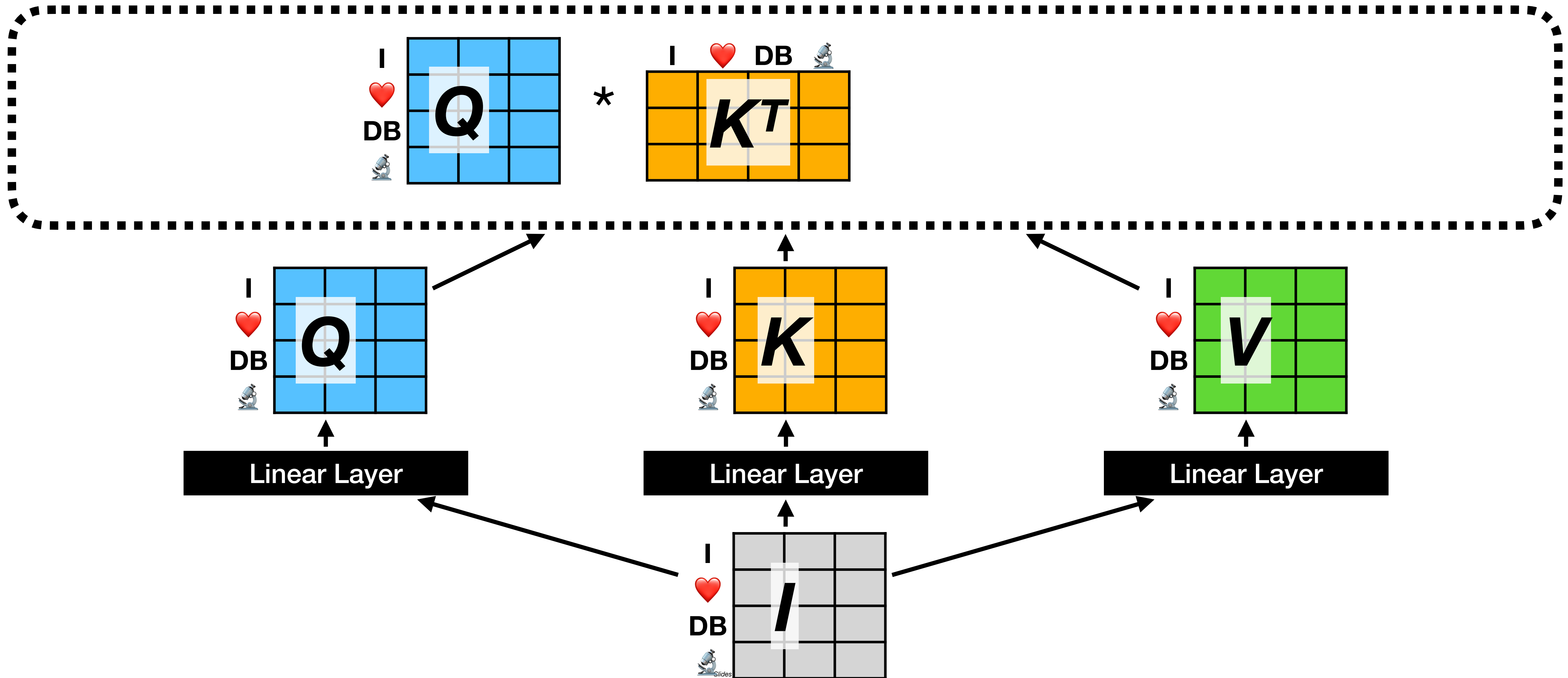
Attention



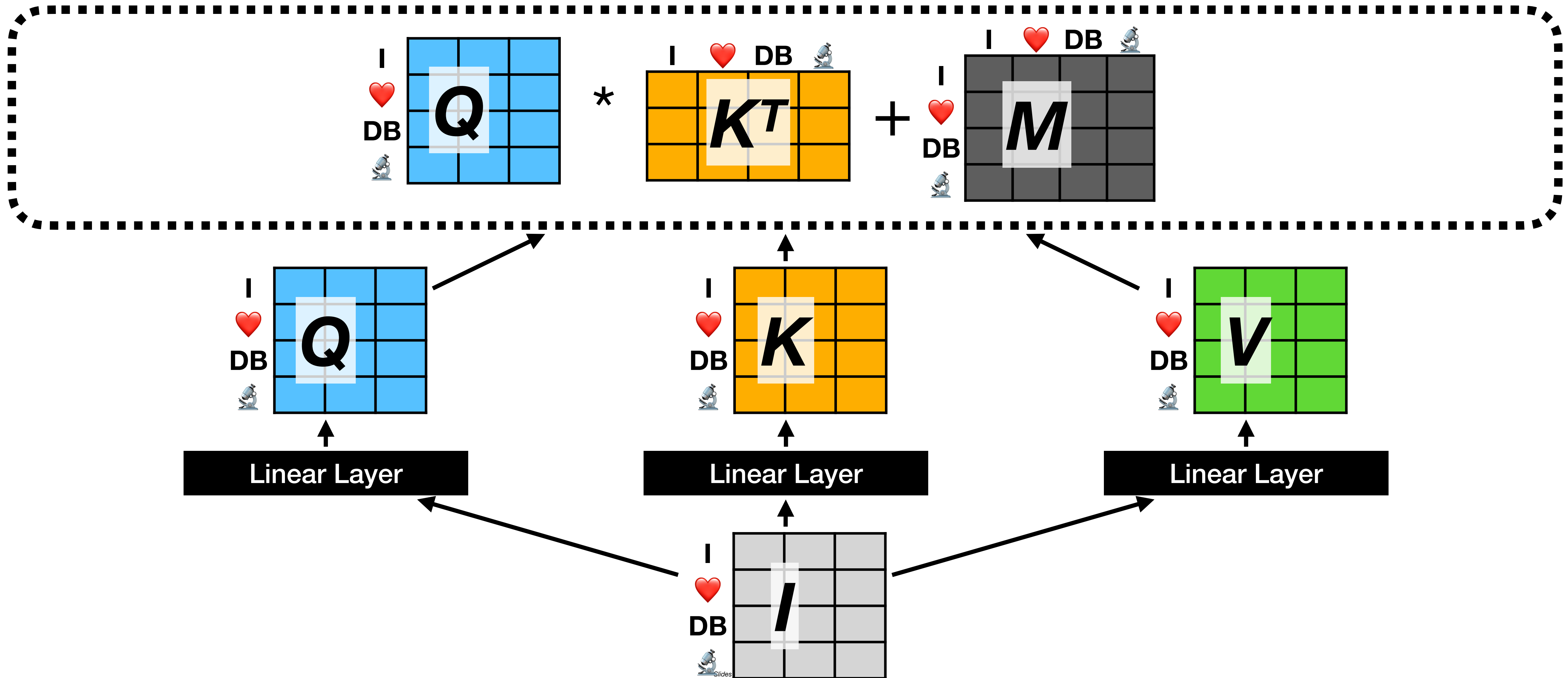
Attention



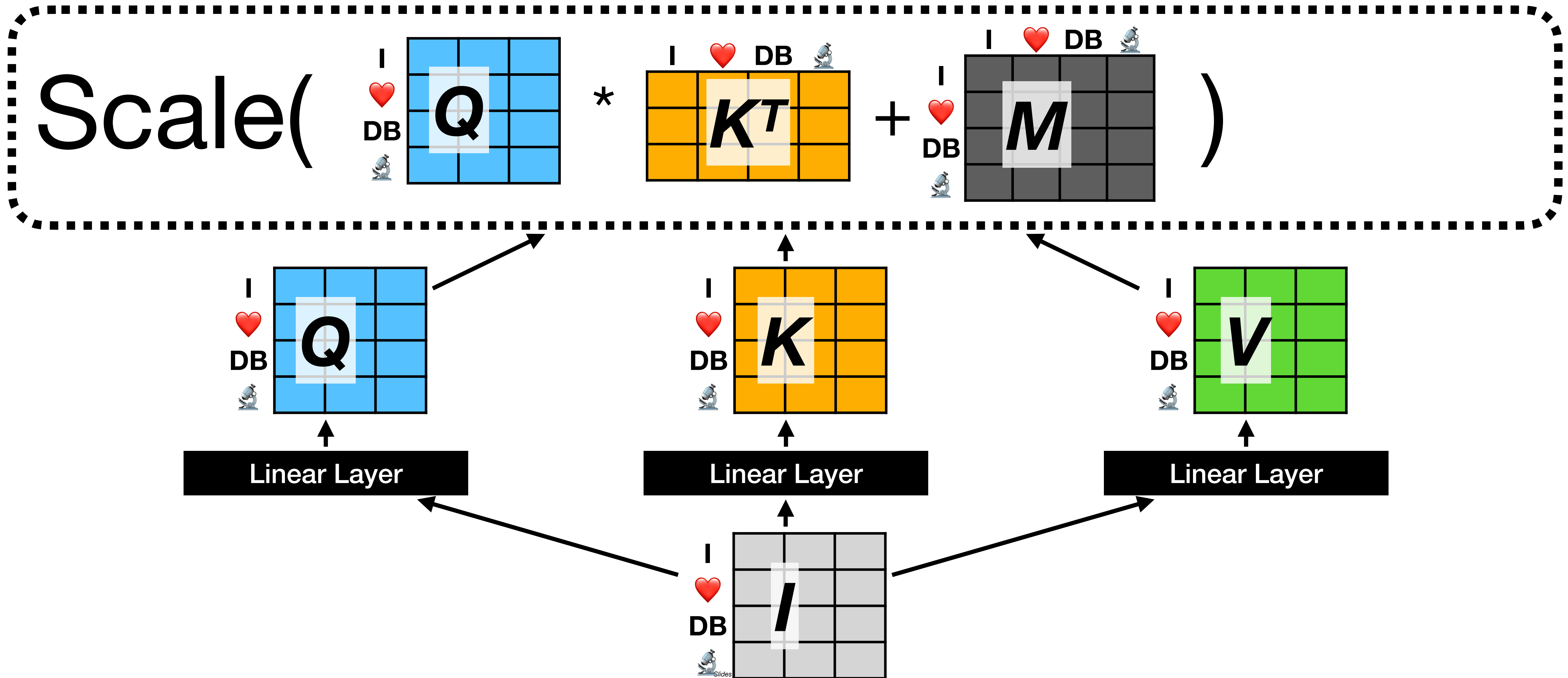
Attention



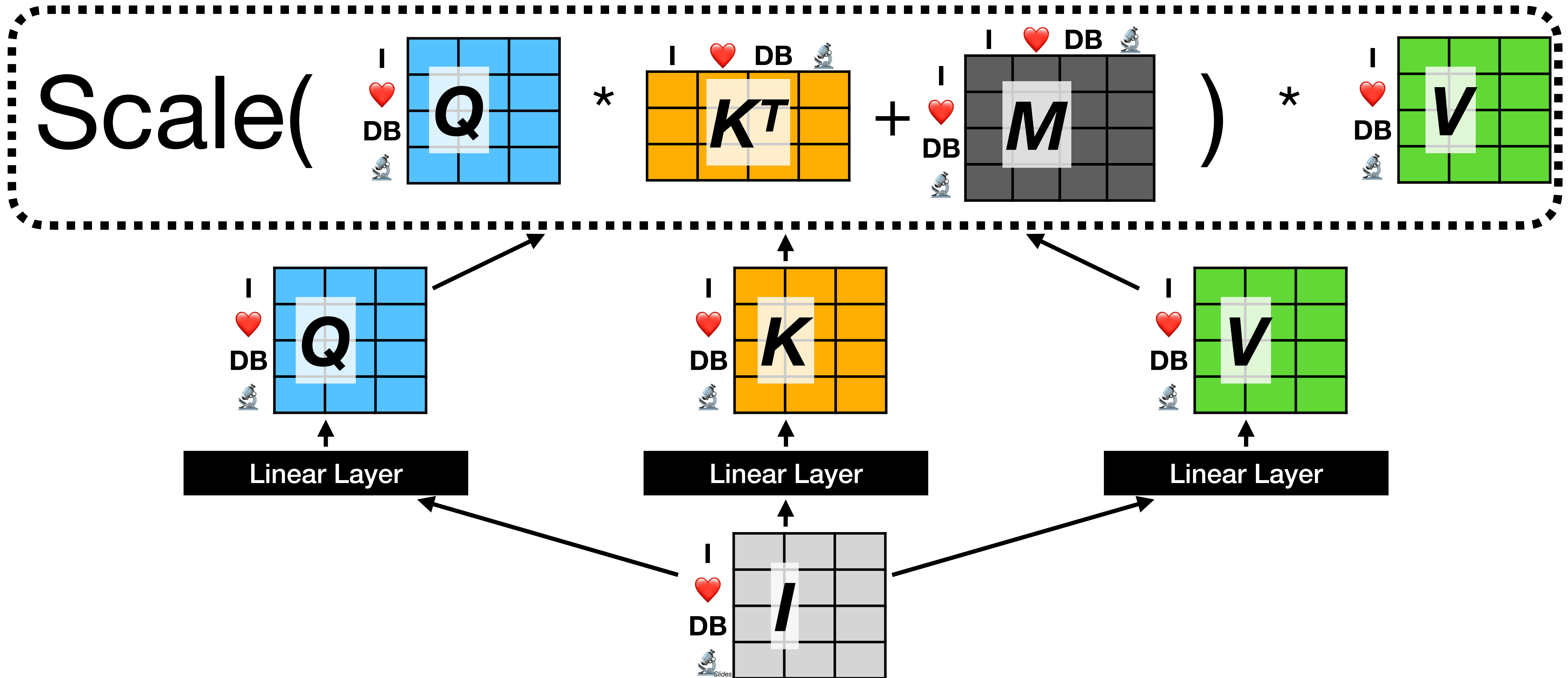
Attention



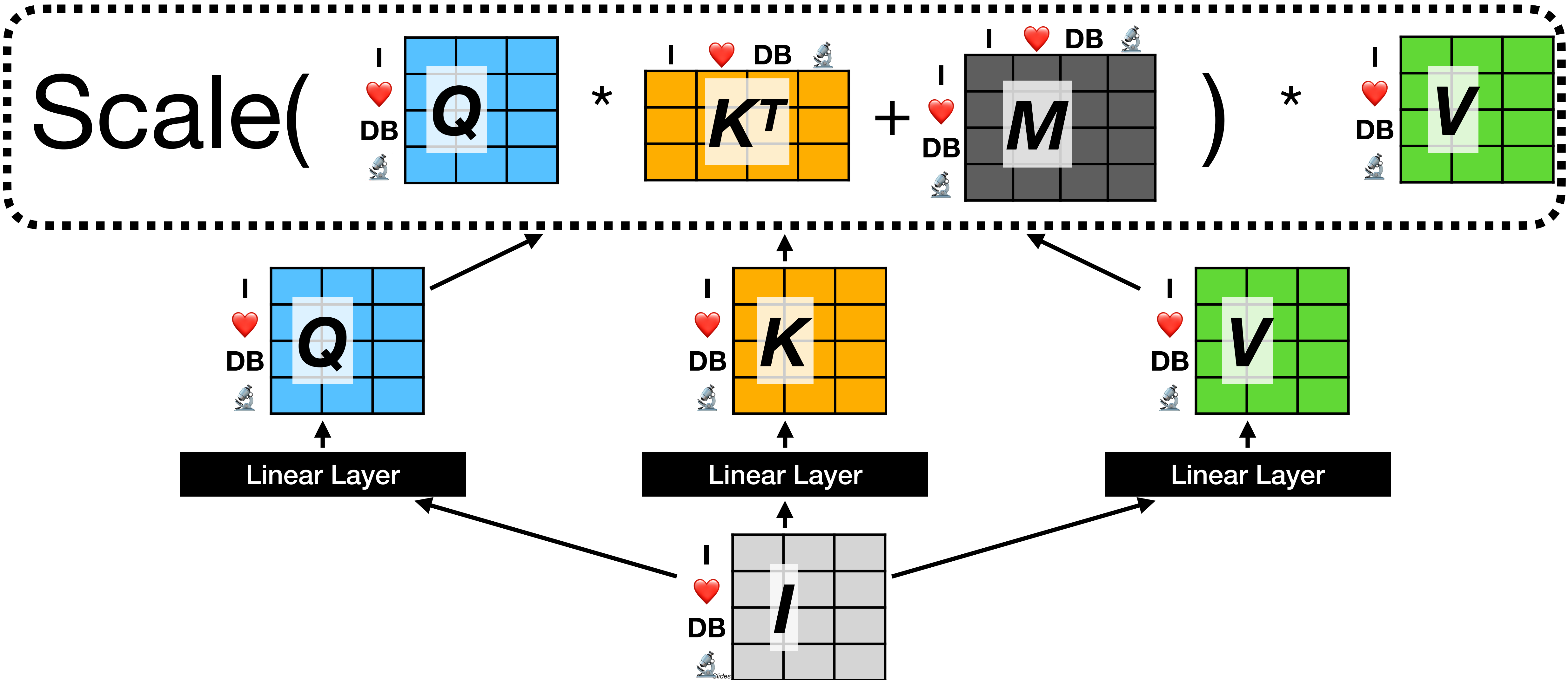
Attention



Attention

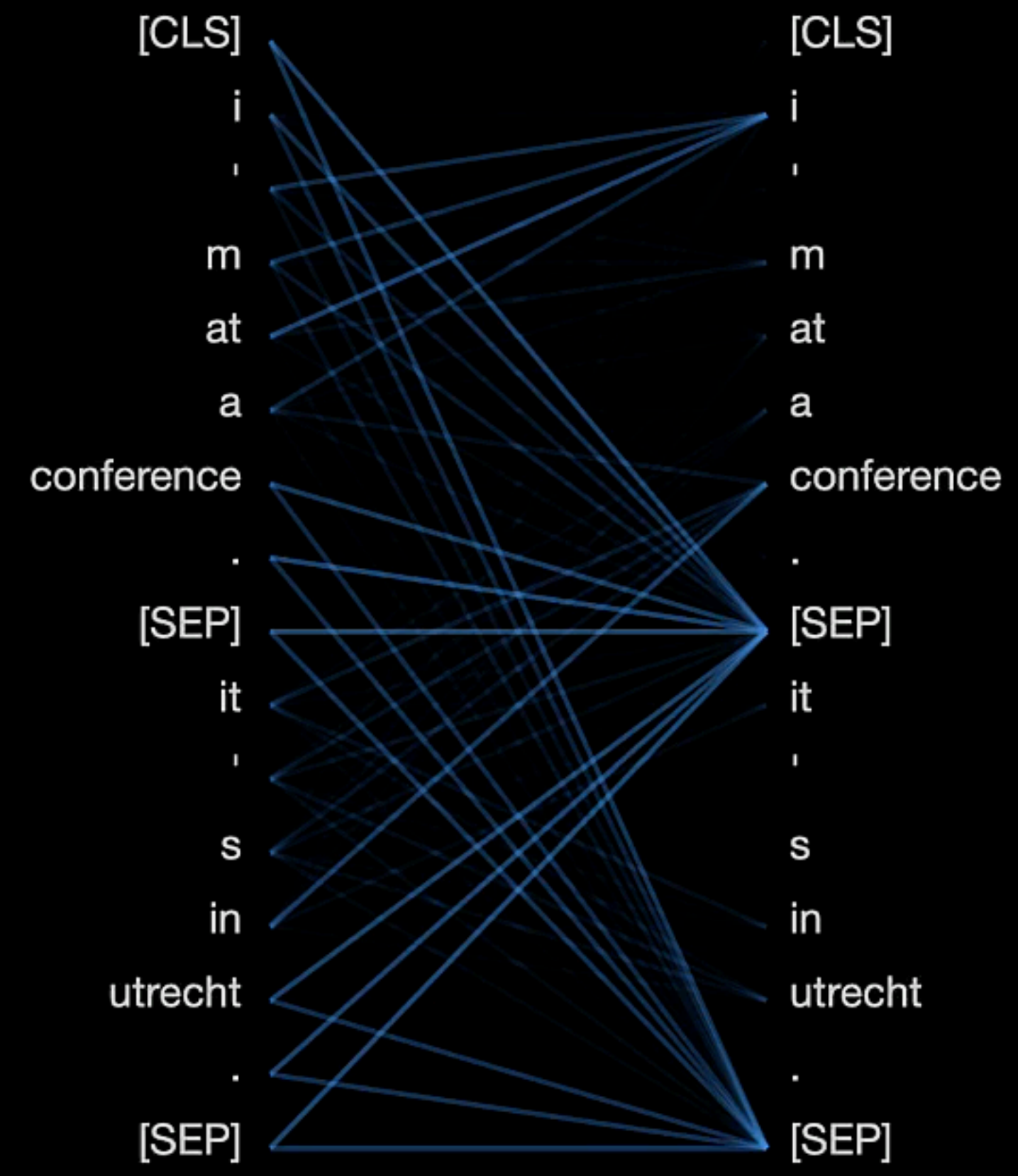


Attention



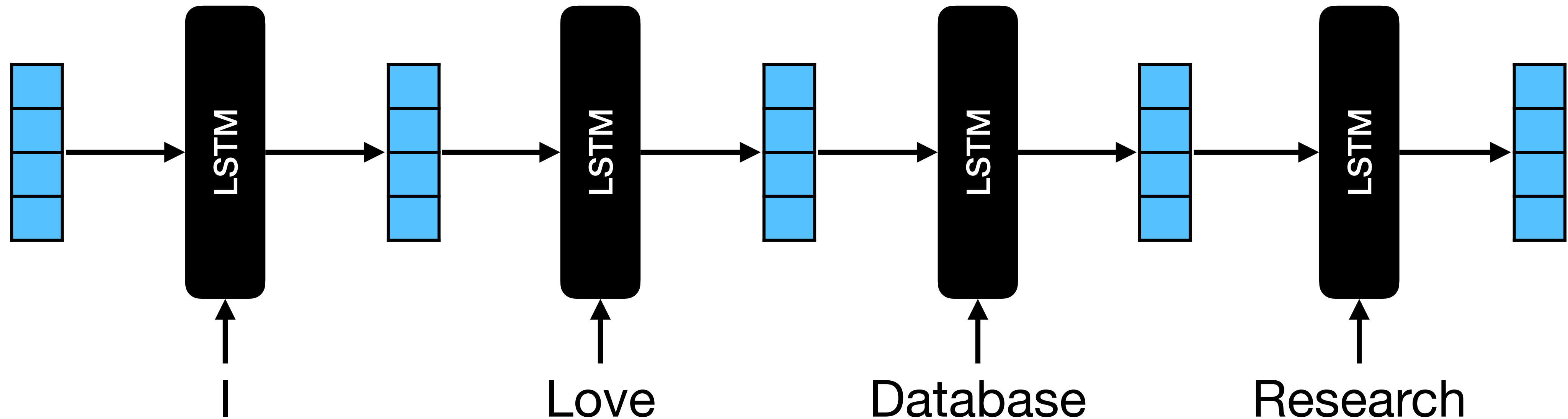
100%|██████████| 433/433 [00:00<00:00, 1097361.71B/s]
100%|██████████| 440473133/440473133 [00:08<00:00, 54642543.60B/s]
100%|██████████| 231508/231508 [00:00<00:00, 2674353.40B/s]

Layer: 8 ▾ Head: 10 ▾ Attention: All ▾

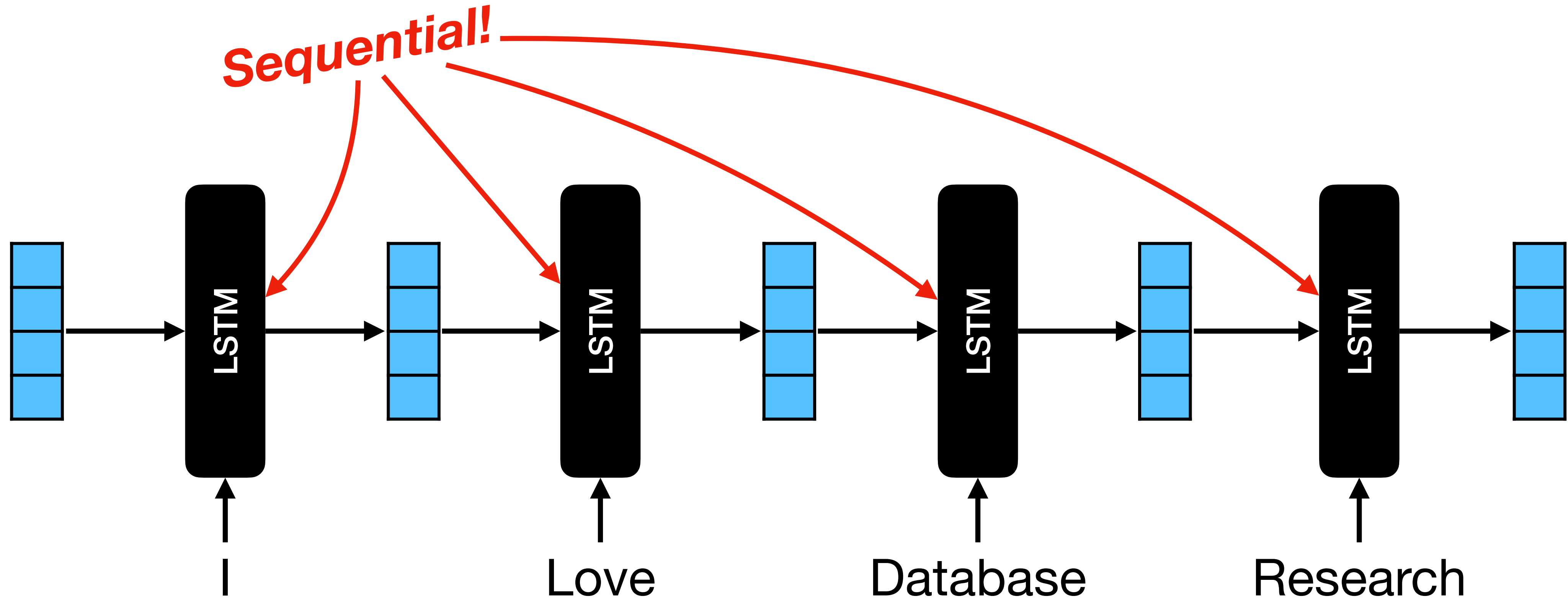


4

Recurrent Neural Network



Recurrent Neural Network



Attention versus Recurrence

d : Vector dimension
 n : Sequence length

Layer Type	Complexity per Layer
Self-Attention	$O(n^2 * d)$
Recurrent	$O(n * d^2)$

Attention versus Recurrence

d : Vector dimension
 n : Sequence length

Layer Type	Complexity per Layer
Self-Attention	$O(n^2 * d)$
Recurrent	$O(n * d^2)$

Self-attention is ...

Faster if $d > n$

Attention versus Recurrence

d : Vector dimension
 n : Sequence length

Layer Type	Complexity per Layer	Sequential Operations
Self-Attention	$O(n^2 * d)$	$O(1)$
Recurrent	$O(n * d^2)$	$O(n)$

Self-attention is ... Faster if $d > n$

Attention versus Recurrence

d : Vector dimension
 n : Sequence length

Layer Type	Complexity per Layer	Sequential Operations
Self-Attention	$O(n^2 * d)$	$O(1)$
Recurrent	$O(n * d^2)$	$O(n)$

Self-attention is ...

Faster if $d > n$

More parallelizable

Attention versus Recurrence

d : Vector dimension
 n : Sequence length

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$

Self-attention is ... Faster if $d > n$ More parallelizable

Attention versus Recurrence

d : Vector dimension
 n : Sequence length

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 * d)$	$O(1)$	$O(1)$
Recurrent	$O(n * d^2)$	$O(n)$	$O(n)$

Self-attention is ...

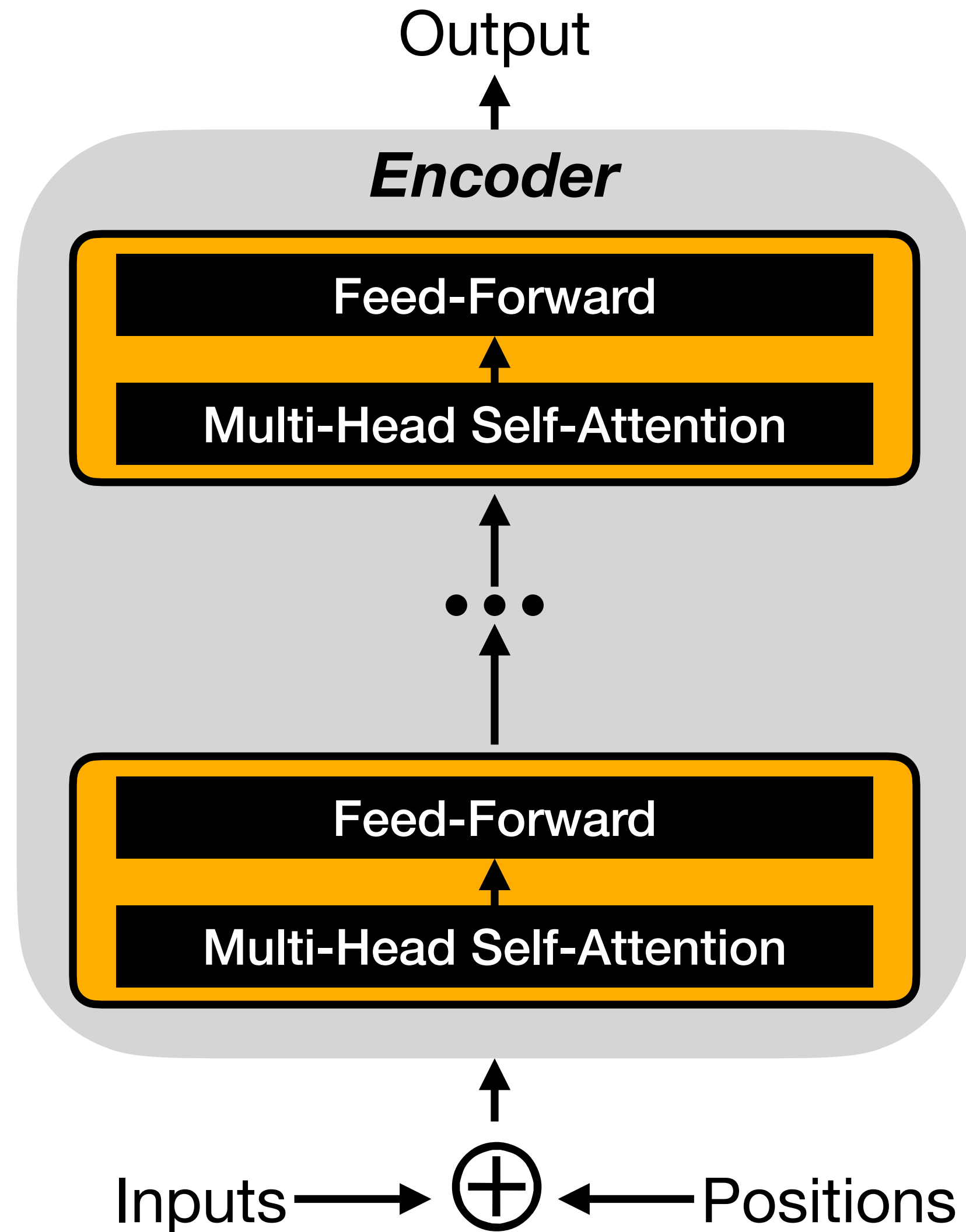
Faster if $d > n$

More parallelizable

Easier to learn

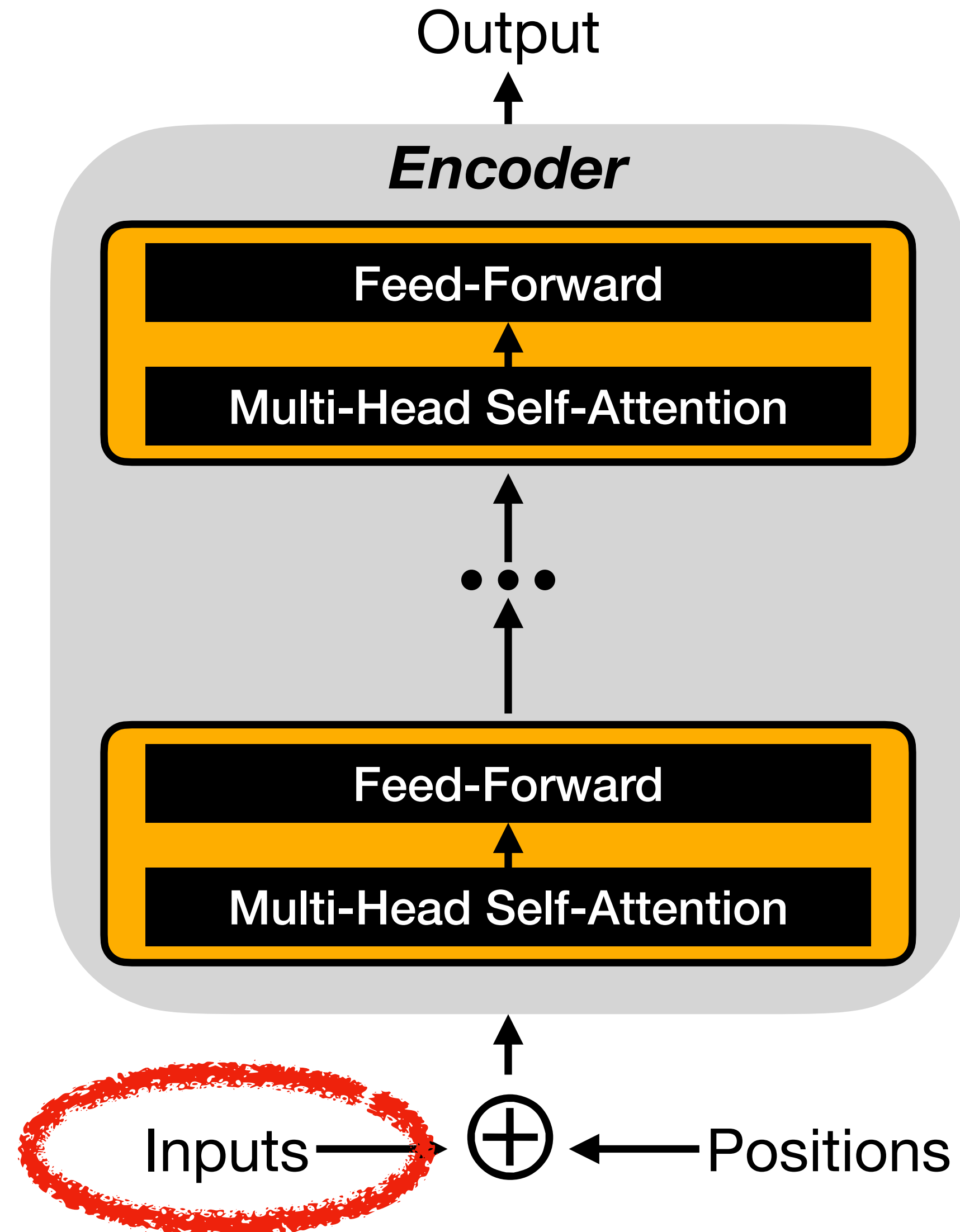
The Transformer

(Details omitted: skip connections, layer normalization, masking)



The Transformer

(Details omitted: skip connections, layer normalization, masking)



Representing Text: Tokenization

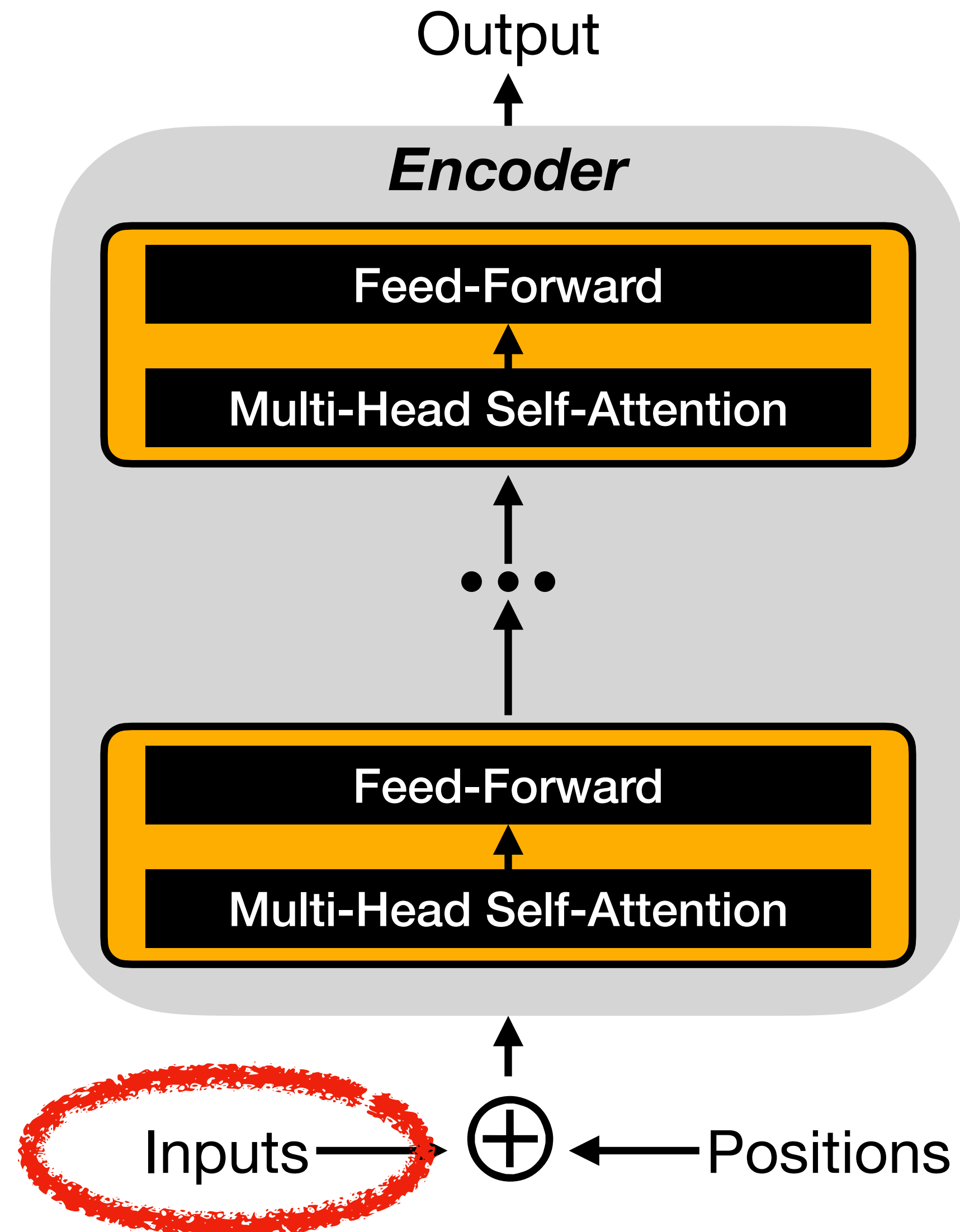
- Simple approach: map each **character** to token ID
 - Inefficient representation leading to bad performance
- Better approach: map each **word** to token ID
 - Leads to very large vocabulary, inefficient
 - May have to resort to <Unknown> token
- Typical approach: map **sub-words** to token IDs
 - Introduce IDs only for frequent sequences
 - Can avoid use of <Unknown> tokens

Representing Text: Embedding

- Can map text to sequence of **token IDs**
- Could represent by **one-hot** encoding
- However: large #dimensions, **inconvenient**
- Better: map to lower-dimensional token **embeddings**

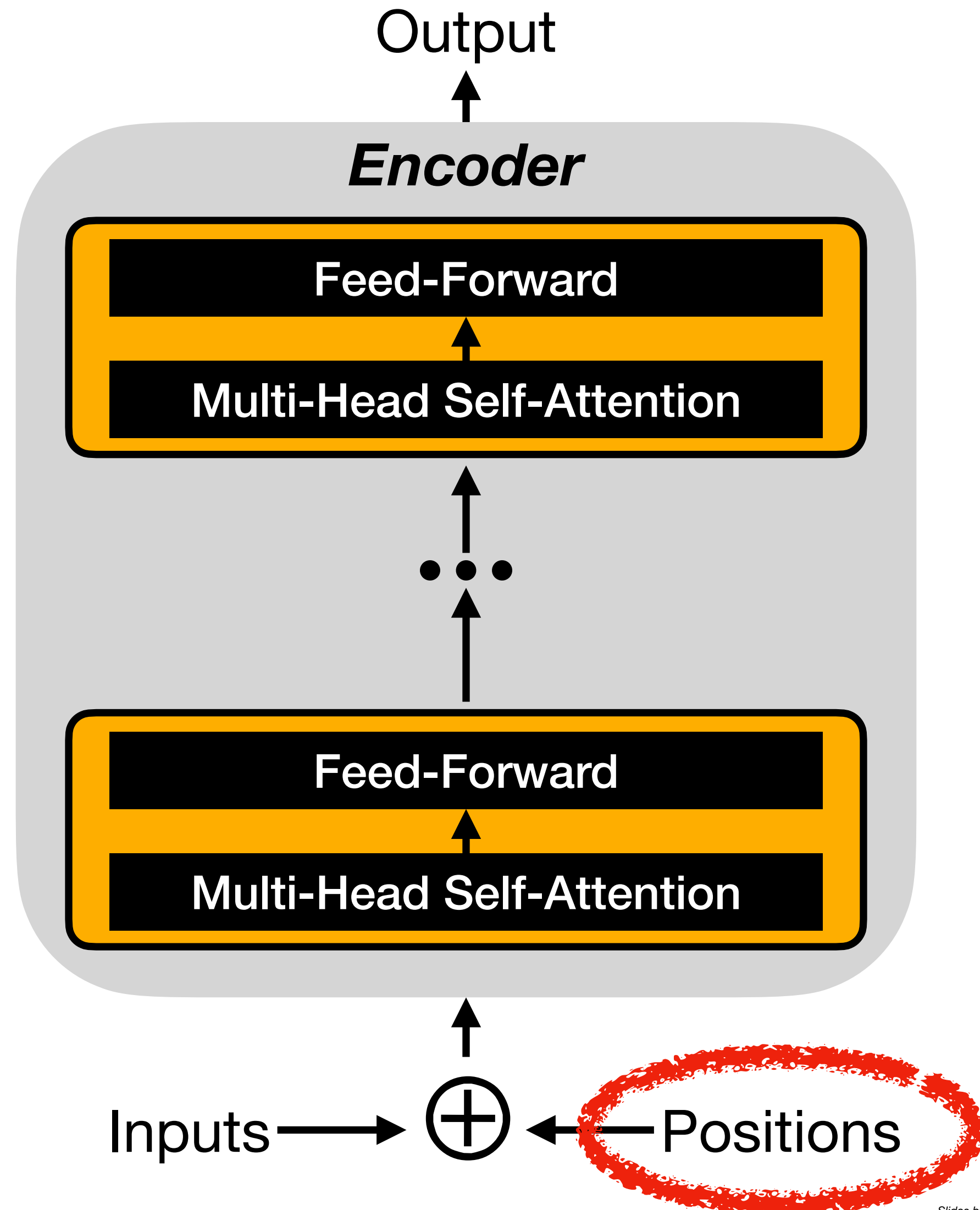
The Transformer

(Details omitted: skip connections, layer normalization, masking)



The Transformer

(Details omitted: skip connections, layer normalization, masking)

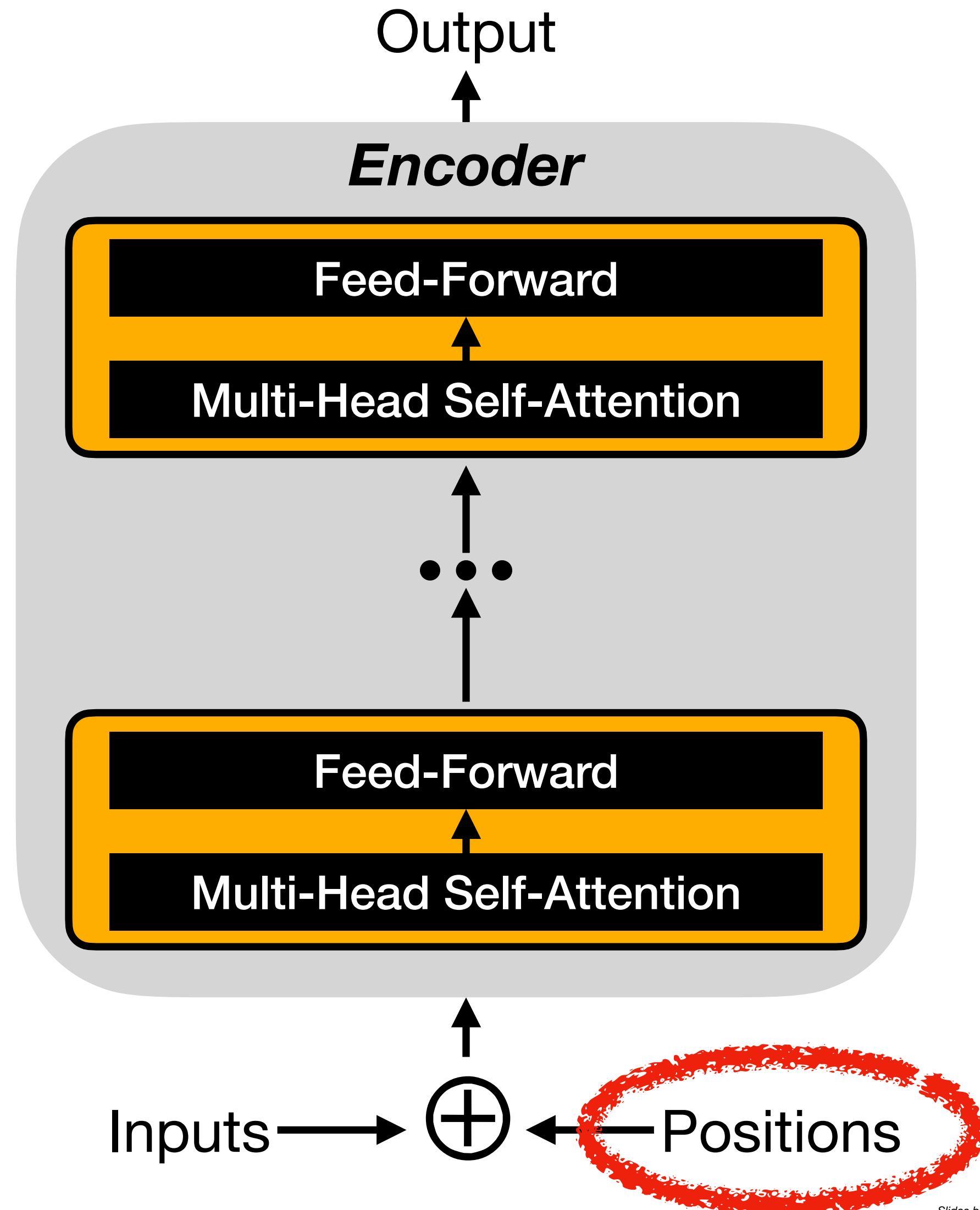


Representing Positions

- Recurrent neural network get information on **token positions**
 - Implicitly, since network applied to **consecutive** positions
- Transformer models need **explicit** information on positions
 - Add **positional encoding** to token embeddings (vector)
 - E.g., use **sine and cosine** functions on position

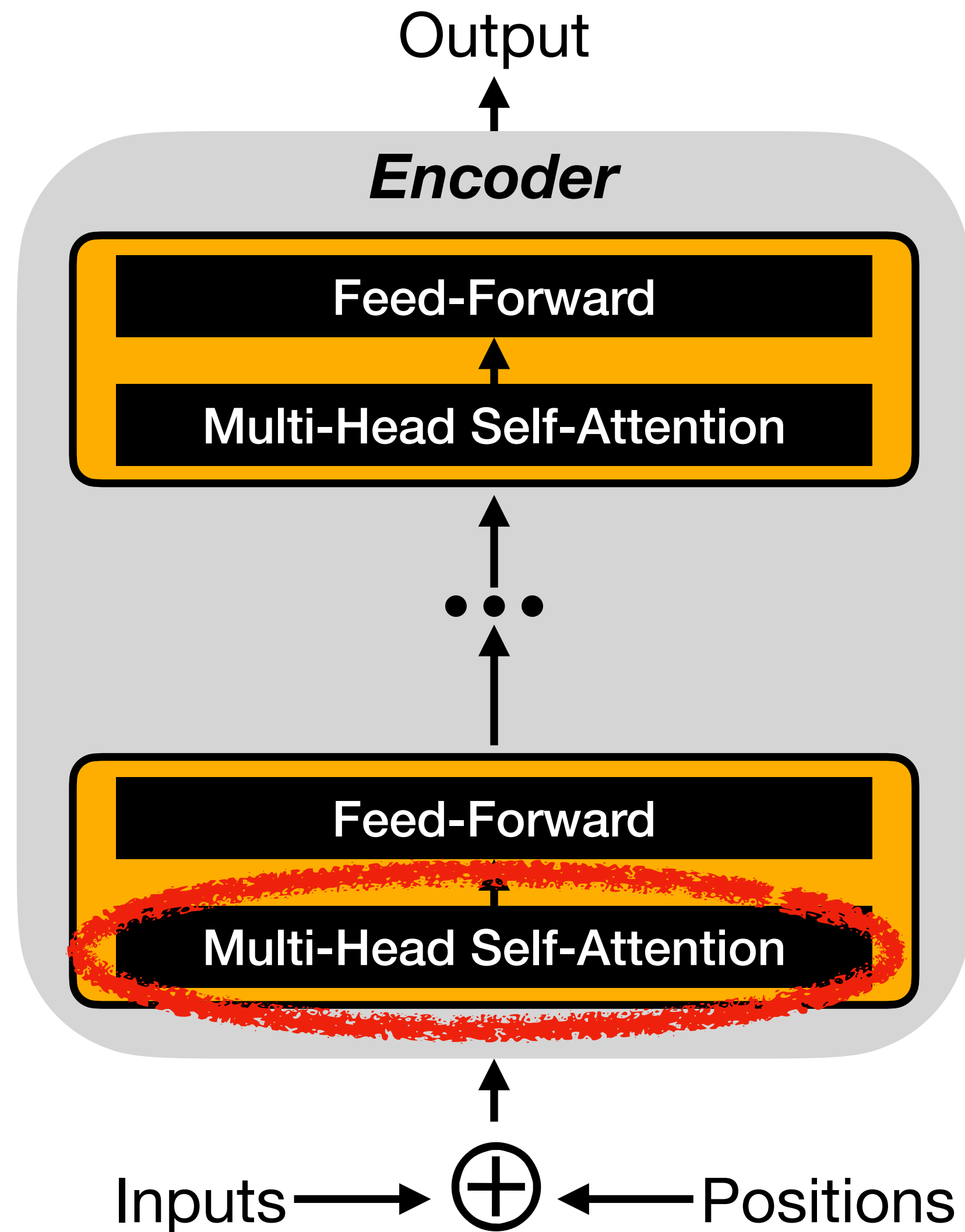
The Transformer

(Details omitted: skip connections, layer normalization, masking)



The Transformer

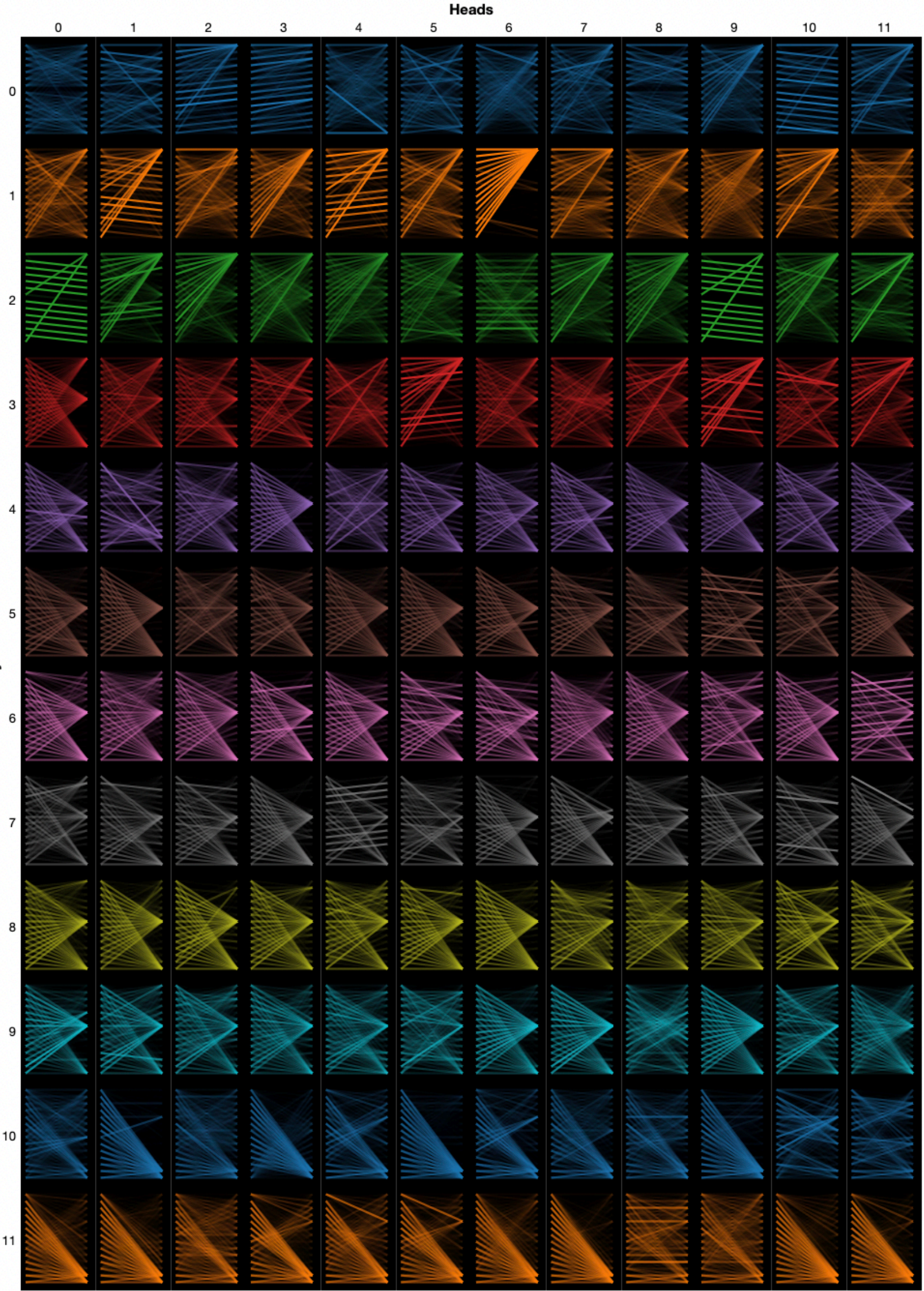
(Details omitted: skip connections, layer normalization, masking)



Multi-Head Self-Attention

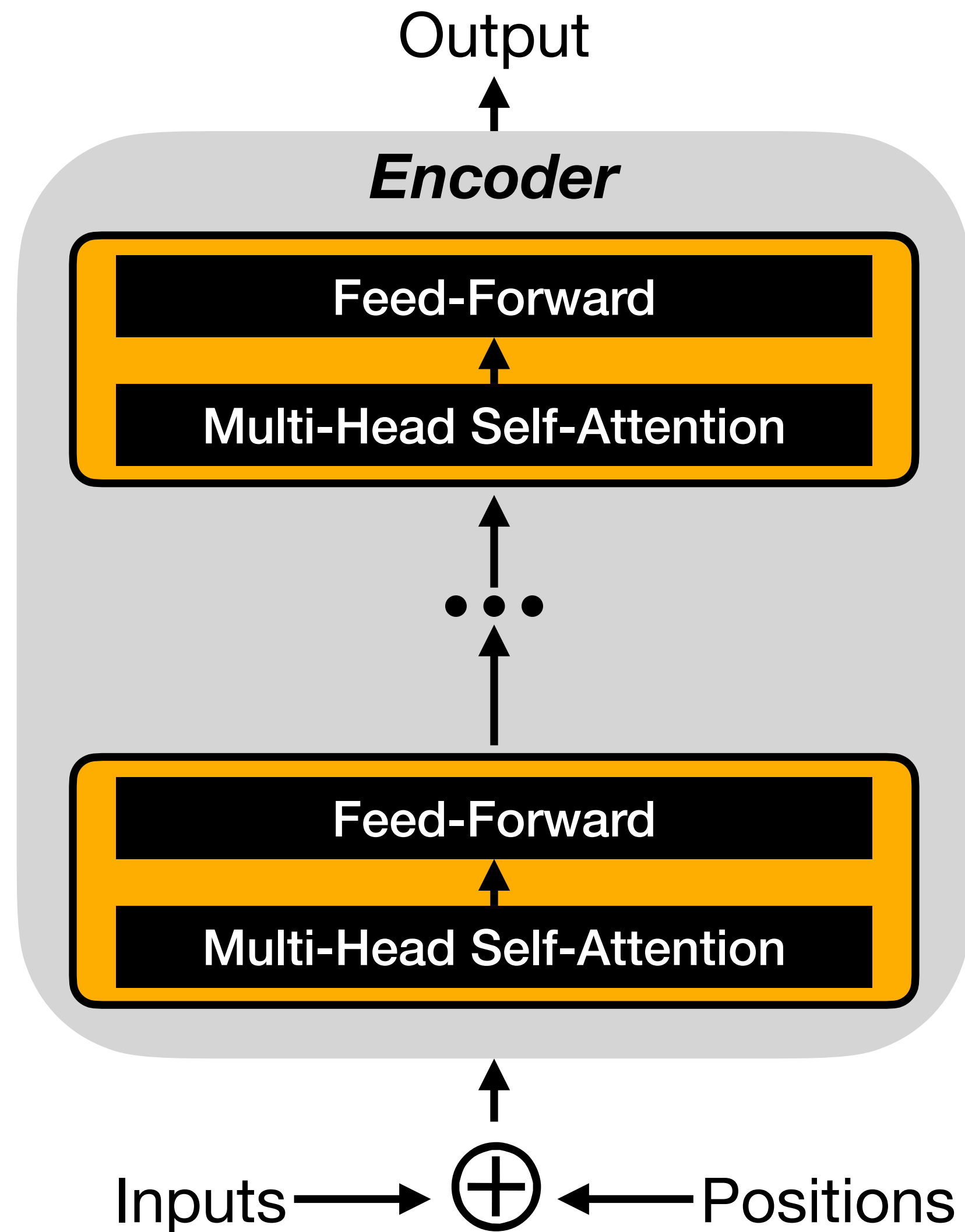
- **Self-Attention**
 - Associate keys and values with input tokens
 - Infer connections between input tokens
- **Multi-Head Attention**
 - Use multiple attention mechanisms per layer
 - Allows considering different connection types

Multi-Head, Multi-Layer Attention Visualization



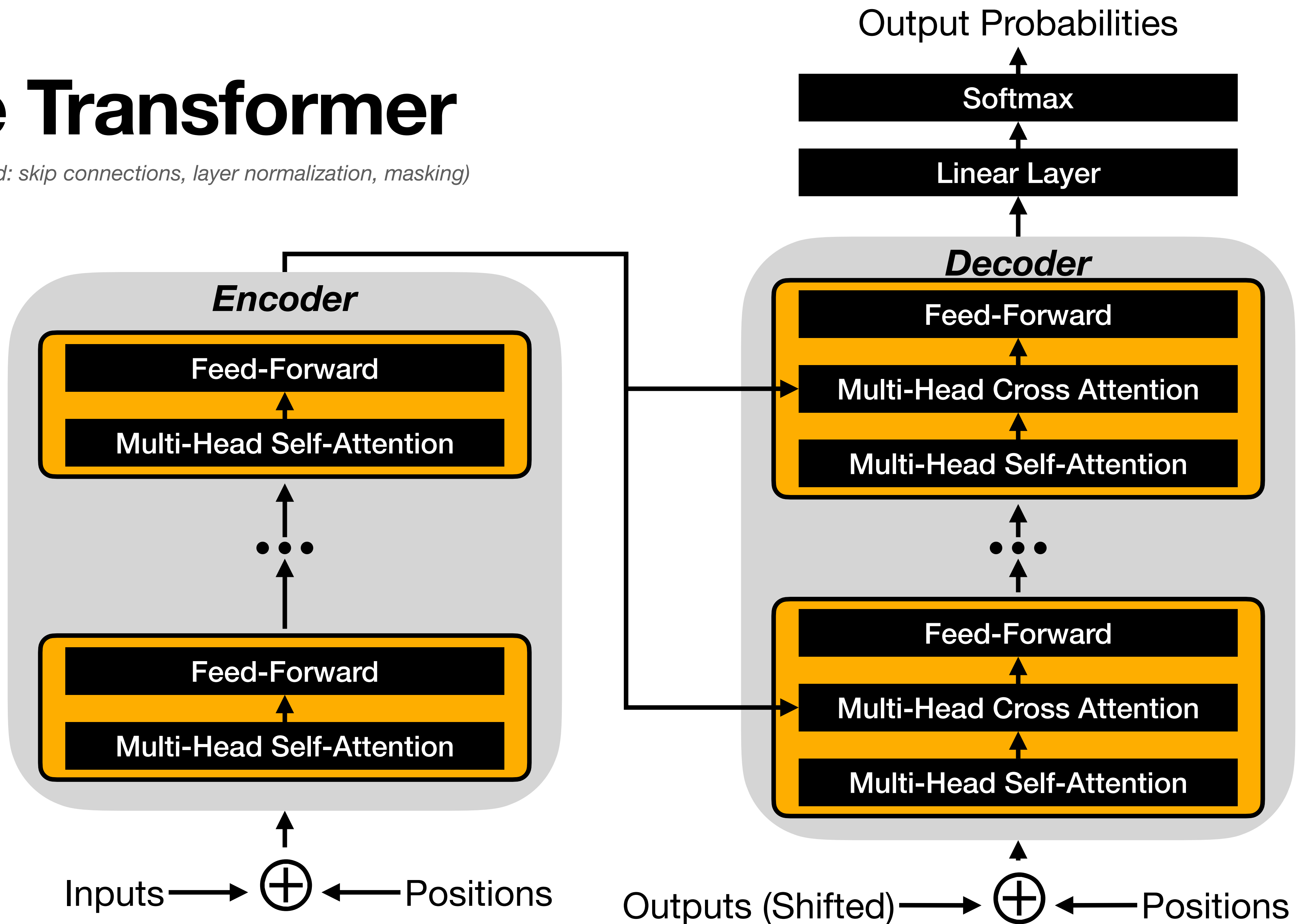
The Transformer

(Details omitted: skip connections, layer normalization, masking)



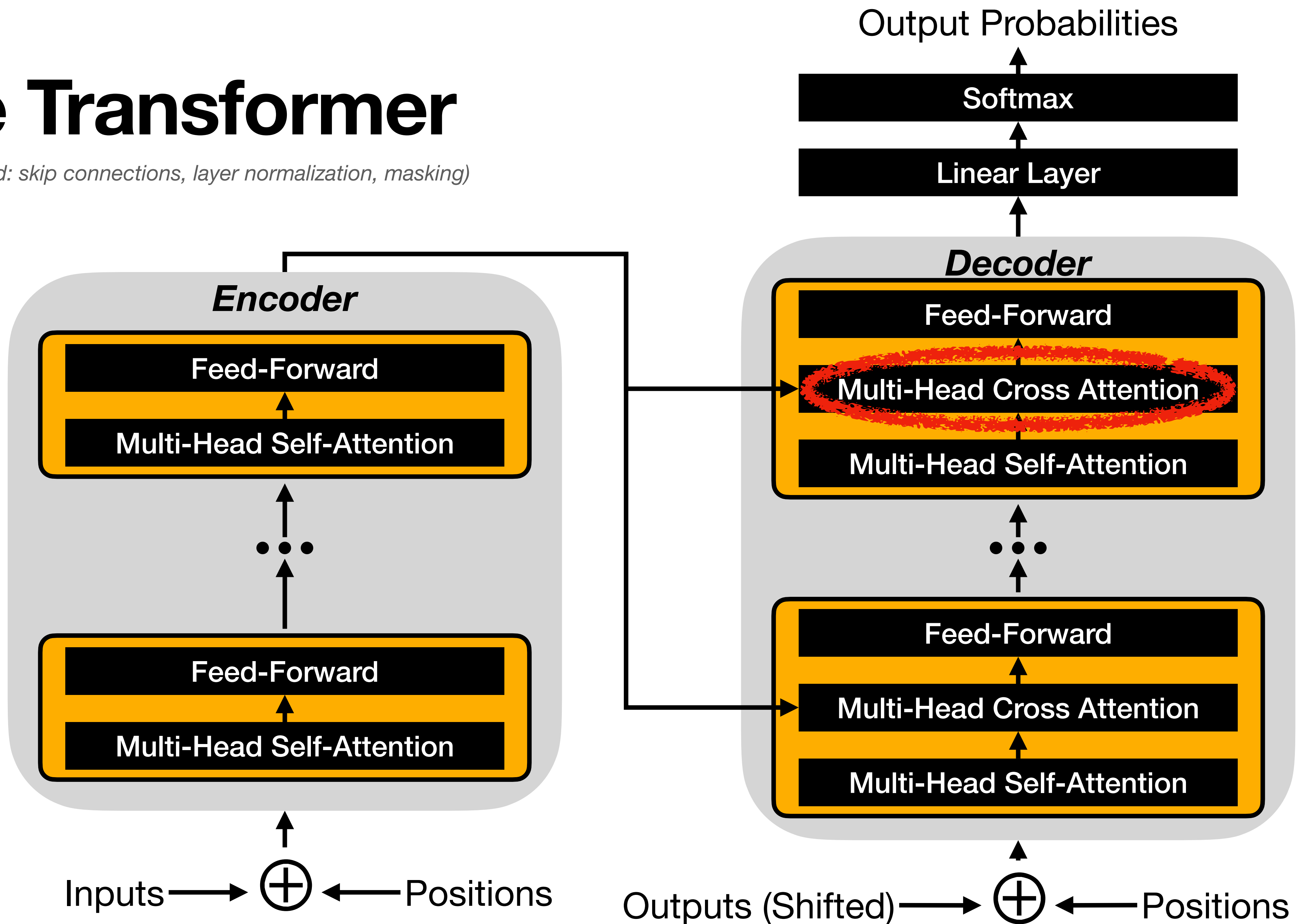
The Transformer

(Details omitted: skip connections, layer normalization, masking)



The Transformer

(Details omitted: skip connections, layer normalization, masking)

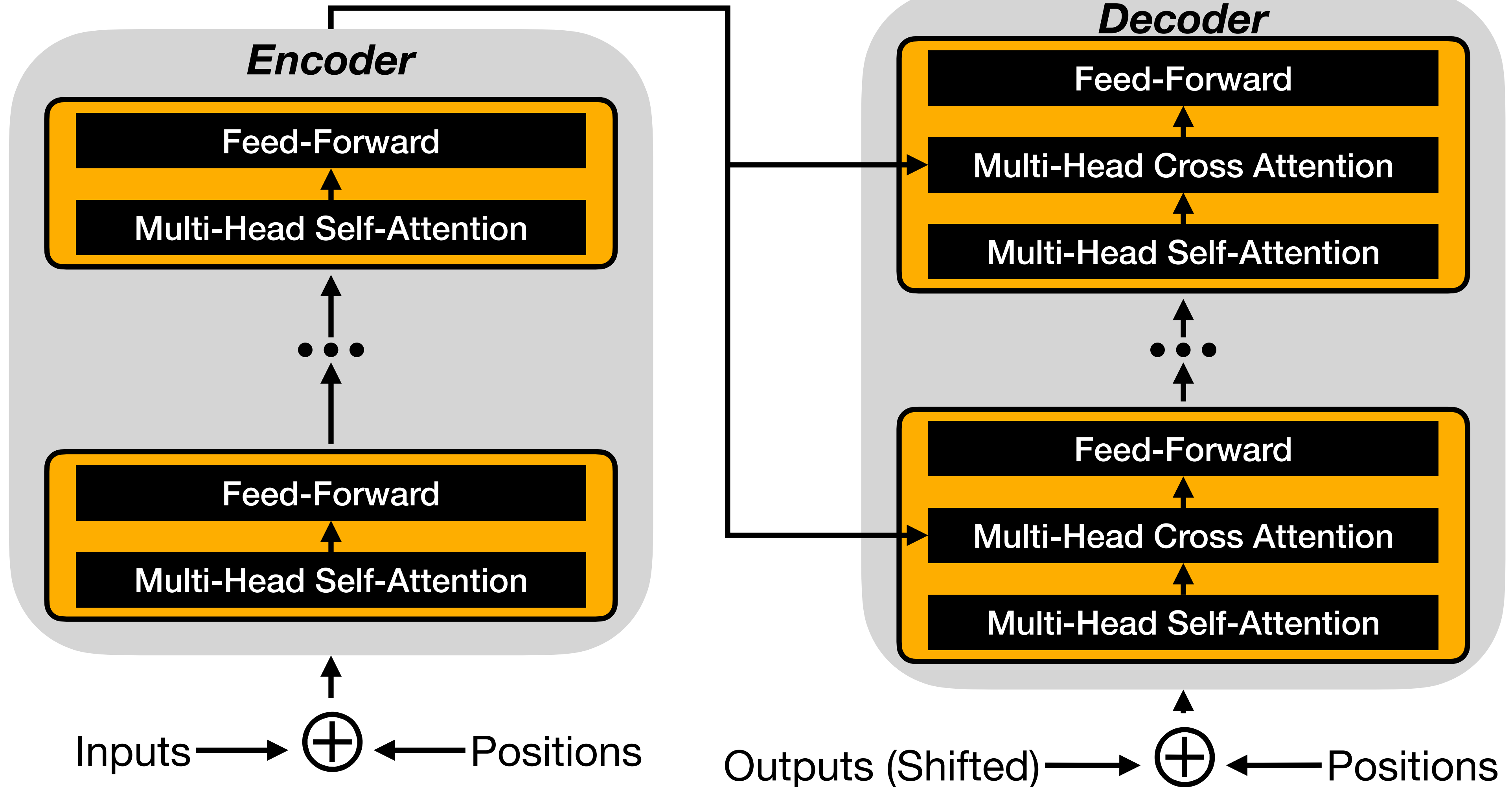


Cross Attention

- Enables decoder to **query** output of encoder
- **Queries** are associated with decoder input
- **Keys/Values** associated with encoder output

The Transformer

(Details omitted: skip connections, layer normalization, masking)

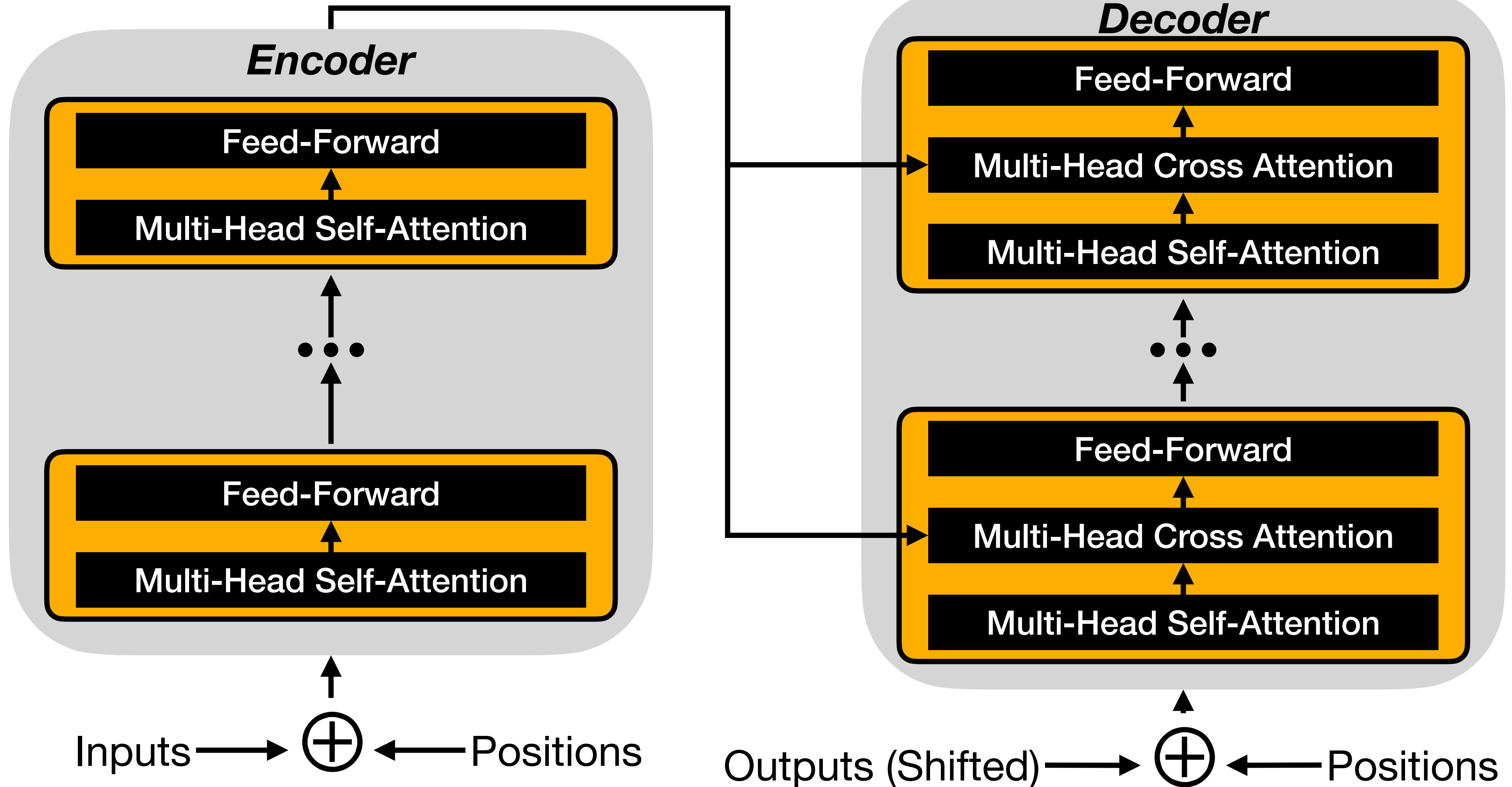


Decoding Methods

- **Greedy**: select most likely token
 - Low diversity, may not maximize sequence probability
- **Beam search**: consider token combinations
- **Sampling** with temperature
 - Temperature chooses degree of randomness
 - From pure sampling ($T=1$) to greedy ($T=0$)
- Sampling **variants**
 - Top-k: only sample among k most likely tokens
 - Nucleus: sample among tokens with total probability p

The Transformer

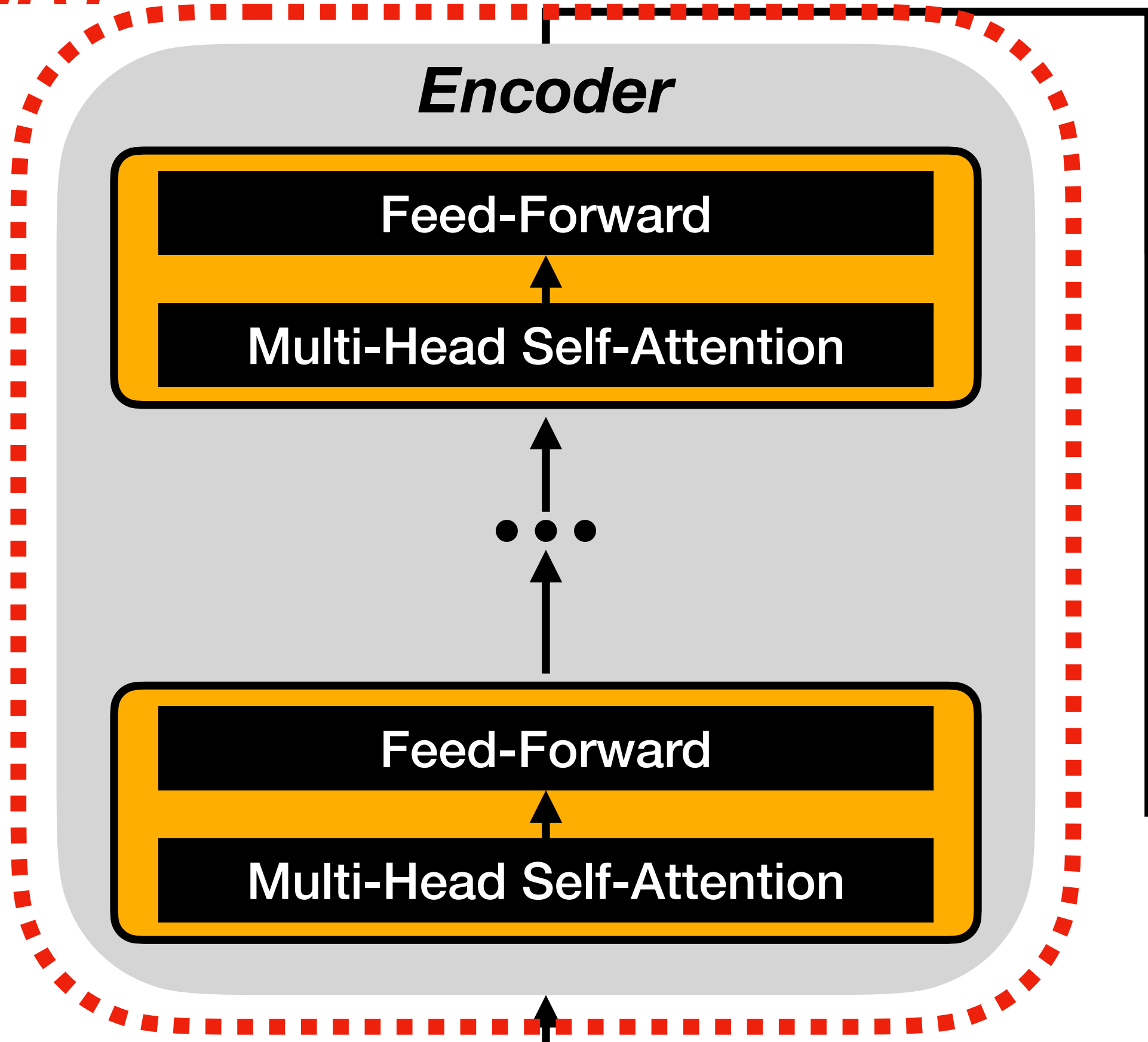
(Details omitted: skip connections, layer normalization, masking)



The Transformer

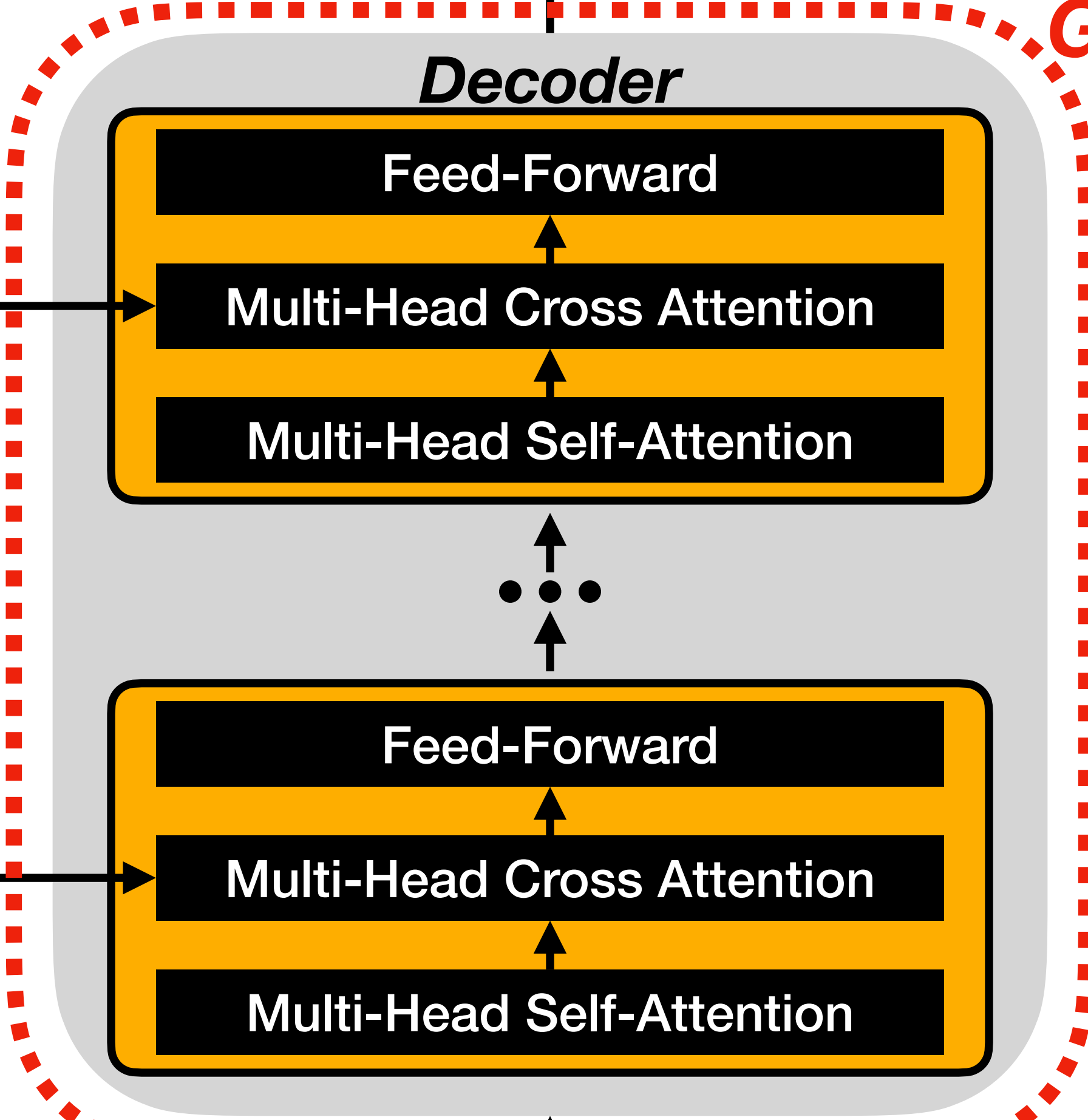
(Details omitted: skip connections, layer normalization, masking)

BERT

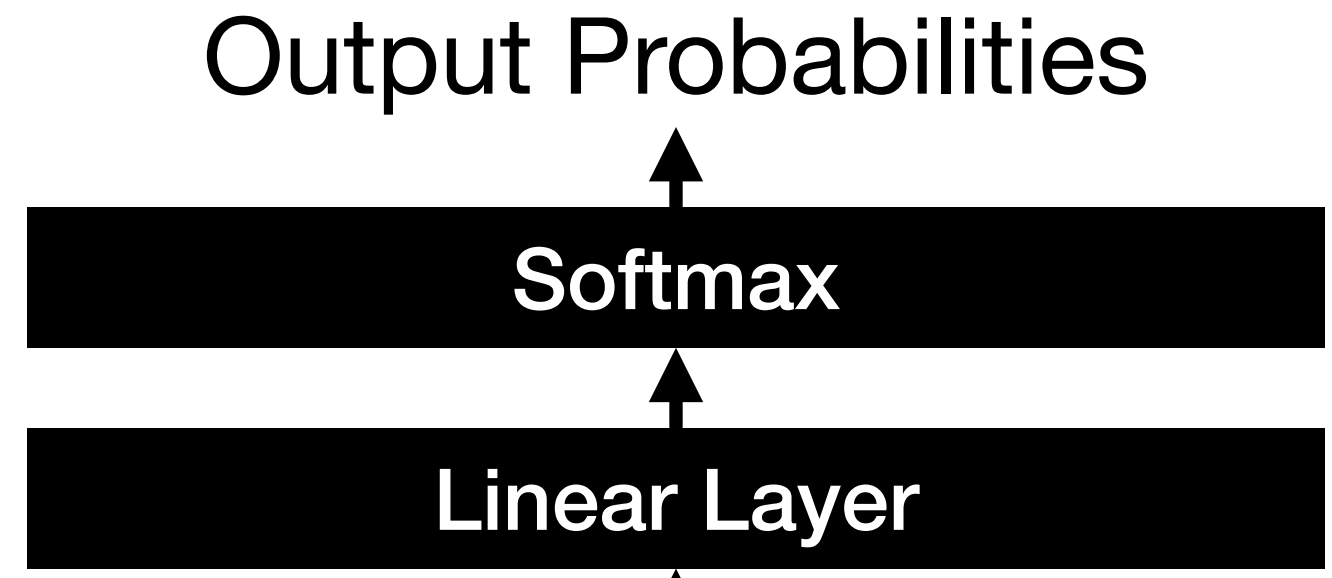


Inputs \rightarrow \oplus \leftarrow Positions

GPT



Outputs (Shifted) \rightarrow \oplus \leftarrow Positions



Transformer: Summary

- Key idea: **attention** mechanisms
- **Multi-layer, multi-head** attention
- Full Transformer: **encoder + decoder**
- May only implement **one of** the two



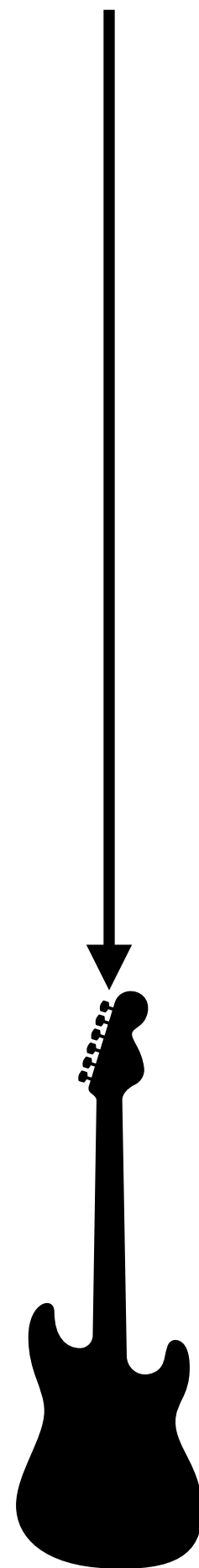
Transfer Learning

Transfer Learning: Idea

Untrained

Transfer Learning: Idea

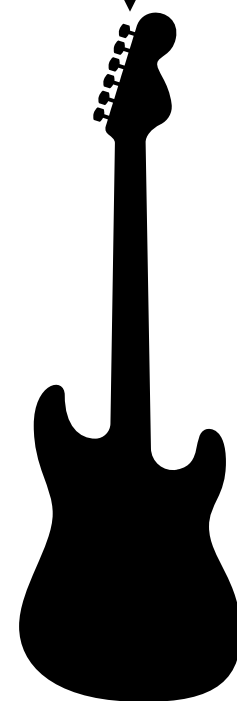
Untrained



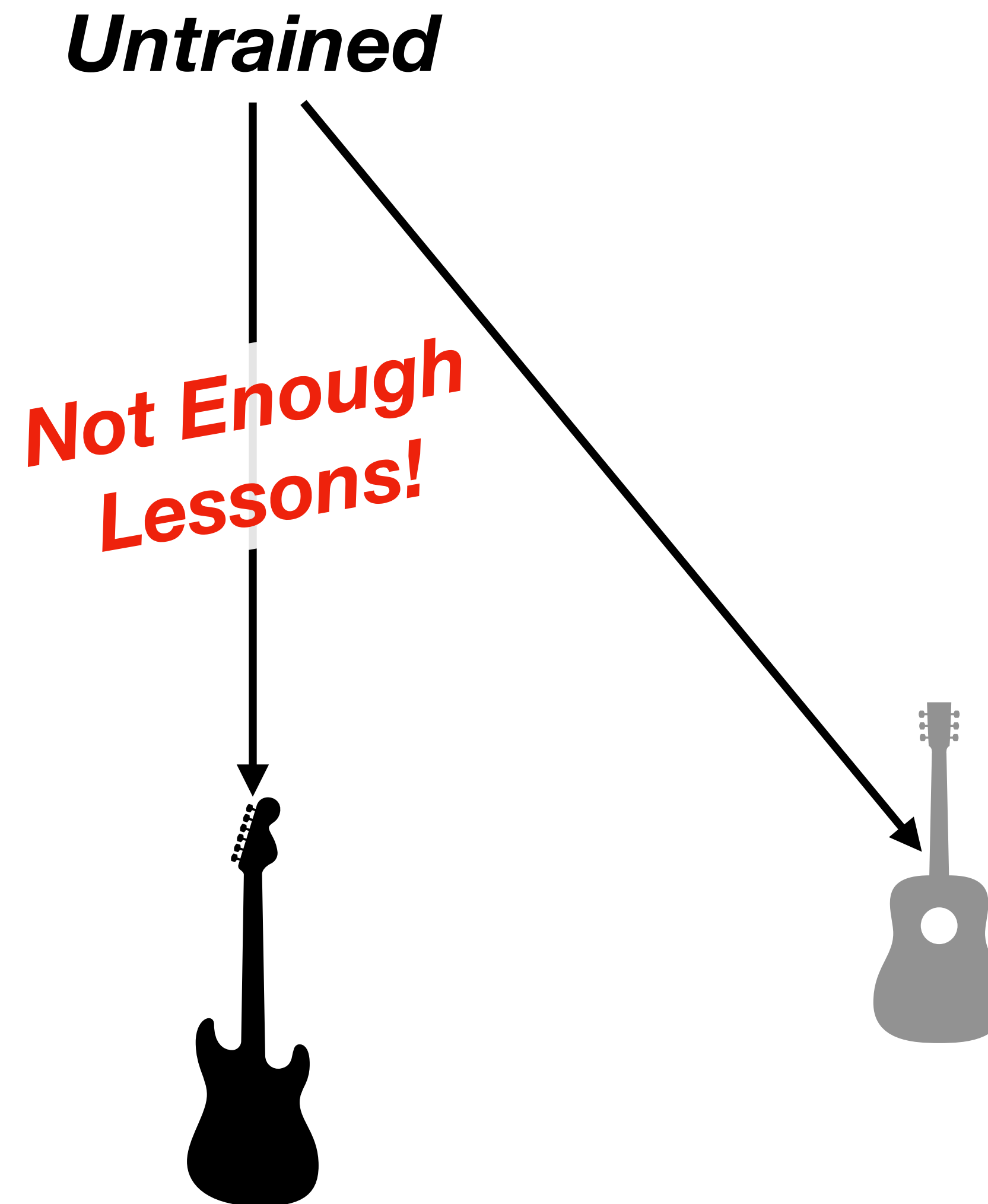
Transfer Learning: Idea

Untrained

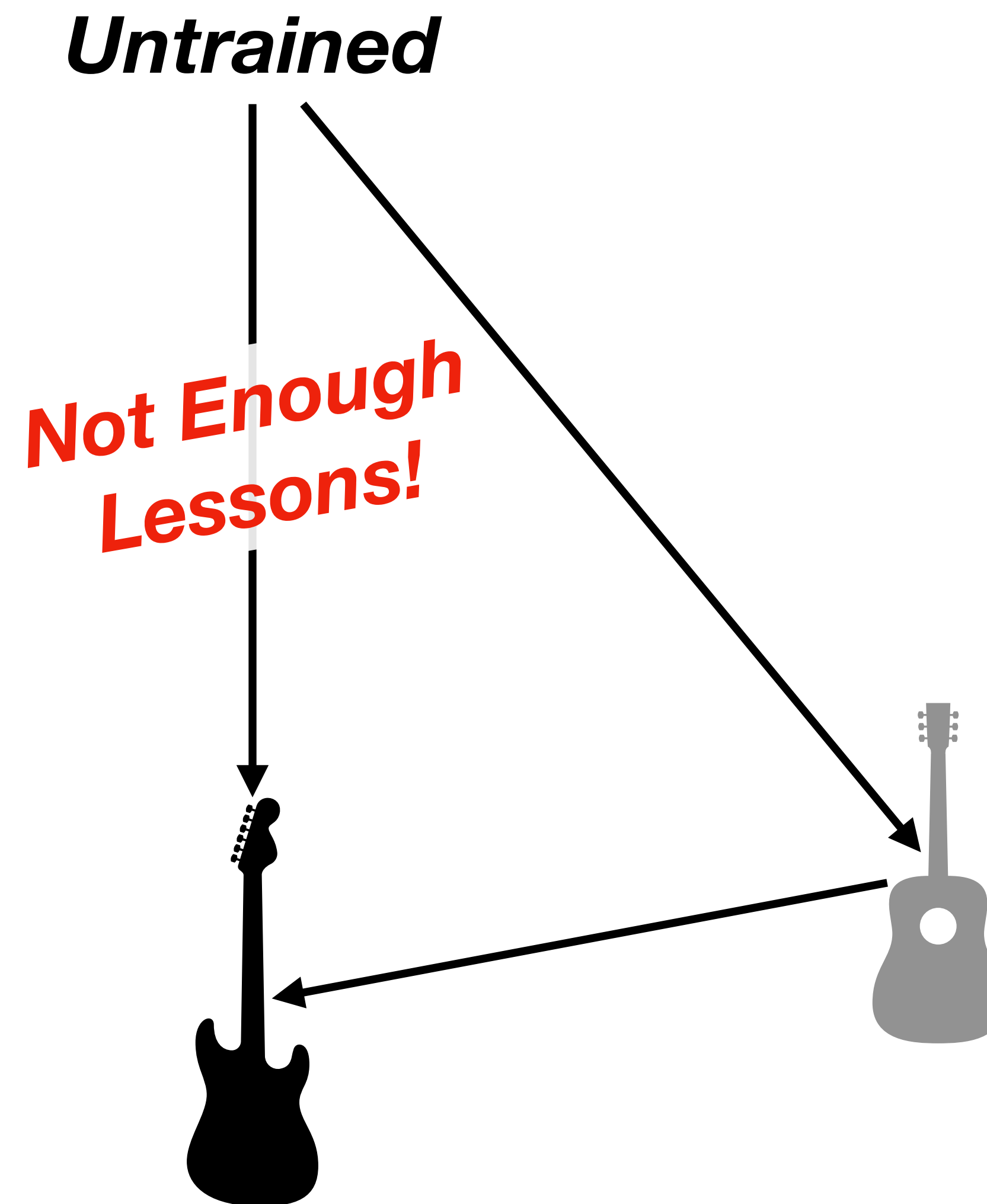
**Not Enough
Lessons!**



Transfer Learning: Idea



Transfer Learning: Idea



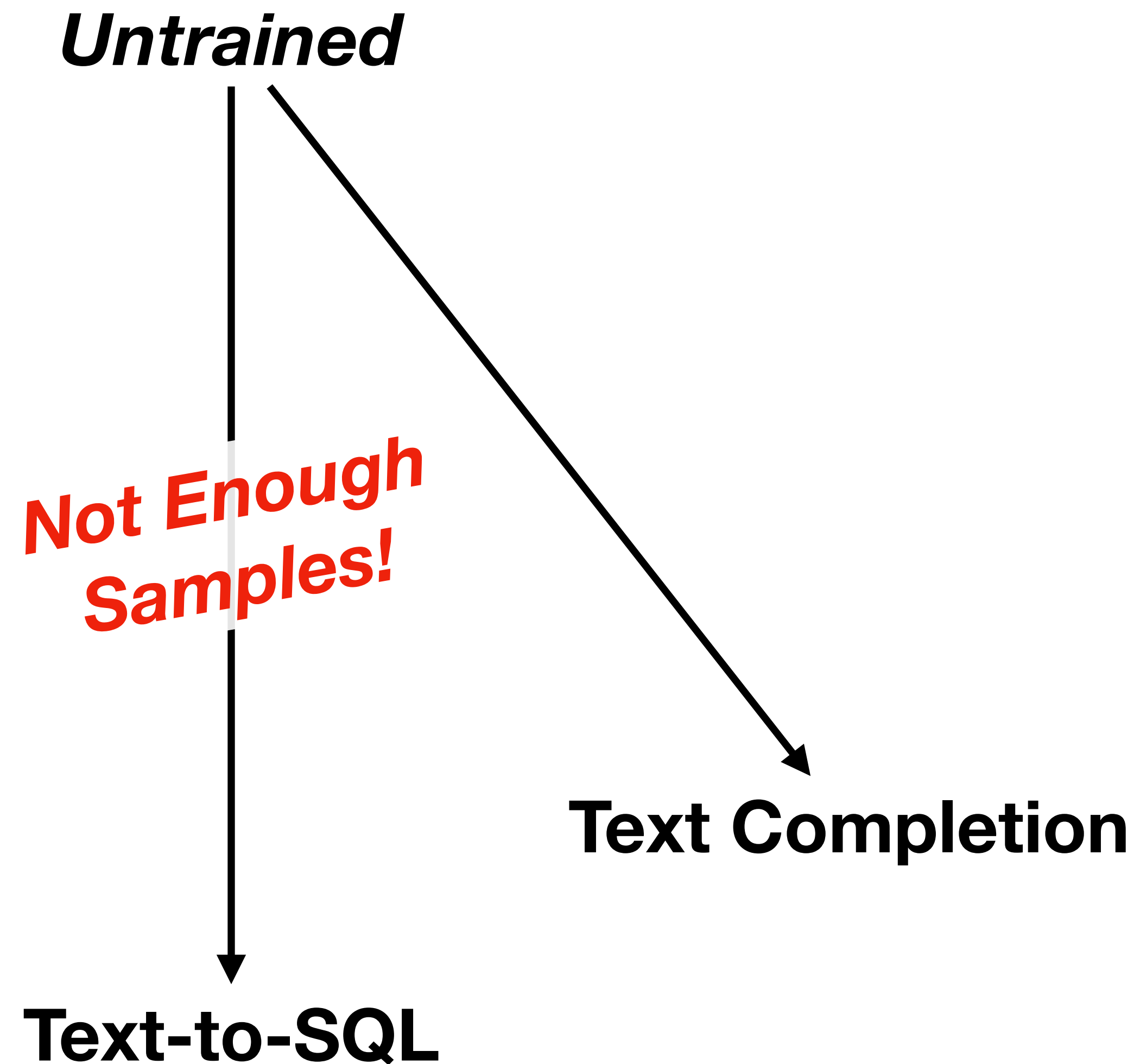
Transfer Learning: Idea

Untrained

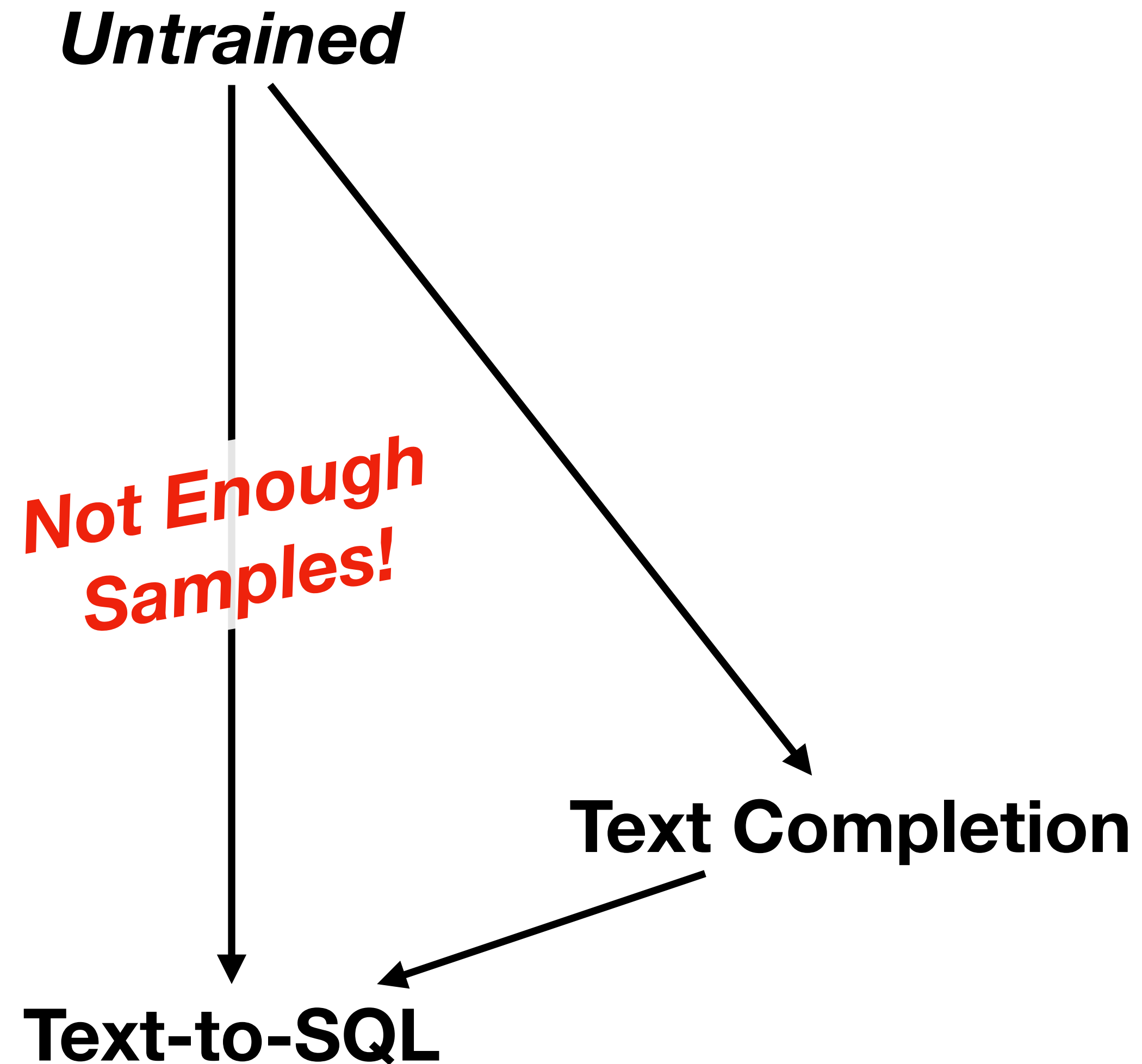
**Not Enough
Samples!**

Text-to-SQL

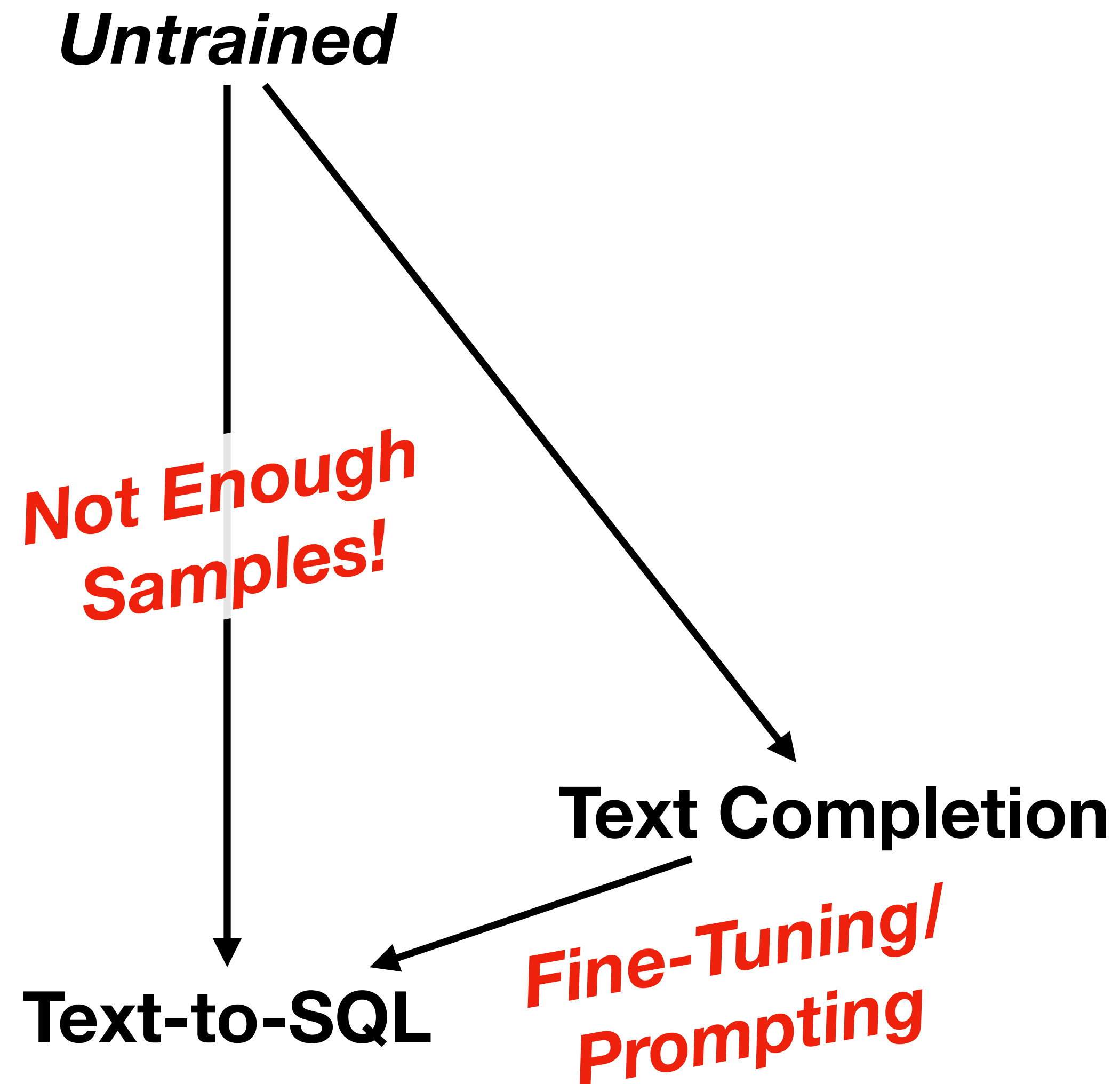
Transfer Learning: Idea



Transfer Learning: Idea



Transfer Learning: Idea



Pre-Training

- Trains skills that are useful for **various** language processing tasks
- Very **expensive** and often performed by large corporations
 - E.g., pre-training GPT-3 costs about 5 million USD
 - Efforts aimed at collaborative pre-training (e.g., BLOOM)
- Two **design decisions**
 - Pre-training objective
 - Pre-training corpus

Pre-Training

- Trains skills that are useful for **various** language processing tasks
- Very **expensive** and often performed by large corporations
 - E.g., pre-training GPT-3 costs about 5 million USD
 - Efforts aimed at collaborative pre-training (e.g., BLOOM)
- Two **design decisions**
 - Pre-training objective
 - Pre-training corpus

Causal Language Modeling

- **Source Text**
 - The quick brown fox jumps over the lazy dog.
- **Training Sample**
 - **Input:** The quick brown fox
 - **Output:** jumps
- E.g., used for training **GPT** models

Current techniques restrict the power of the pre-trained representations ...
The major limitation is that standard language models are *unidirectional* ...

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova

Google AI Language

{jacobdevlin, mingweichang, kentonl, kristout}@google.com

Abstract

We introduce a new language representation model called **BERT**, which stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers. Unlike recent language representation models (Peters et al., 2018a; Rad-

There are two existing strategies for applying pre-trained language representations to downstream tasks: *feature-based* and *fine-tuning*. The feature-based approach, such as ELMo (Peters et al., 2018a), uses task-specific architectures that include the pre-trained representations as addi-

Masked Language Modeling

- **Source Text**
 - The quick brown fox jumps over the lazy dog.
- **Training Sample**
 - **Input:** The quick brown fox [MASK] over the lazy dog.
 - **Output:** jumps
- Provides **more context** that can be used for prediction

A key advantage ... is the noising flexibility; arbitrary transformations can be applied to the original text, including changing its length.

BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension

**Mike Lewis*, Yinhan Liu*, Naman Goyal*, Marjan Ghazvininejad,
Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, Luke Zettlemoyer**
Facebook AI

`mikelewis@fb.com, yinhan@ai2incubator.com, naman@fb.com`

Abstract

We present BART, a denoising autoencoder for pretraining sequence-to-sequence models. BART is trained by (1) corrupting text with an arbitrary noising function, and (2) learning a model to reconstruct the original text. It uses

masked tokens (Joshi et al., 2019), the order in which masked tokens are predicted (Yang et al., 2019), and the available context for replacing masked tokens (Dong et al., 2019). However, these methods typically focus on particular types of end tasks (e.g. span prediction, generation, etc.), limiting their applicability.

In this paper, we present BART, which pre-trains

Denoising: Token Deletion

- **Source Text**
 - The quick brown fox jumps over the lazy dog.
- **Training Sample**
 - **Input:** The brown fox lazy dog.
 - **Output:** The quick brown fox jumps over the lazy dog.
- Note: number of output token **not given** a-priori

⇒ Many Possible Objectives

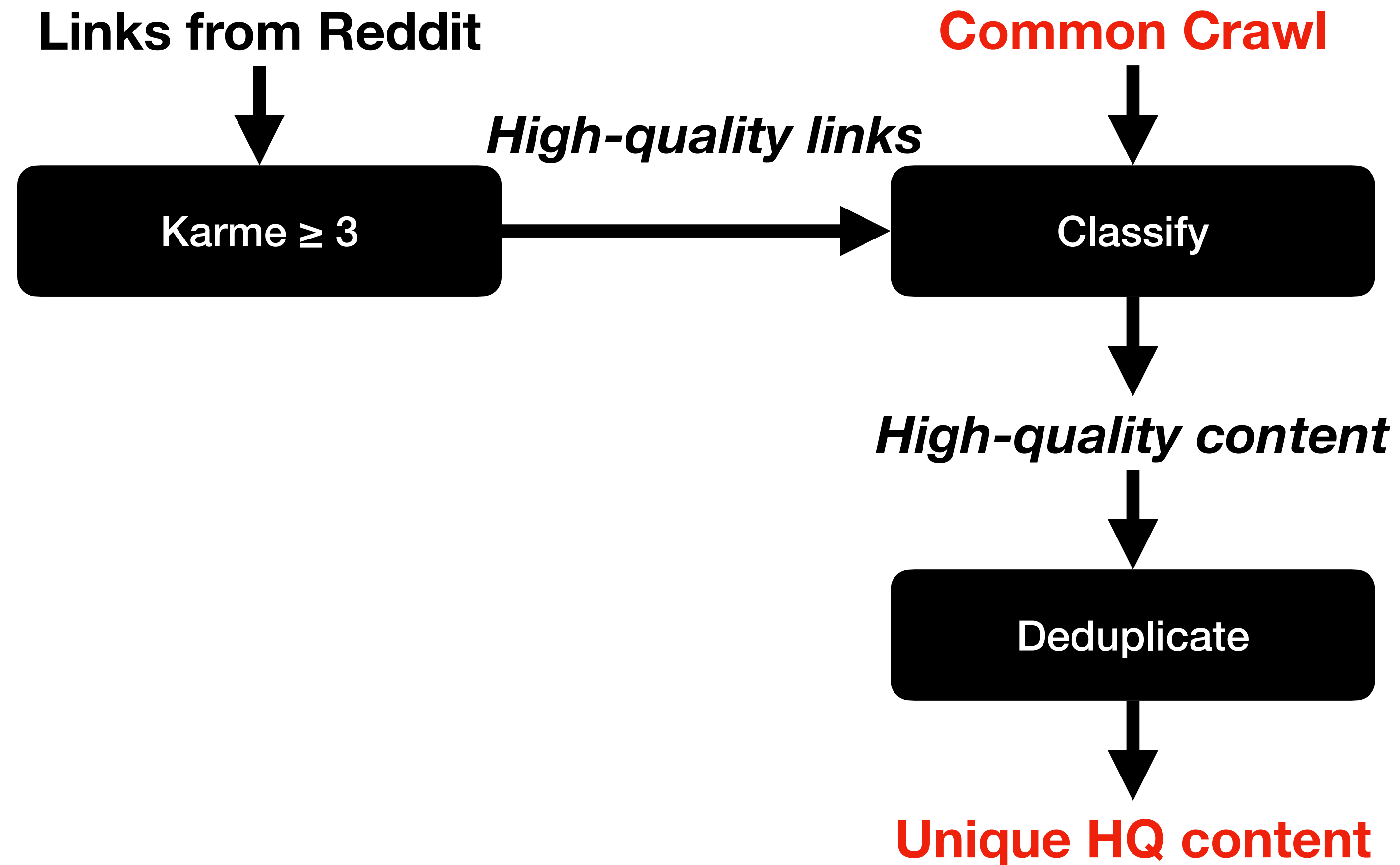
Pre-Training

- Trains skills that are useful for **various** language processing tasks
- Very **expensive** and often performed by large corporations
 - E.g., pre-training GPT-3 costs about 5 million USD
 - Efforts aimed at collaborative pre-training (e.g., BLOOM)
- Two **design decisions**
 - Pre-training objective
 - Pre-training corpus

Considerations when Selecting a Corpus

- **Amount** of training data (match model size)
- **Quality** of training data (e.g., curation by humans)
- **Legal** considerations (e.g., rights of authors)
- Minimizing implicit **bias** (e.g., dominant country)
- Ensure **diversity** (e.g., multiple languages)
- ...

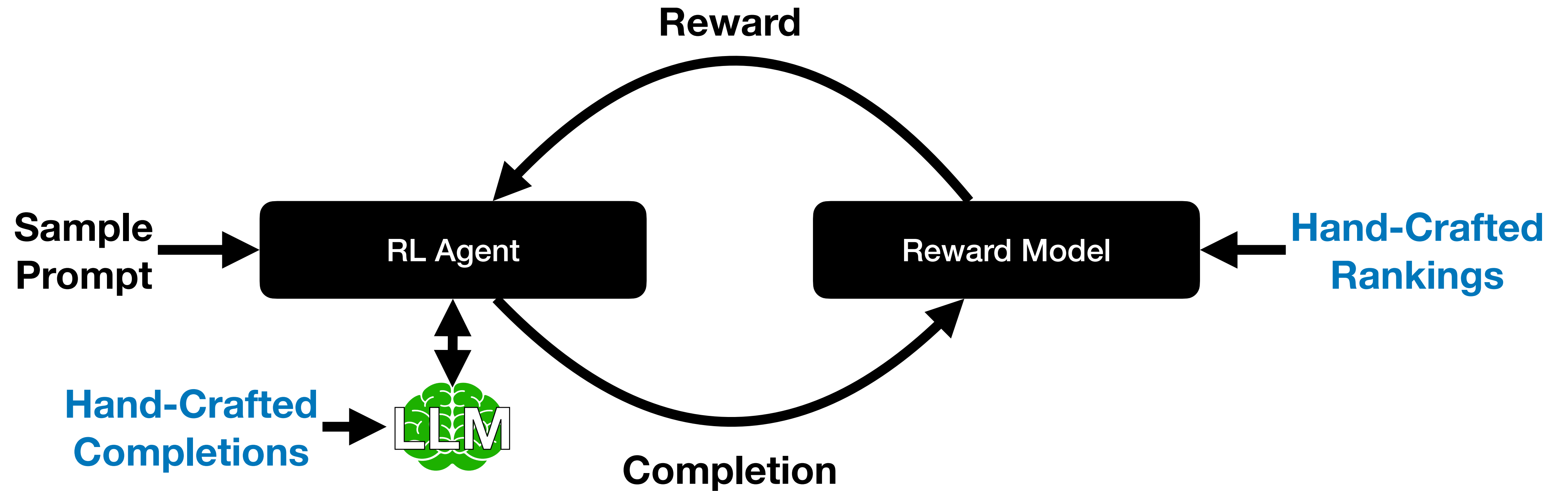
GPT-3 Pre-Training Data



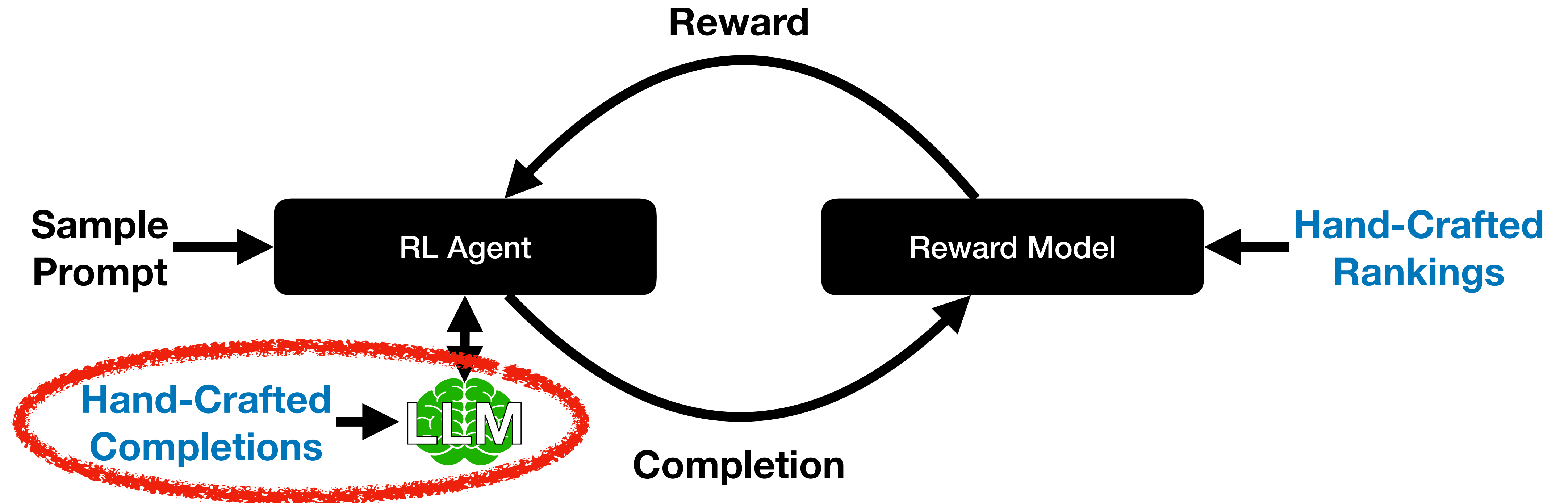
Alignment

- Models trained for completion may produce **mis-aligned** output
 - Does not follow instructions
 - Untruthful (hallucinations)
 - Toxic or harmful content
- Add training stage for **alignment**
 - Exploits manually generated labels

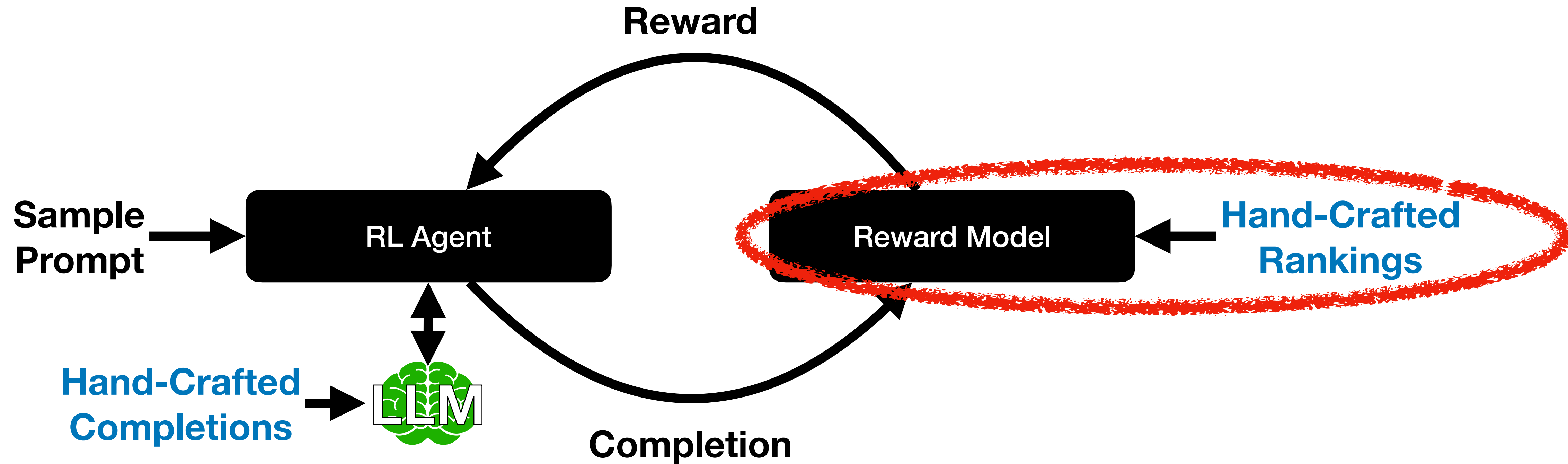
Alignment for GPT-4



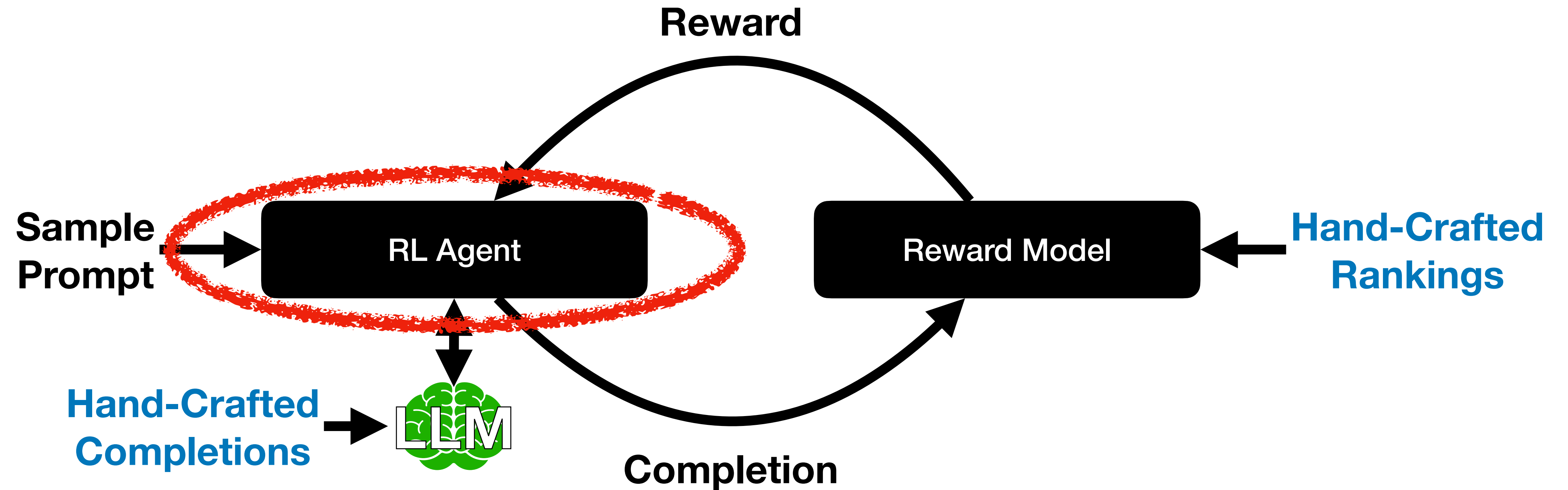
Alignment for GPT-4



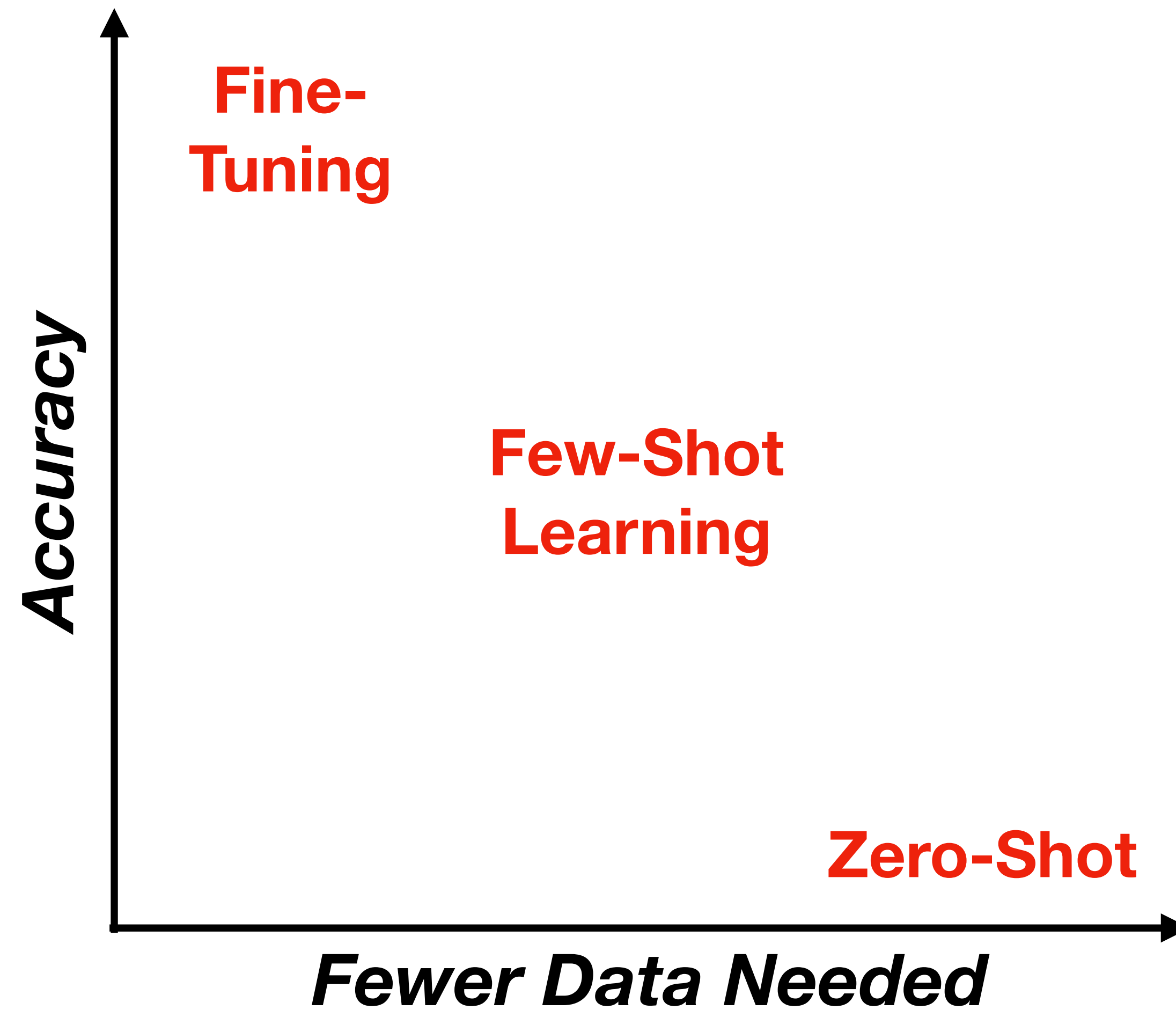
Alignment for GPT-4



Alignment for GPT-4



Task Specialization Methods

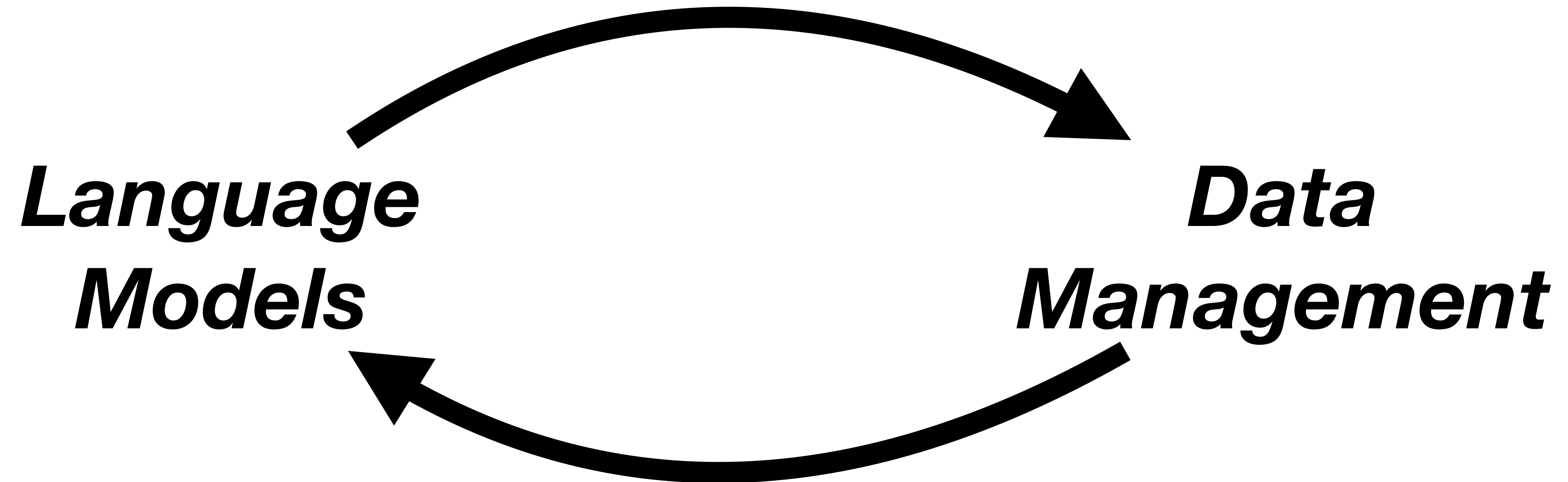


Transfer Learning: Summary

- Use **pre-training** on unlabeled data
 - Different objectives & corpora
- **Alignment**
 - Refine model using manual labels
- **Specialization** to new tasks
 - Fine-tuning: change weights
 - Prompting: specify as input

Conclusion

Research Opportunities



Conclusions

- Significant **progress** in natural language processing
 - **Transformer** Model
 - **Transfer** Learning
- Various **interfaces** and libraries
- New **applications** in data management



www.itrummer.org

itrummer@cornell.edu

Demo: Visualizing Attention

<https://colab.research.google.com/drive/1DG2h6uakCsSVmU0Vem0E5qUGWeADTqe5>