Rapid Application Development using Large Language Models

Speaker Name, Title | Event/Date

Large Language Models

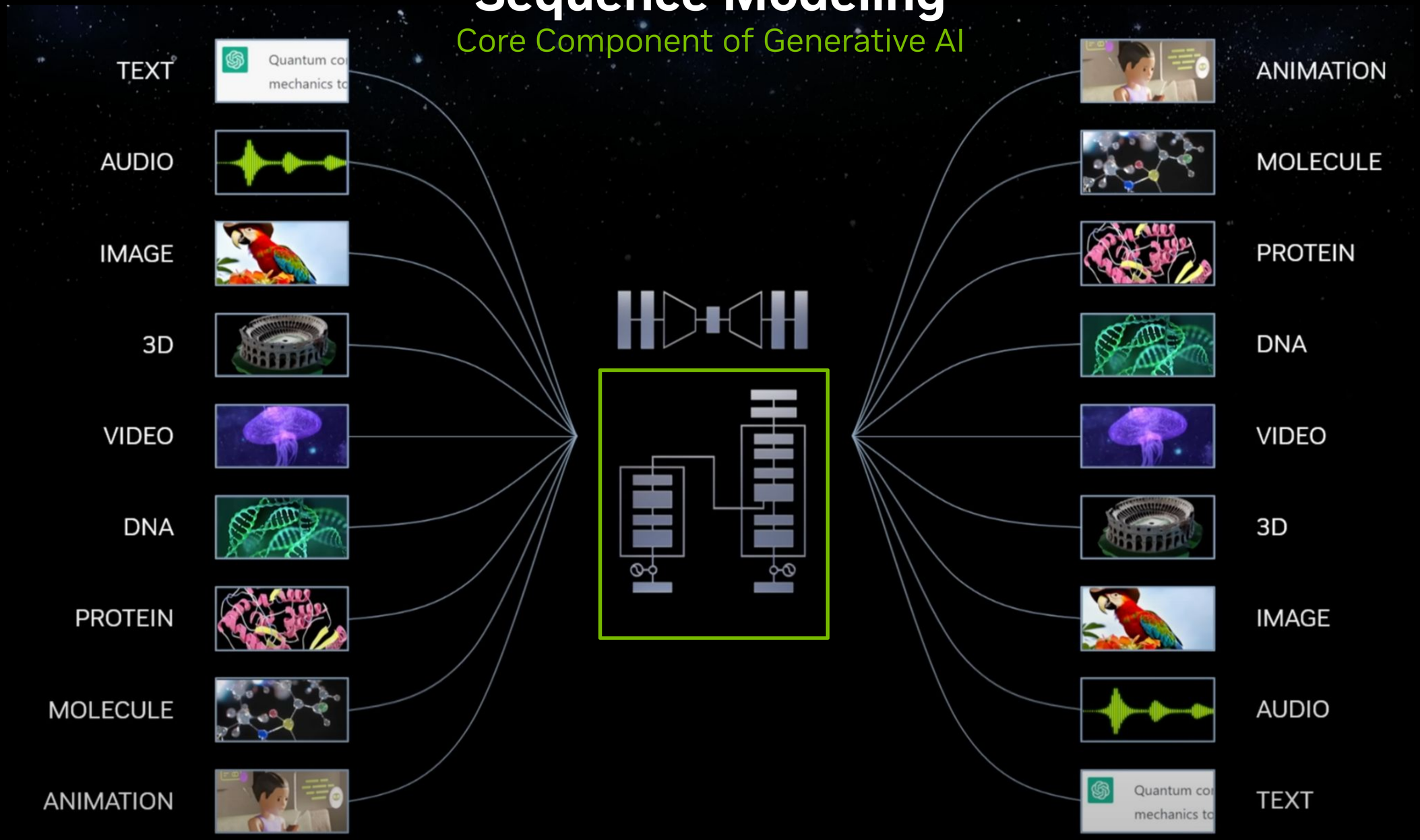
Backbones for Language Understanding

Backbones for language tasks, including classification and generation.

https://www.nvidia.com/content/dam/en-zz/Solutions/lp/large-language-models-ebook/nvidia-llm-ebook-og.jpg

Sequence Modeling

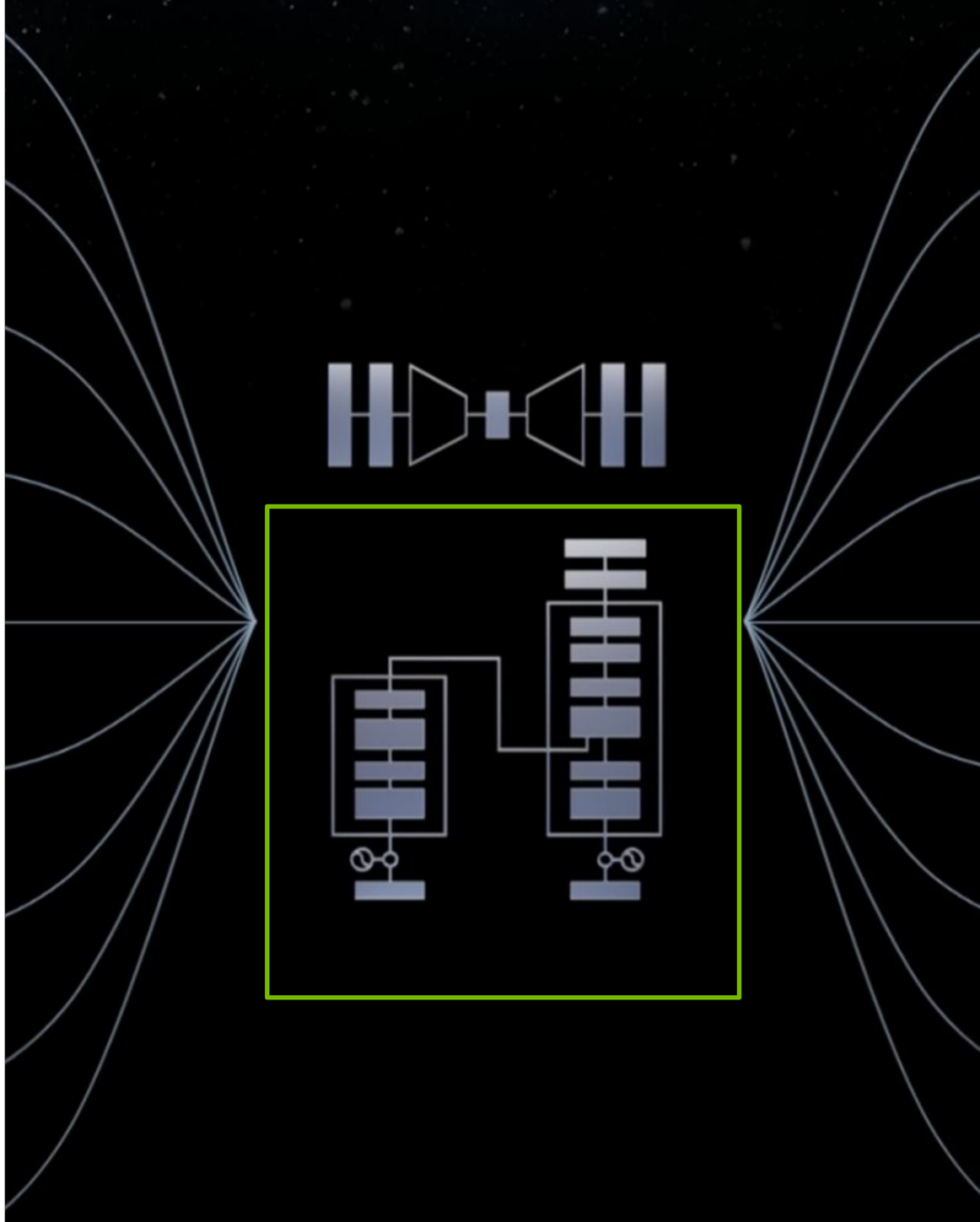
Core Component of Generative AI



Course Objectives

Rapid LLM Application Development

- **Understand How LLMs Work**
  - Architectures
  - Intuitions
  - Design Paradigms
- **Use LLMs To Solve Interesting Problems**
  - Language Reasoning
  - Text Generation
  - Chat Assistants

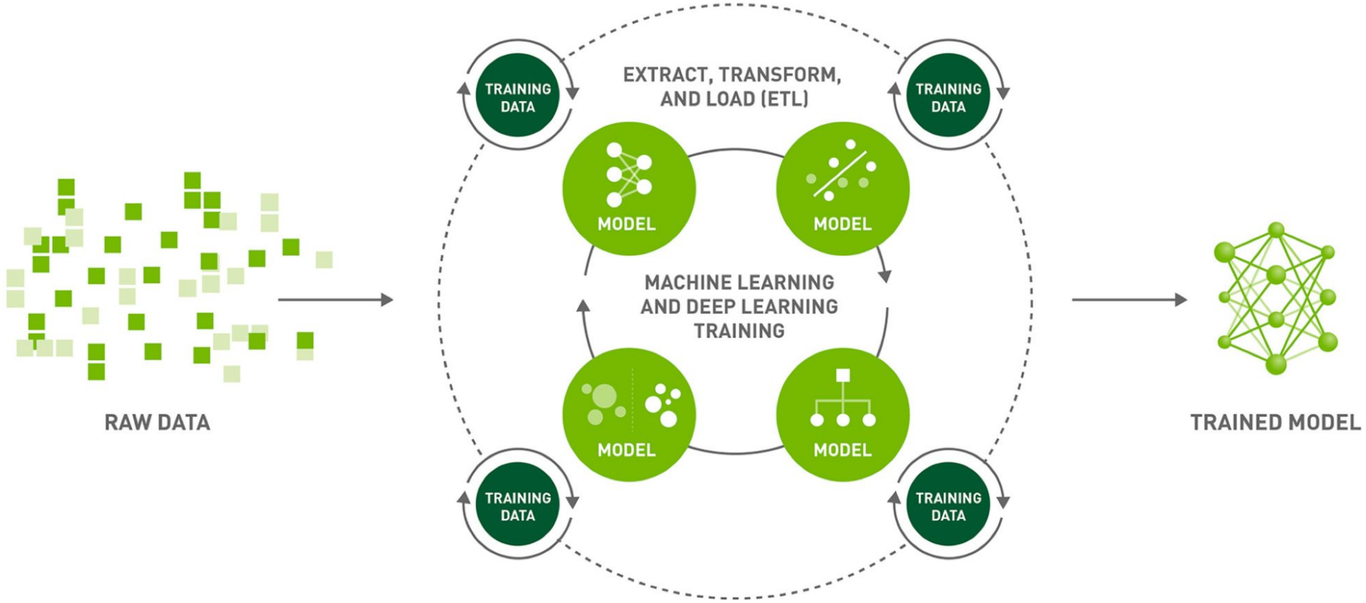


python™ NumPy

Keras PyTorch

Prerequisites

Fundamentals of Deep Learning
Intermediate Python Required



Core Tools

Open-Source LLM Ecosystem

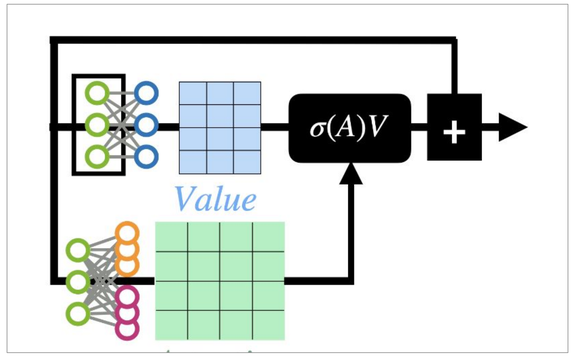
Transformers

LangChain

Course Schedule

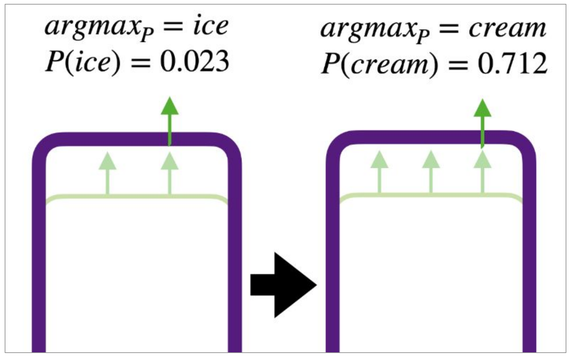
What's The Plan?

Getting Started

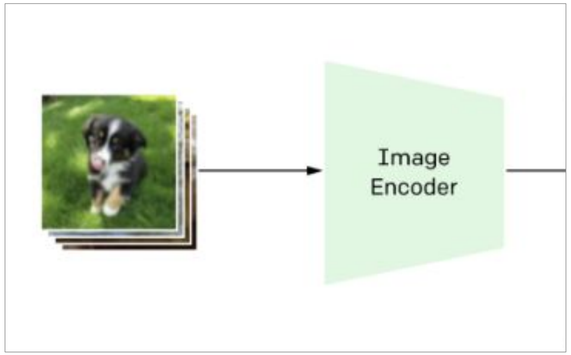


Transformer Intuitions

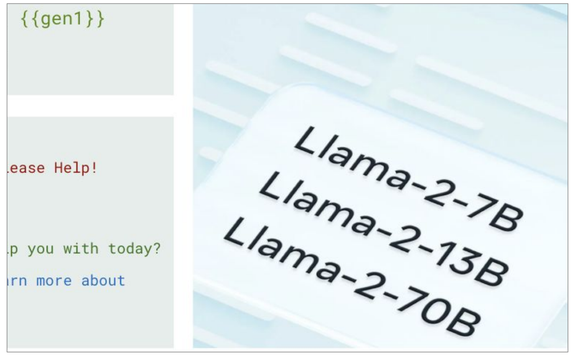
Encoder Tasks



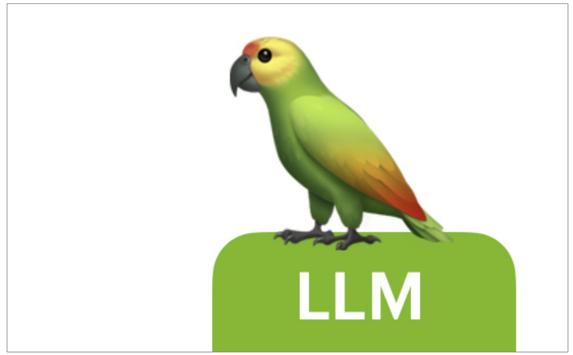
Seq2Seq



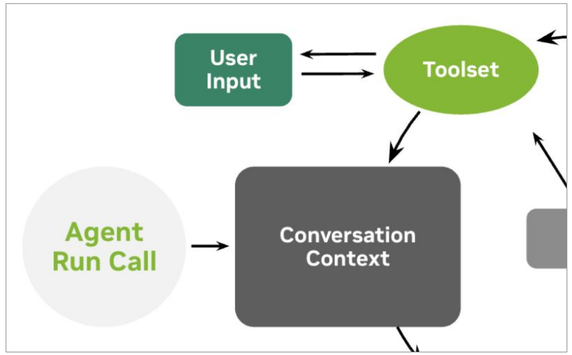
Multimodality



Text Generation



Stateful LLMs



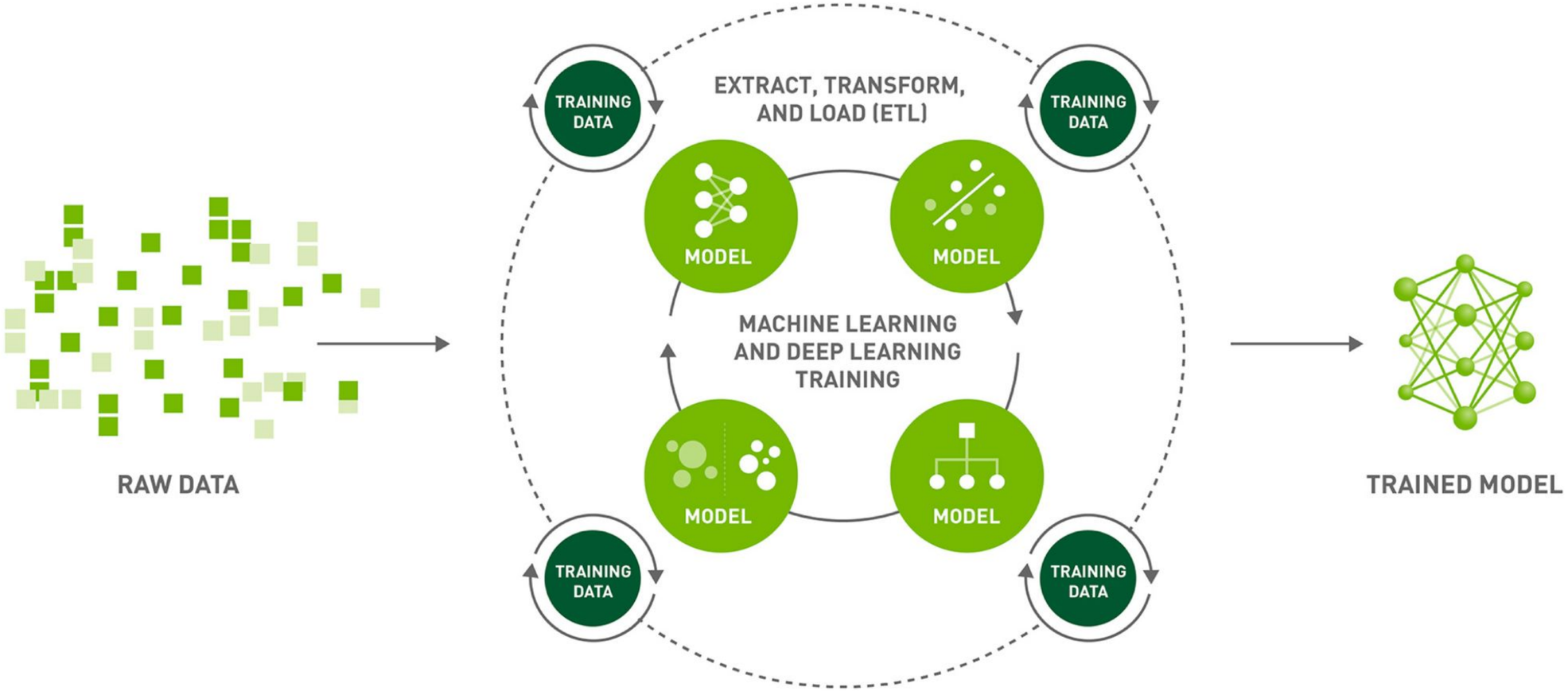
Agent Assessment

Part 1: From DL To LLMs

- From Deep Learning to Language Modeling
- Pulling In Our First Model
- Model Architecture Intuitions
- Using Encoders For Encoding

Deep Learning

(Machine Learning with Deep Models)

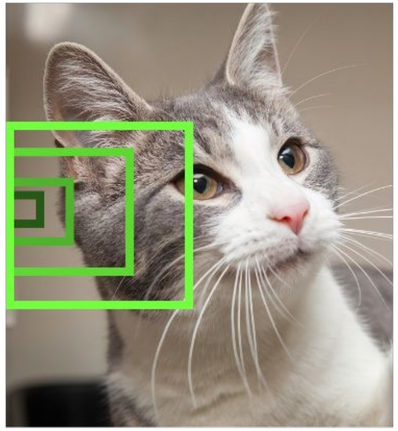


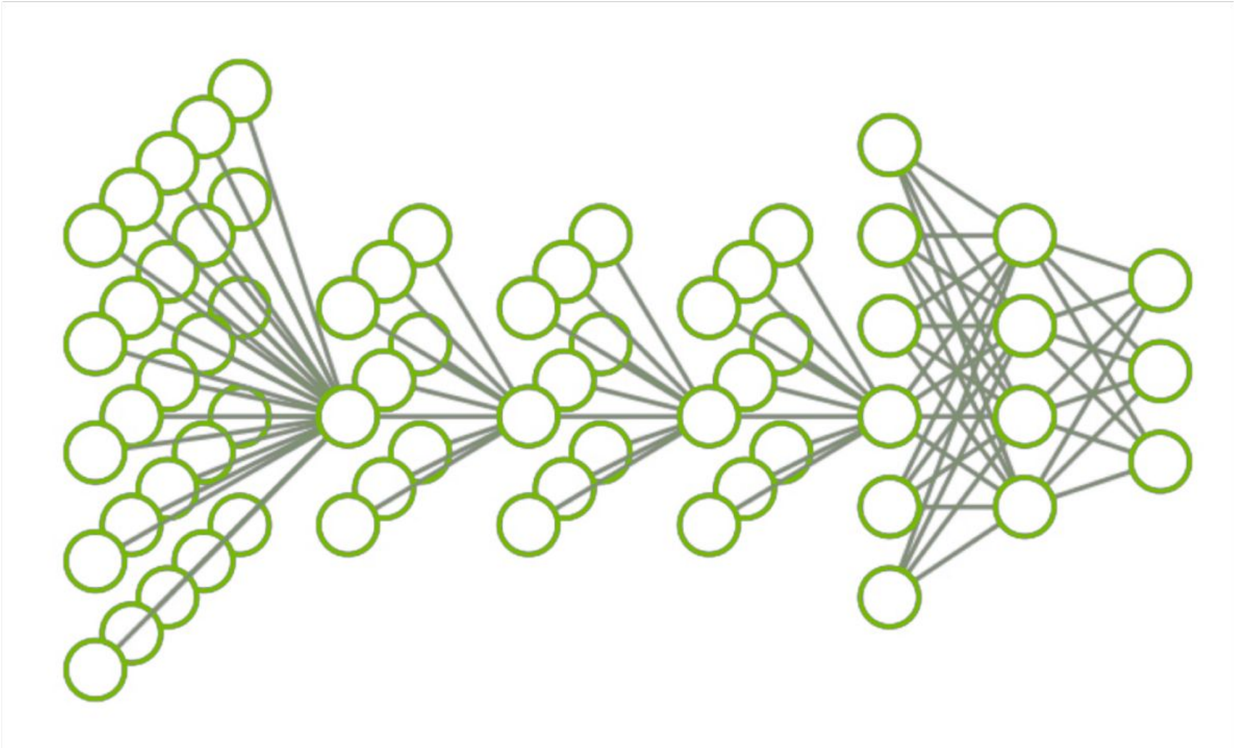
https://www.nvidia.com/en-us/deep-learning-ai/software/rapids/

Deep Learning

(Machine Learning with Deep Models)

Cat [100%]
Dog [0%]
Bird [0%]





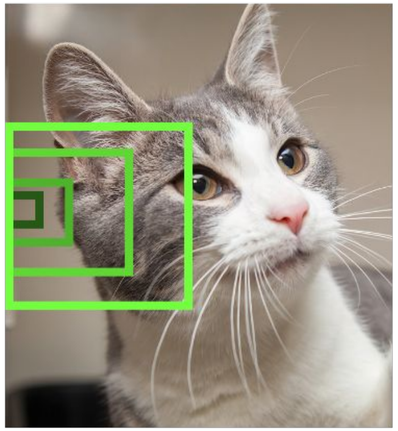
Cat [80%]
Dog [10%]
Bird [10%]

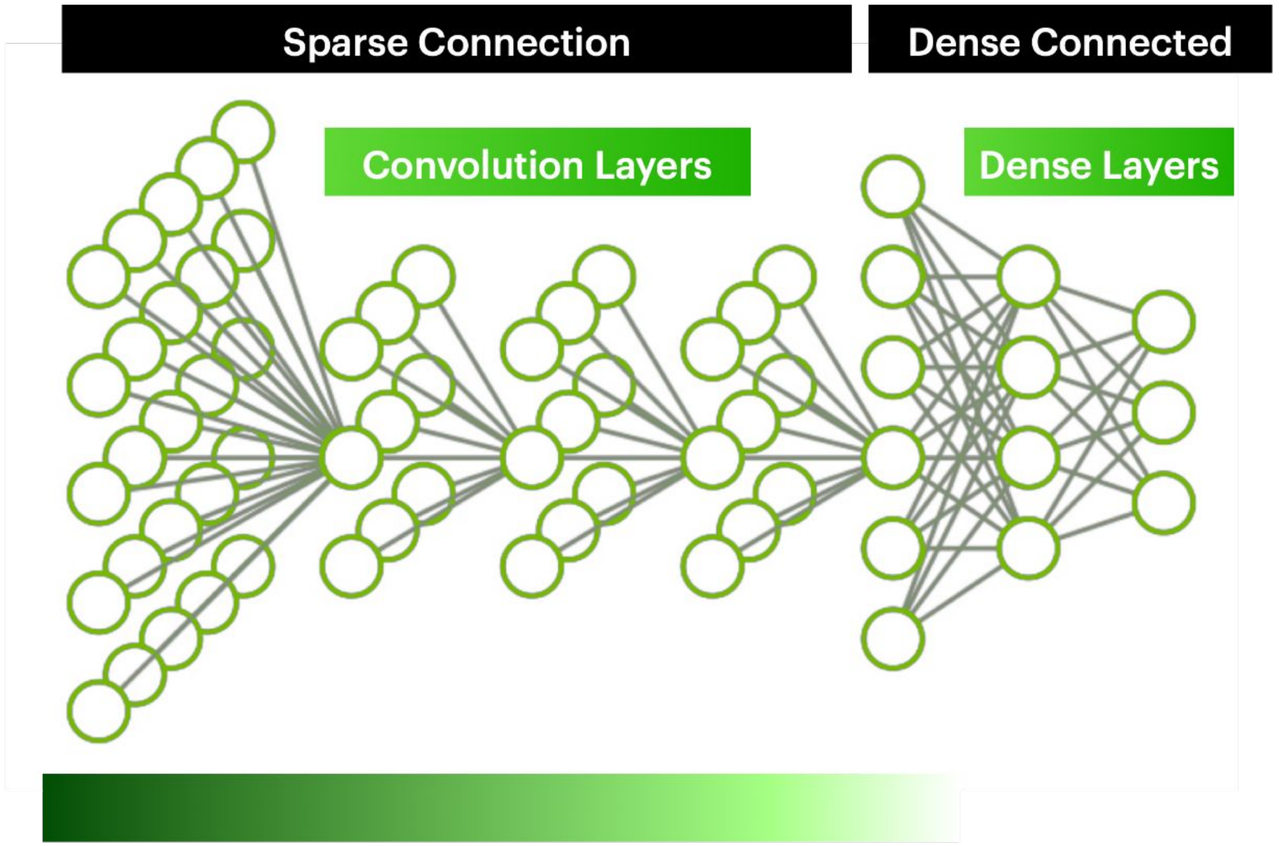
https://www.rover.com/blog/can-my-cat-eat-salt/

Deep Learning

(Machine Learning with Deep Models)

Cat [100%]
Dog [0%]
Bird [0%]





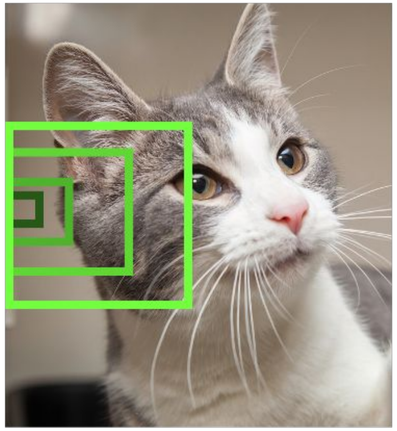
Cat [80%]
Dog [10%]
Bird [10%]

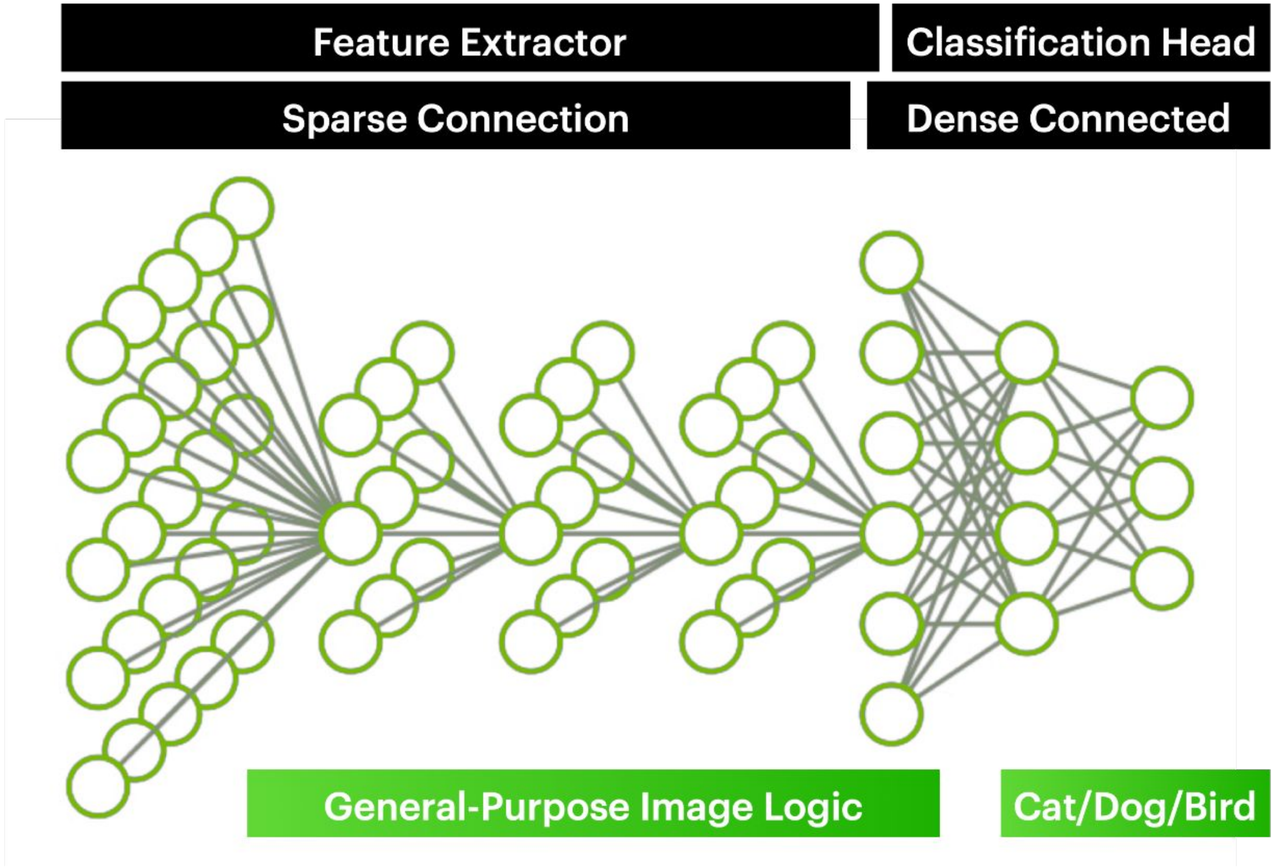
https://www.rover.com/blog/can-my-cat-eat-salt/

Deep Learning

(Machine Learning with Deep Models)

Cat [100%]
Dog [0%]
Bird [0%]





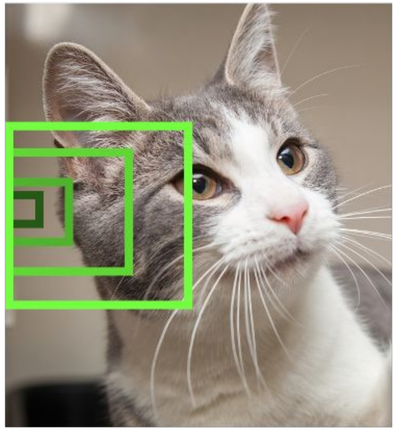
Cat [80%]
Dog [10%]
Bird [10%]

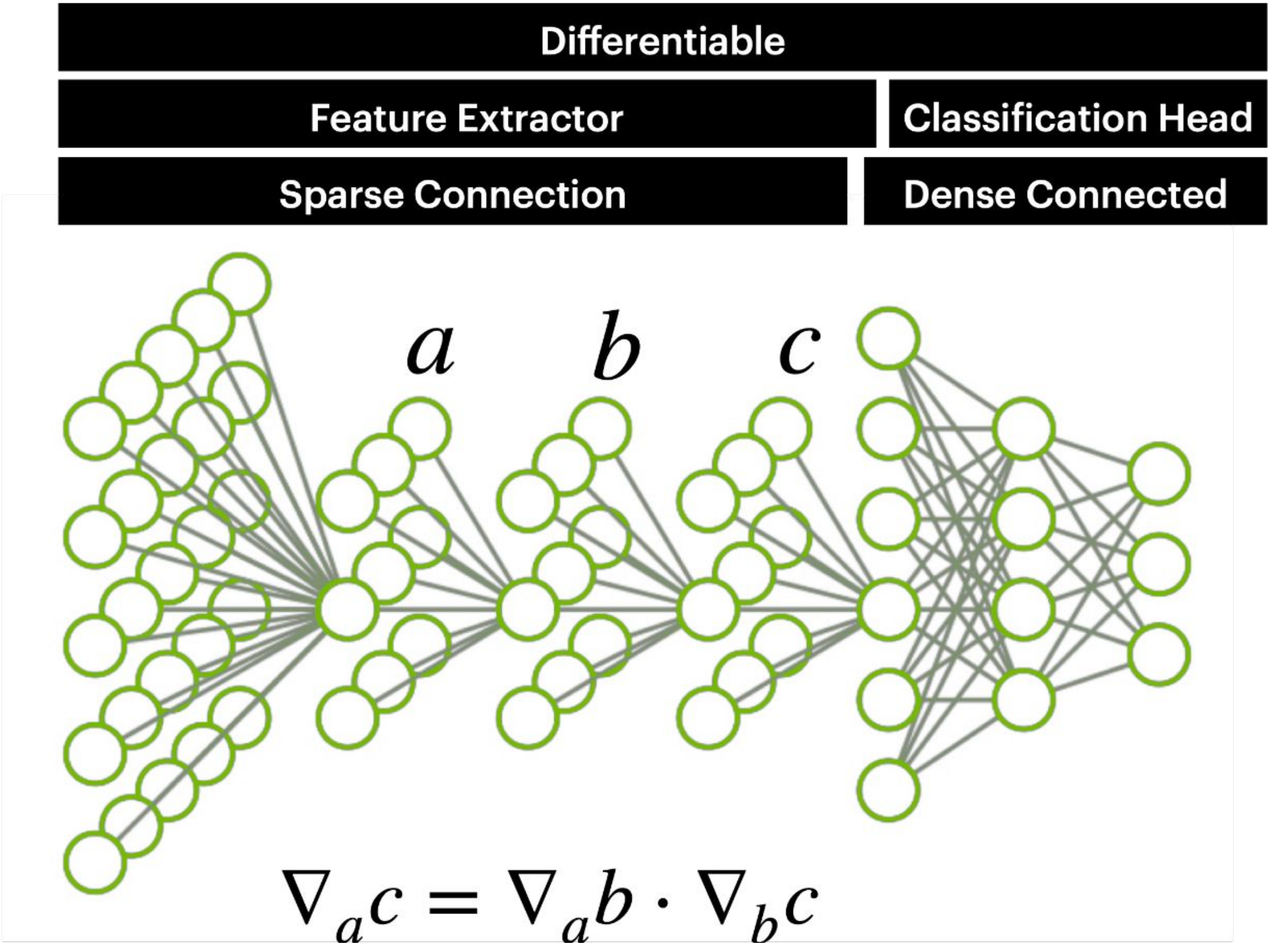
https://www.rover.com/blog/can-my-cat-eat-salt/

Deep Learning

(Machine Learning with Deep Models)

Cat [100%]
Dog [0%]
Bird [0%]



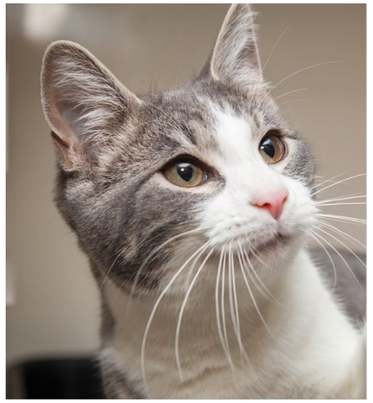


Cat [80%]
Dog [10%]
Bird [10%]

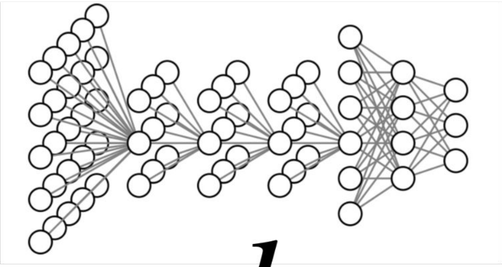
https://www.rover.com/blog/can-my-cat-eat-salt/

Deep Learning

(Machine Learning with Deep Models)



x



h_θ

$\hat{y}$

Cat	[80%]
Dog	[10%]
Bird	[10%]

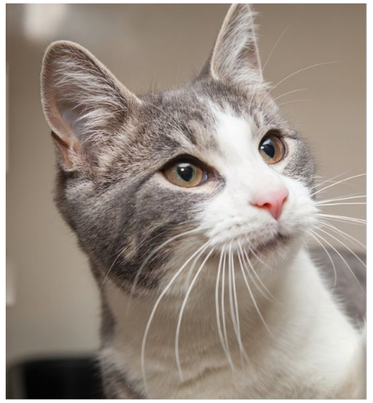
y

Cat	[100%]
Dog	[0%]
Bird	[0%]

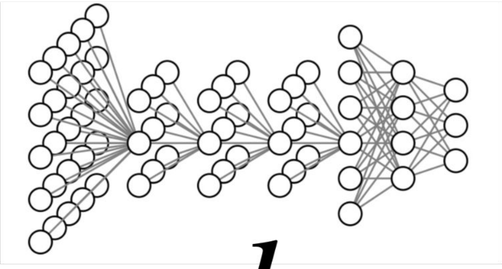
https://www.rover.com/blog/can-my-cat-eat-salt/

Deep Learning

(Machine Learning with Deep Models)



x



h_θ

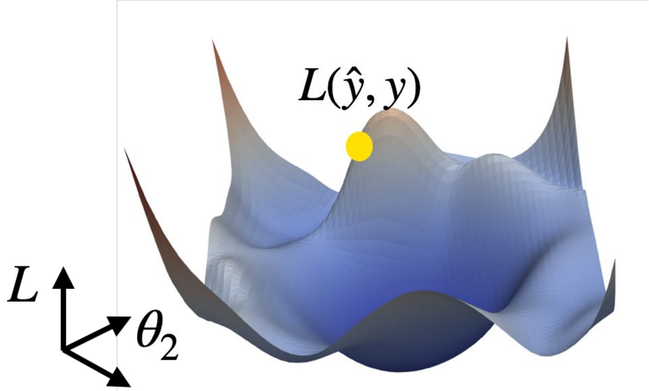
$\hat{y}$

Cat	[80%]
Dog	[10%]
Bird	[10%]

y

Cat	[100%]
Dog	[0%]
Bird	[0%]

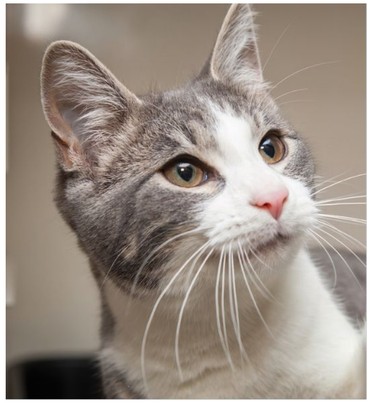
$L(\hat{y}, y)$



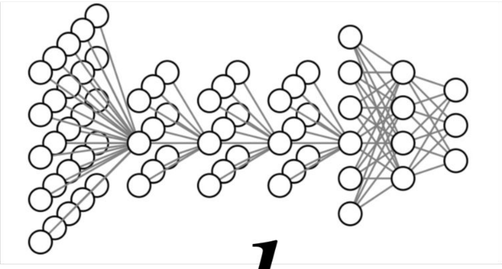
https://www.rover.com/blog/can-my-cat-eat-salt/

Deep Learning

(Machine Learning with Deep Models)



x



h_θ

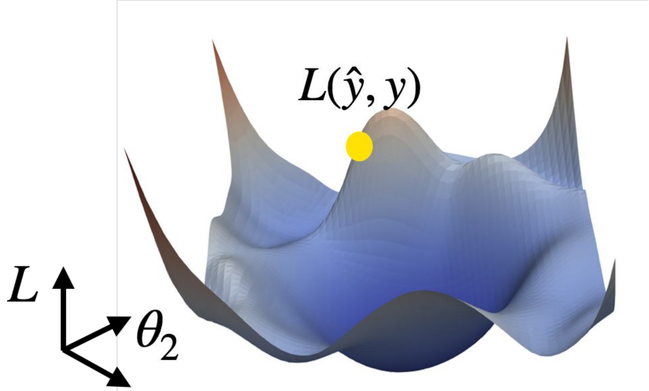
$\hat{y}$

Cat	[80%]
Dog	[10%]
Bird	[10%]

y

Cat	[100%]
Dog	[0%]
Bird	[0%]

$L(\hat{y}, y)$

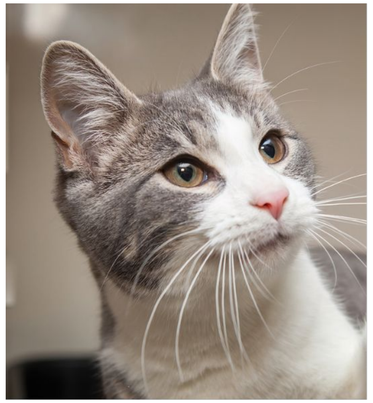


$\nabla_{\hat{y}} L$ and $\nabla_\theta \hat{y}$ are known $\implies$ $\nabla_\theta L$ is known

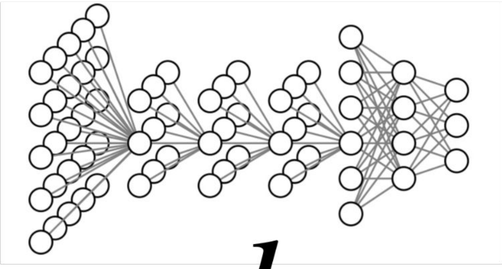
https://www.rover.com/blog/can-my-cat-eat-salt/

Deep Learning

(Machine Learning with Deep Models)



x



$h_{\theta'}$

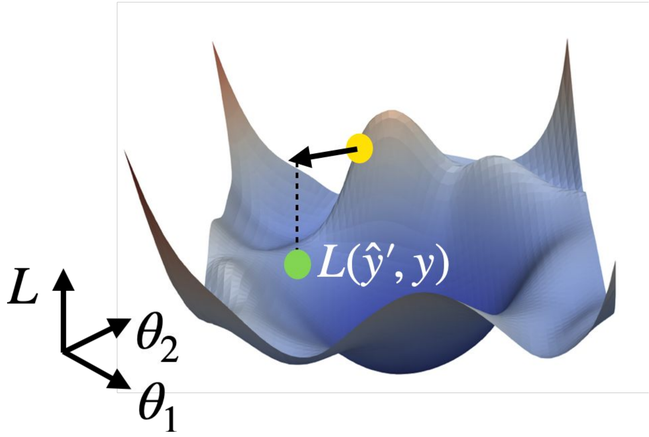
$\hat{y}'$

Cat	[90%]
Dog	[5%]
Bird	[5%]

y

Cat	[100%]
Dog	[0%]
Bird	[0%]

$L(\hat{y}', y)$



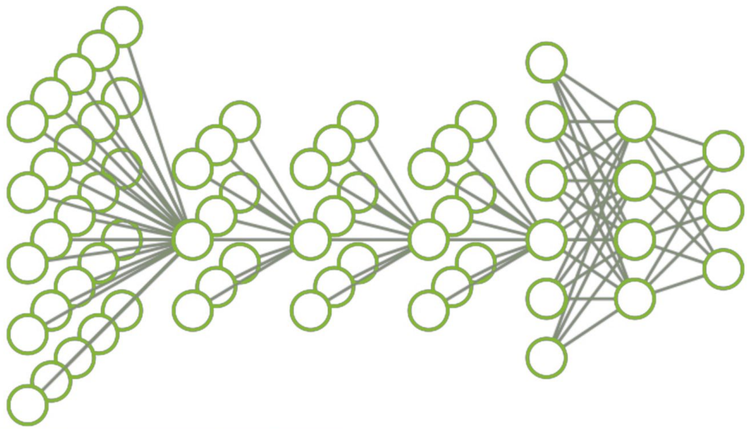
$\nabla_{\hat{y}} L$ and $\nabla_{\theta} \hat{y}$ are known $\implies$ $\nabla_{\theta} L$ is known

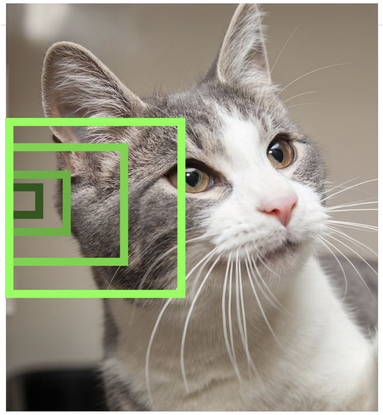
$$\theta' = \theta - \eta \nabla_{\theta} L$$

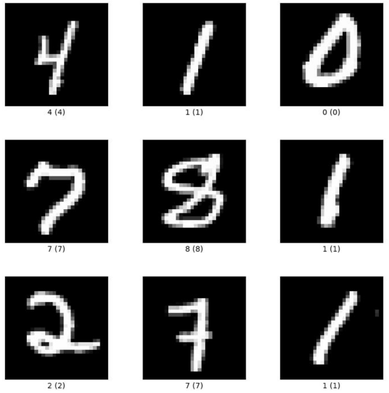
https://www.rover.com/blog/can-my-cat-eat-salt/

From Deep Learning to LLMs

Our Goal For The Course







Fundamentals of Deep Learning

A colorful cat → likely positive cat

A colorful cat → Nothing wrong with that

→ A colorful cat

A colorful cat →

Language Reasoning

What Is Language?

Discrete Labels?

Hello! How is it going?!

Sequence of Letters

Tokenization

Hello! How is it going?!

Ordered Sequence of Classes

The Base Formulation

Dealing With Language As An Input

Hello

!

How

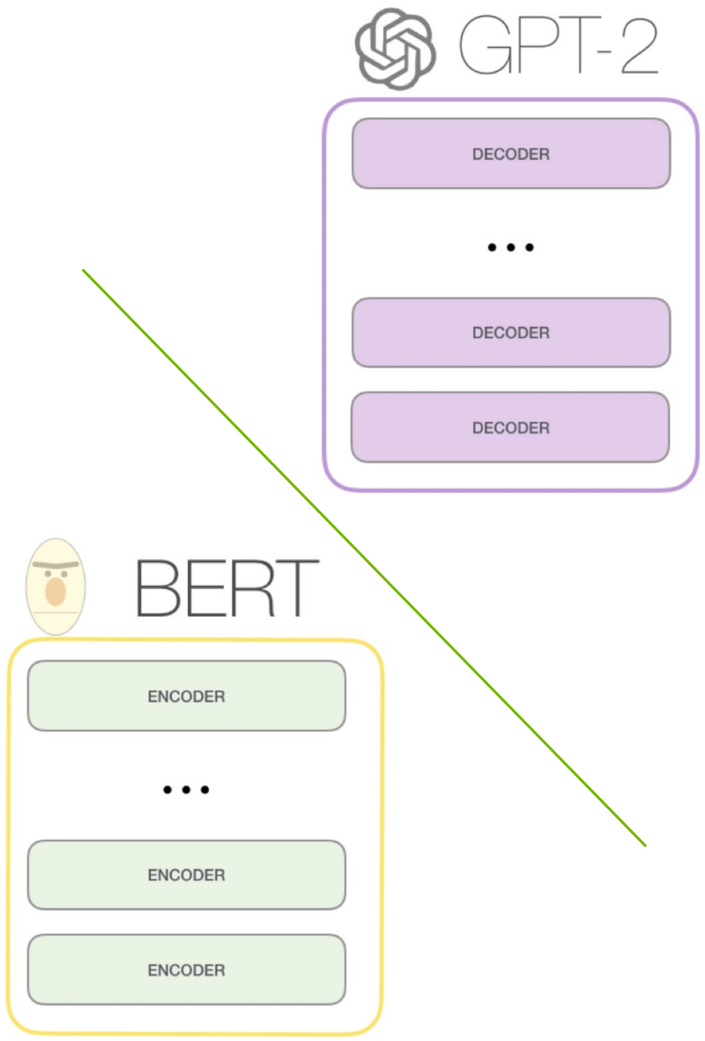
is

it

going

?

!



sequence prediction

[90% pos, 10% neg]

pre-token prediction

Hello! How is it going?!

forecasting prediction

It's going great!

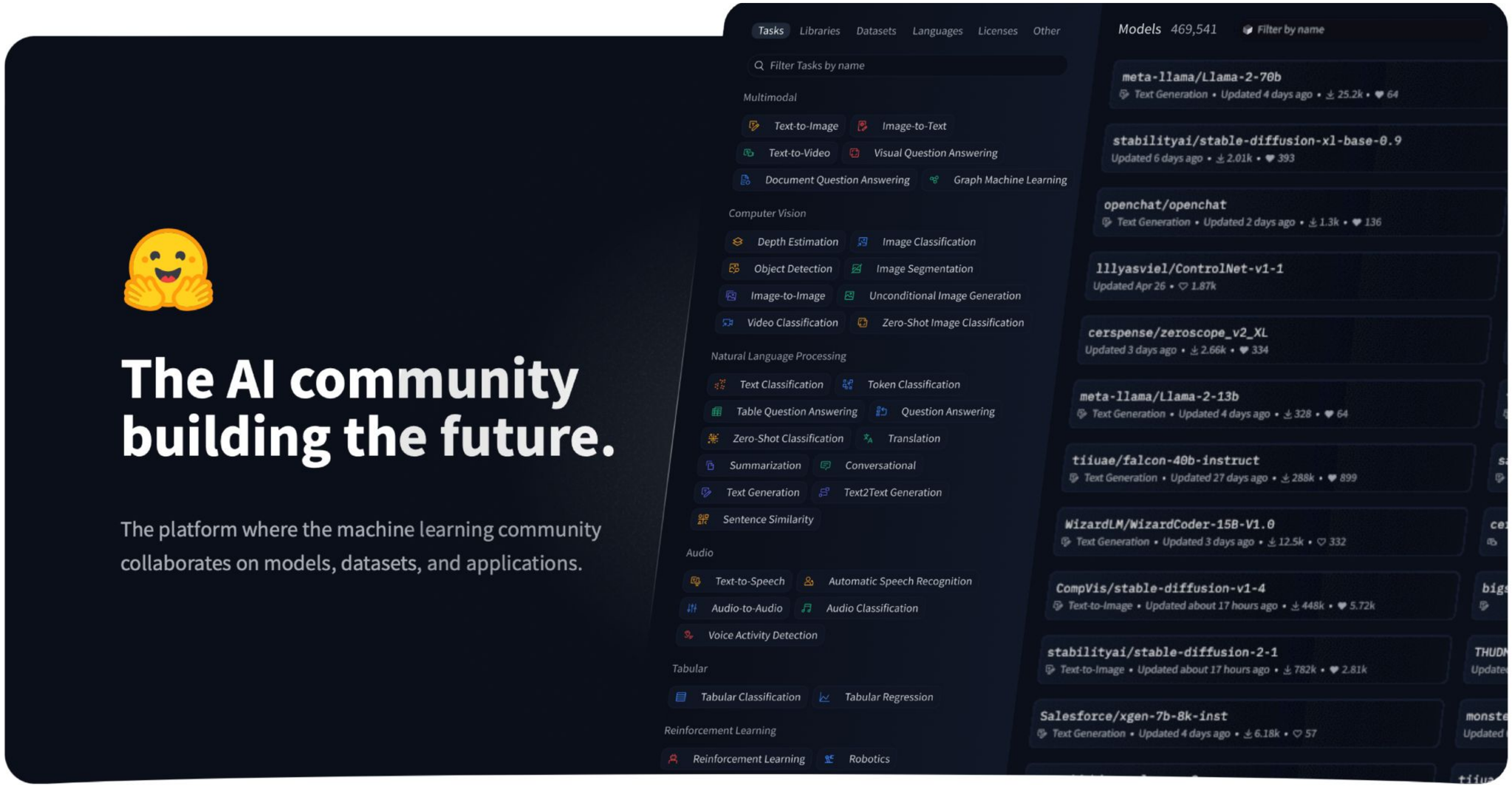
http://jalammar.github.io/illustrated-gpt2/

Part 1: From DL To LLMs

- From Deep Learning to Language Modeling
- **Pulling In Our First Model**
- Model Architecture Intuitions
- Using Encoders For Encoding

HuggingFace

Open-Source Large Model Community



HuggingFace

Open-Source Large Models

Large Open-Sourced Repository

- Very easy to use, write to, pull from
- So many models!!

Amazing For Experimentation

- See what all ***can*** be done!
- See how ***others*** have done things!
- Use their models for your use-cases ***sometimes***

Transformers

Models

huggingface.co/models

https://huggingface.co/models

HuggingFace

Open-Source Large Models

Large Open-Sourced Repository

- Very easy to use, write to, pull from
- So many models!!

Amazing For Experimentation

- See what all ***can*** be done!
- See how ***others*** have done things!
- Use their models for your use-cases ***sometimes***

Big Catch

- Check EULA licenses per-model per-dataset
- Might have to recreate system or train yourself

Transformers

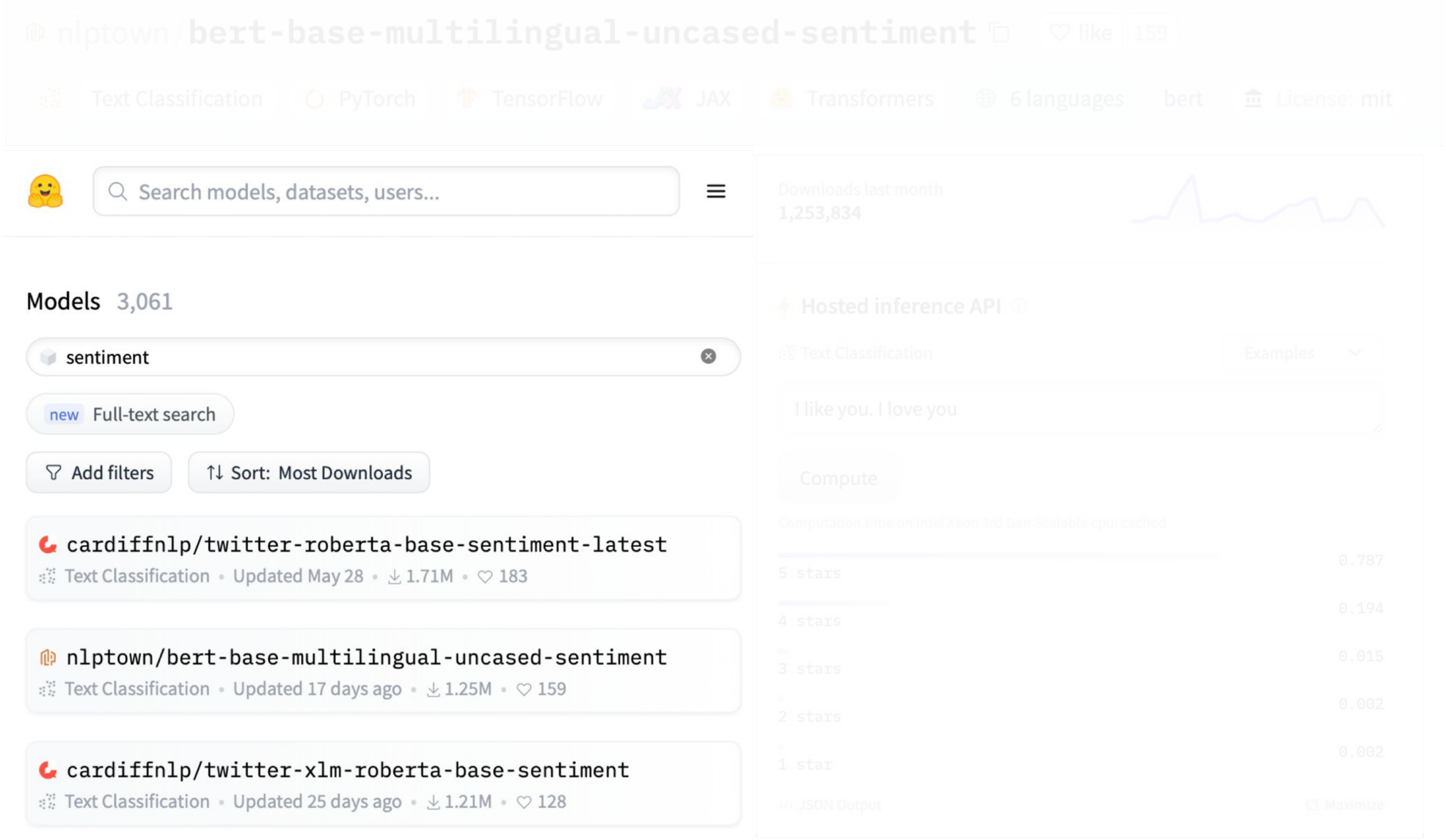
Models

huggingface.co/models

https://huggingface.co/models

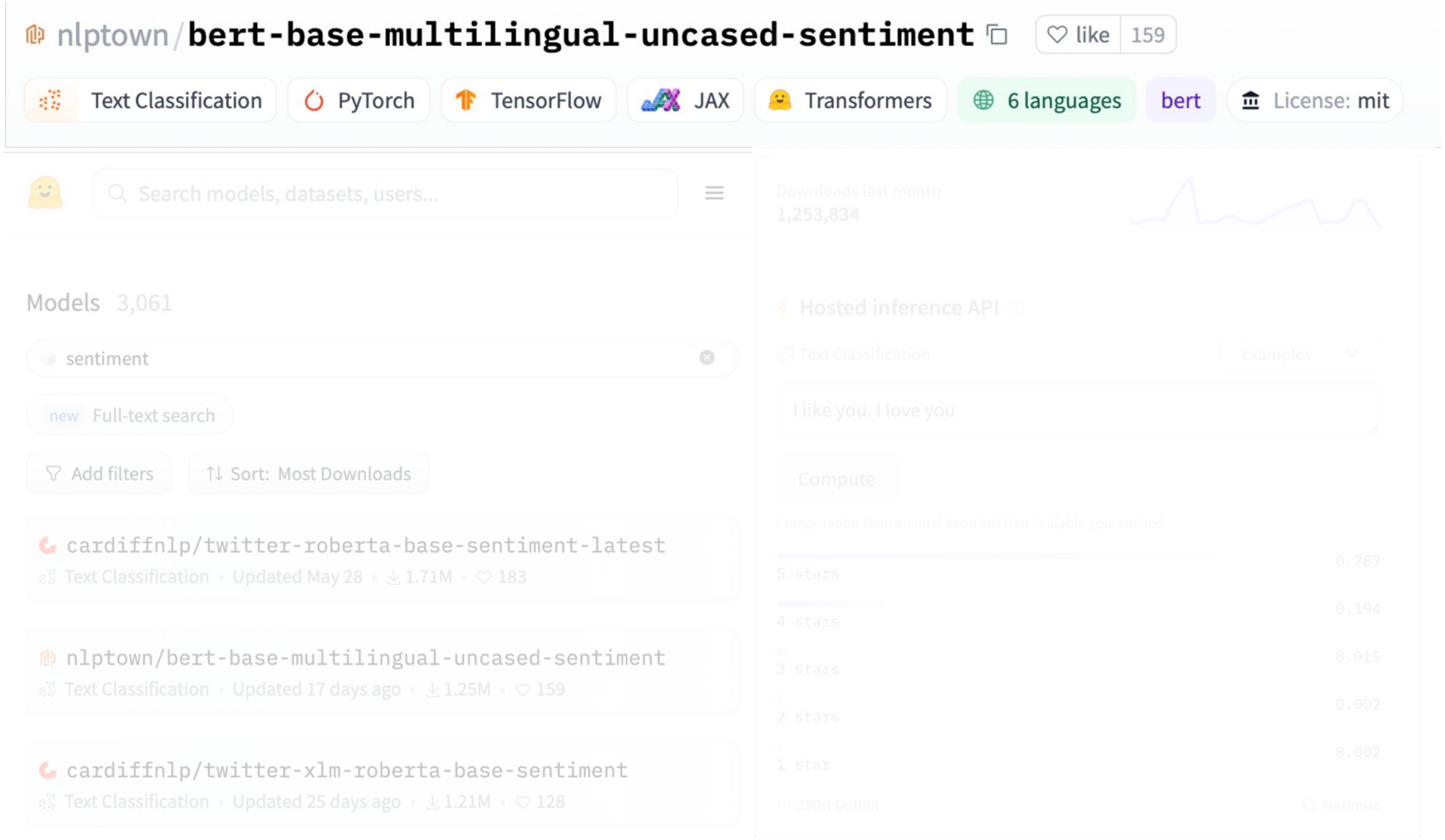
Pulling In A Simple Model

Lightweight BERT Model



Pulling In A Simple Model

Lightweight BERT Model



Pulling In A Simple Model

Lightweight BERT Model

nlptown/**bert-base-multilingual-uncased-sentiment** like 159

Text Classification | PyTorch | TensorFlow | JAX | Transformers | 6 languages | bert | License: mit

Search models, datasets, users...

Models 3,061

sentiment

new Full-text search

Add filters | Sort: Most Downloads

cardiffnlp/twitter-roberta-base-sentiment-latest
Text Classification • Updated May 28 • 1.71M • 183

nlptown/bert-base-multilingual-uncased-sentiment
Text Classification • Updated 17 days ago • 1.25M • 159

cardiffnlp/twitter-xlm-roberta-base-sentiment
Text Classification • Updated 25 days ago • 1.21M • 128

Downloads last month
1,253,834

Hosted inference API

Text Classification | Examples

I like you. I love you

Compute

Computation time on Intel Xeon 3rd Gen Scalable cpu: cached

5 stars	0.787
4 stars	0.194
3 stars	0.015
2 stars	0.002
1 star	0.002

JSON Output | Maximize

Pulling In A Simple Model

Lightweight BERT Model

How to use from the 🤗/transformers library

Use in Transformers

Copy

```python
# Use a pipeline as a high-level helper
from transformers import pipeline

pipe = pipeline("text-classification", model="nlptown/bert-base-multilingual-uncased-sentiment")
```

Copy

```python
# Load model directly
from transformers import AutoTokenizer, AutoModelForSequenceClassification

tokenizer = AutoTokenizer.from_pretrained("nlptown/bert-base-multilingual-uncased-sentiment")
model = AutoModelForSequenceClassification.from_pretrained("nlptown/bert-base-multilingual-uncased-se
```

Quick Links

- Read model documentation
- Read docs on high-level-pipeline
- Read our learning resources

Pulling In A Simple Model

Lightweight BERT Model

```
# !pip install -q transformers

from transformers import pipeline
pipe = pipeline("text-classification", model="nlptown/bert-base-multilingual-uncased-sentiment")
```

```
Downloading (…)lve/main/config.json: 100%  953/953 [00:00<00:00, 20.8kB/s]
Downloading pytorch_model.bin: 100%  669M/669M [00:07<00:00, 101MB/s]
Downloading (…)okenizer_config.json: 100%  39.0/39.0 [00:00<00:00, 943B/s]
Downloading (…)solve/main/vocab.txt: 100%  872k/872k [00:00<00:00, 1.25MB/s]
Downloading (…)cial_tokens_map.json: 100%  112/112 [00:00<00:00, 1.41kB/s]
```

Running Your Simple Model

Lightweight BERT Model

```
pipe("hello")    = [{'label': '5 stars', 'score': 0.5614452362060547}]
pipe("world")    = [{'label': '5 stars', 'score': 0.4746573567390442}]
pipe("awesome")  = [{'label': '5 stars', 'score': 0.830653965473175}]
pipe("happy")    = [{'label': '5 stars', 'score': 0.5925828814506531}]

pipe("ay")       = [{'label': '5 stars', 'score': 0.4219609498977661}]
pipe("yay")      = [{'label': '3 stars', 'score': 0.3043082058429718}]
pipe("uh")       = [{'label': '1 star', 'score': 0.3023006021976471}]
pipe("eh")       = [{'label': '2 stars', 'score': 0.3670440018177032 5}]

pipe("fluffy")   = [{'label': '1 star', 'score': 0.4247331917285919}]
pipe("cuddly")   = [{'label': '4 stars', 'score': 0.2446891665458679 2}]
pipe("comfy")    = [{'label': '5 stars', 'score': 0.2954443991184234 6}]
```

Running Your Simple Model

Lightweight BERT Model

```
pipe("I loved my old pillow more than anything! Definitely more than this one.")
```

```
[{'label': '5 stars', 'score': 0.9359365105628967}]
```

```
pipe("This new pillow is like a rock; just as comfortable and just as useful.")
```

```
[{'label': '5 stars', 'score': 0.835628867149353}]
```

```
pipe("I hated my old pillow so much, but this one is nothing like it!")
```

```
[{'label': '1 star', 'score': 0.5815897583961487}]
```

Pipeline Components

Input -> Tensor -> Tensor -> Output

preprocess

Tokenize the data and convert to tensors

Arrange into dictionary for forward

AutoTokenizer

forward

Pass tensor dict into Model.forward

Arrange into dictionary for postprocess

AutoModelFor<TASK>

postprocess

Take all of the data from **AutoModelFor<TASK>**

Convert to intuitive output per task specs

<TASK>Pipeline

Pipeline

It's so [MASK]! → It's so fun!

Pipeline Components

Input -> Tensor -> Tensor -> Output

```
msg = "It's so fluffy I'm gonna die"
prep_fn  = pipe.preprocess
model_fn = pipe.model  # = pipe.model.forward
post_fn  = pipe.postprocess

v1 = prep_fn(msg)
v2 = model_fn(**v1)
v3 = post_fn({k:v.detach() for k,v in v2.items()})
```

```
v1:
 > input_ids: tensor([[  101, 10197,   112,   161, 10297, 19341, 23877, 26356,   151,   112,
           155, 47016, 10121,   102]])
 > token_type_ids: tensor([[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]])
 > attention_mask: tensor([[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]])

v2:
 > logits: tensor([[ 2.2736,  2.0283,  0.7437, -1.8262, -2.5241]],
       grad_fn=<AddmmBackward0>)

v3:
 > label: '1 star'
 > score: 0.4940895438194275
```

Pipeline Components

Input -> Tensor -> Tensor -> Output

Tied To The Model

String → Dict → Dict → String

preprocess

It's so fluffy!

```
enc = tokenizer.encode(msg)
```

[101, 10197, 112, 161, ... , 106, 102]

```
dec = tokenizer.decode(enc)
```

[CLS] It's so fluffy! [SEP]

Tied To The Tokenizer

postprocess

[2.274, 2.028, 0.744, -1.826, -2.524]

```
probs = torch.exp of logits
```

[9.714, 7.601, 2.104, 0.161, 0.080]

```
score = max(probs)/sum(probs)
```

0.4941

```
label = id2label[argmax(probs)]
```

1 star

Tied To Pipeline

Pipeline Components

Input -> Tensor -> Tensor -> Output

preprocess

Tokenize the data and convert to tensors

Arrange into dictionary for forward

AutoTokenizer

forward

Pass tensor dict into Model.forward

Arrange into dictionary for postprocess

AutoModelFor<TASK>

postprocess

Take all of the data from **AutoModelFor<TASK>**

Convert to intuitive output per task specs

<TASK>Pipeline

It's so [MASK]! → **Pipeline** → It's so fun!

Pipeline Components

Input -> Tensor -> Tensor -> Output

preprocess

Tokenize the data and convert to tensors

Arrange into dictionary for forward

AutoTokenizer

system HF STAFF Update config.json aa631bb

raw history blame contribute delete No virus

```
28    "layer_norm_eps": 1e-12,
29    "max_position_embeddings": 512,
30    "model_type": "bert",
31    "num_attention_heads": 12,
32    "num_hidden_layers": 12,
33    "output_past": true,
34    "pad_token_id": 0,
35    "pooler_fc_size": 768,
36    "pooler_num_attention_heads": 12,
37    "pooler_num_fc_layers": 3,
38    "pooler_size_per_head": 128,
39    "pooler_type": "first_token_transform",
40    "type_vocab_size": 2,
41    "vocab_size": 105879
42  }
```

```
TOKENIZER_MAPPING_NAMES = OrderedDict(
    [
        (
            "albert",
            (
                "AlbertTokenizer" if is_sentencepiece_a
                "AlbertTokenizerFast" if is_tokenizers_
            ),
        ),
        ("align", ("BertTokenizer", "BertTokenizerFast"
        ("bark", ("BertTokenizer", "BertTokenizerFast"
        ("bart", ("BartTokenizer", "BartTokenizerFast")
        (
            "barthez",
            (
                "BarthezTokenizer" if is_sentencepiece_
                "BarthezTokenizerFast" if is_tokenizers
            ),
        ),
        ("bartpho", ("BartphoTokenizer", None)),
        ("bert", ("BertTokenizer", "BertTokenizerFast"
        ("bert-generation", ("BertGenerationTokenizer"
```

Pipeline

It’s so [MASK]! → It’s so fun!

Pipeline Components

Input -> **Tensor -> Tensor** -> Output

forward

Pass tensor dict into Model.forward

Arrange into dictionary for postprocess

AutoModelFor<TASK>

system HF STAFF Update config.json aa631bb

raw history blame contribute delete


```
{
  "_num_labels": 5,
  "architectures": [
    "BertForSequenceClassification"
  ],
  "attention_probs_dropout_prob": 0.1,
  "directionality": "bidi",
  "finetuning_task": "sentiment-analysis",
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 768,
```


It's so [MASK]! —**Pipeline**→ It's so fun!

Pipeline Components

Input -> Tensor -> **Tensor -> Output**

postprocess

Take all of the data from **AutoModelFor<TASK>**

Convert to intuitive output per task specs

<TASK>Pipeline

system HF STAFF Update config.json aa631bb

raw history blame contribute delete

```
6   "attention_probs_dropout_prob": 0.1,
7   "directionality": "bidi",
8   "finetuning_task": "sentiment-analysis",
9   "hidden_act": "gelu",
```

```
TASK_ALIASES = {
    "sentiment-analysis": "text-classification",
```

```
"text-classification": {
    "impl": TextClassificationPipeline,
    "tf": (TFAutoModelForSequenceClassification,) if is_tf_available() else (),
    "pt": (AutoModelForSequenceClassification,) if is_torch_available() else (),
    "default": {
        "model": {
            "pt": ("distilbert-base-uncased-finetuned-sst-2-english", "af0f99b"),
```

It’s so [MASK]! —**Pipeline**→ It’s so fun!

HF Pipeline Options

Automatic End-to-End Pipelines

HuggingFace Automatic Task Support

- Based on a given task:
  - Pulls in default models, tokenizers, etc.
  - Pulls in default Pipeline/postprocess.
  - Offers Backend Options (“pt”, “tf”, etc.)
  - **Recommends Default Model**

```
TASK_ALIASES = {
    "sentiment-analysis": "text-classification",
    "ner": "token-classification",
    "vqa": "visual-question-answering",
}
SUPPORTED_TASKS = {
    "audio-classification": {…
    "automatic-speech-recognition": {…

    ## distilbert-base-cased:935ac13
    "feature-extraction": {
        "impl": FeatureExtractionPipeline,
        "tf": (TFAutoModel,) if is_tf_available()
        "pt": (AutoModel,) if is_torch_available()
        "default": {"model": {
            "pt": ("distilbert-base-cased", "935ac
            "tf": ("distilbert-base-cased", "935ac
        }}, "type": "multimodal",
    },

    ## distilbert-base-uncased-finetuned-sst-2-eng
    "text-classification": {…
    ## dbmdz/bert-large-cased-finetuned-conll03-en
    "token-classification": {…
    ## distilbert-base-cased-distilled-squad:626af
    "question-answering": {…
```

HF Pipeline Options

Automatic End-to-End Pipelines

NLP Classification

feature-extraction
text-classification
token-classification
fill-mask

NLP Response

question-answering
summarization
translation
text2text-generation
zero-shot-classification

table-qa
document-qa

NLP Generative

text-generation
conversational

Audio

audio-classification
automatic-speech-recognition
text-to-audio
zero-shot-audio-classification

Vision

visual-question-answering
image-classification
image-segmentation
image-to-text
object-detection
depth-estimation
video-classification
mask-generation
zero-shot-image-classification
zero-shot-object-detection

Part 1: From DL To LLMs

- From Deep Learning to Language Modeling
- Pulling In Our First Model
- **Model Architecture Intuitions**
- Using Encoders For Encoding

Our Problem Space

Ordered Sequence of Classes

Hello! How is it going?!

Sequence of Letters

Tokenization

Hello! How is it going?!

Ordered Sequence of Classes

The Base Formulation

Dealing With Language As An Input

Hello

!

How

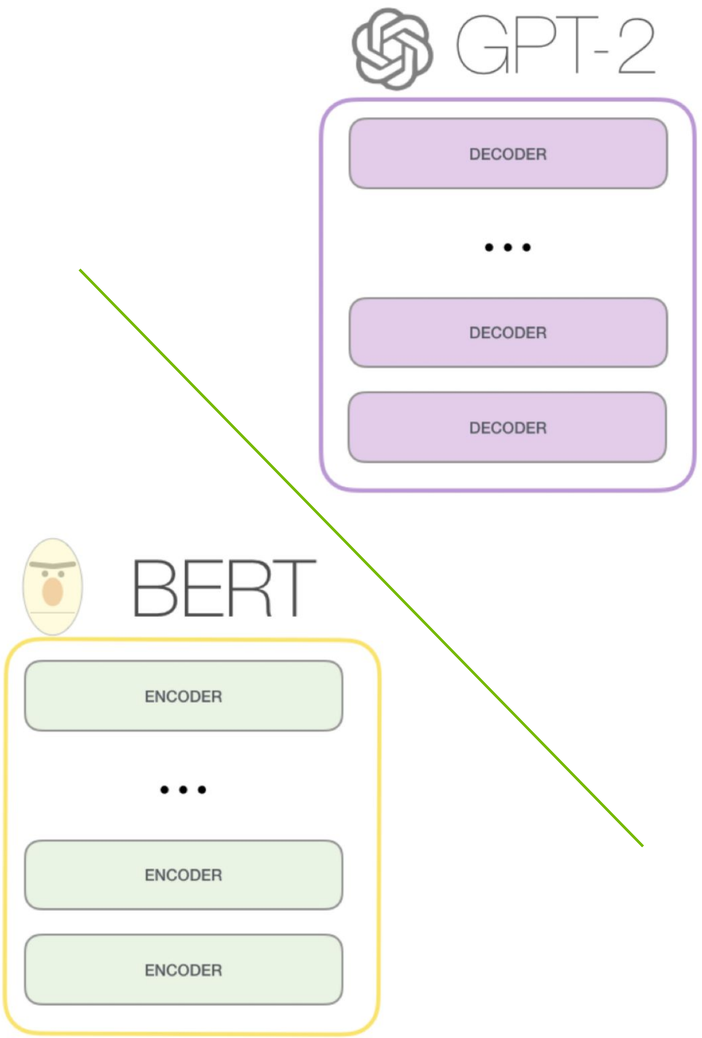
is

it

going

?

!



Sequence-Level Prediction
[90% pos, 10% neg]

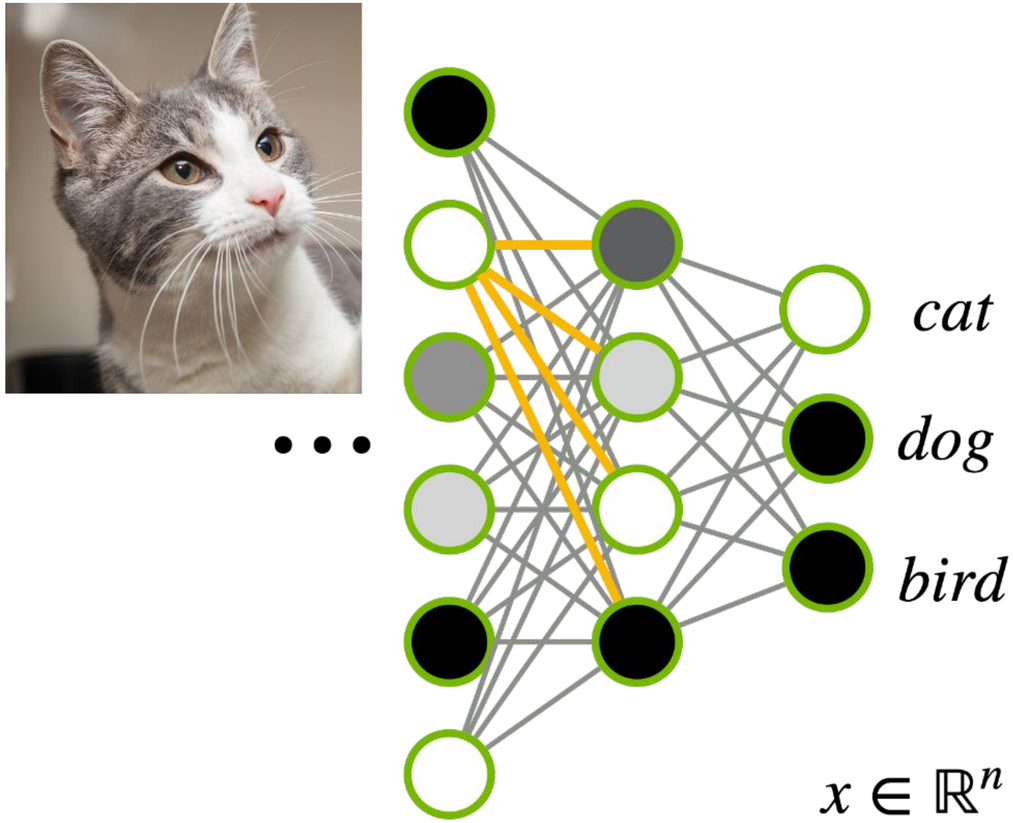
Per-Token Prediction
Hello! How is it going?!

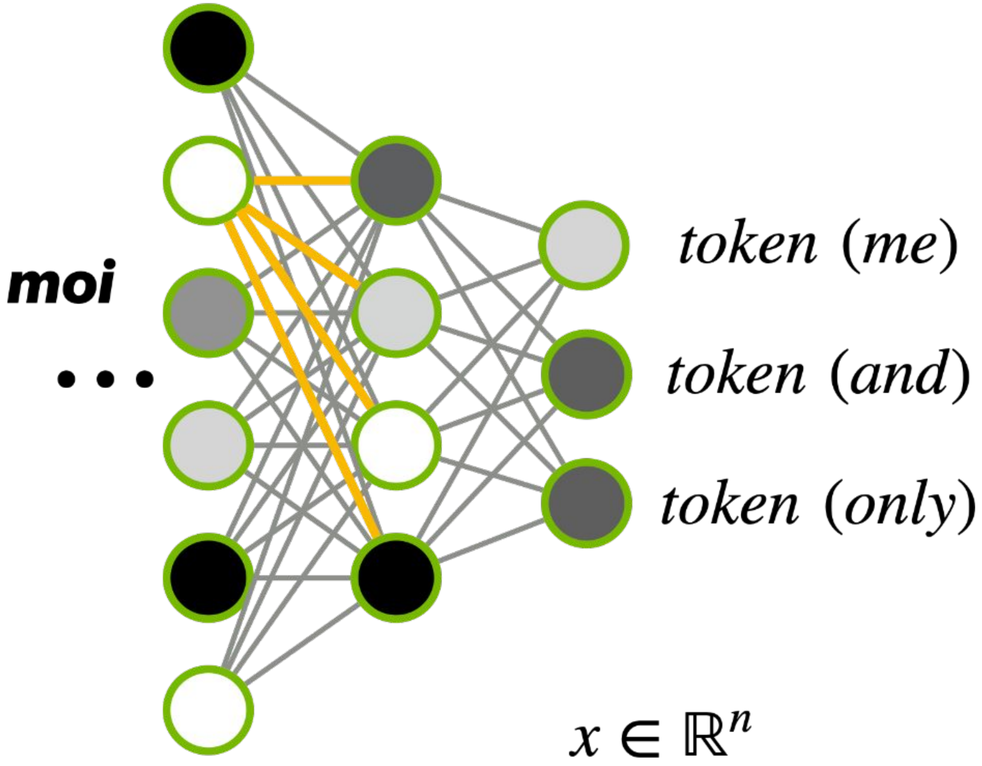
Generation (Forecasting)
It's going great!

http://jalammar.github.io/illustrated-gpt2/

Dealing With Classes

Ordered Sequence of **Classes**

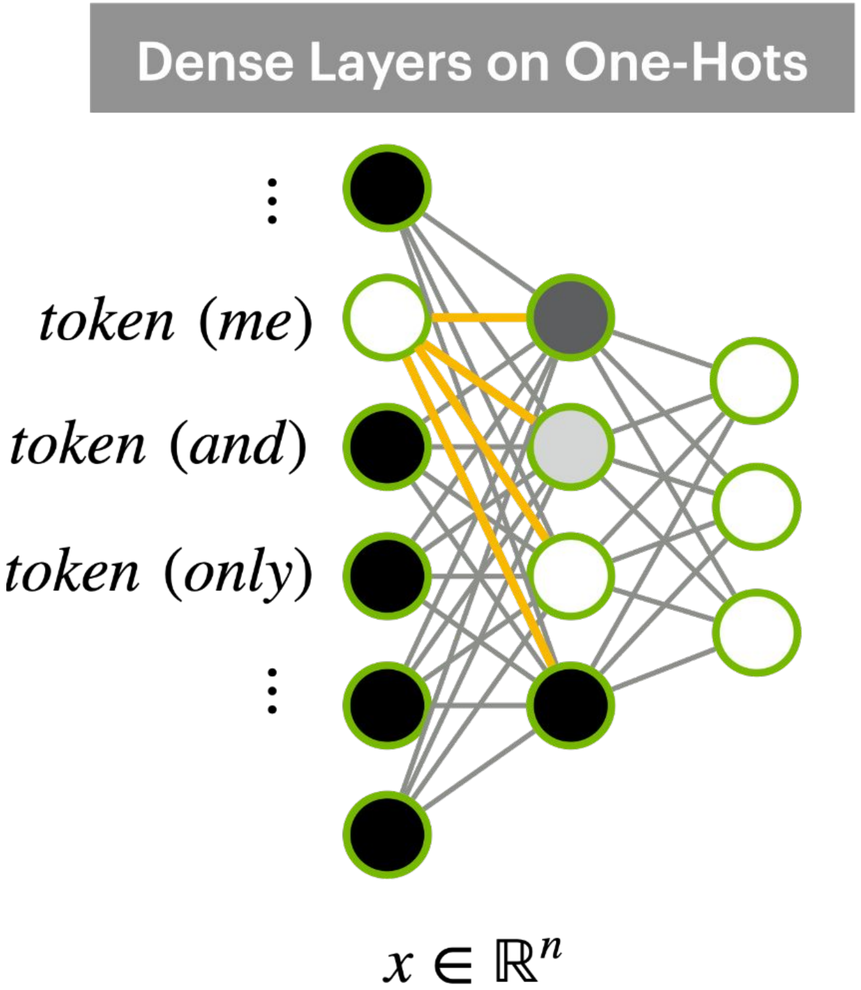




https://www.rover.com/blog/can-my-cat-eat-salt/

Dealing With Classes

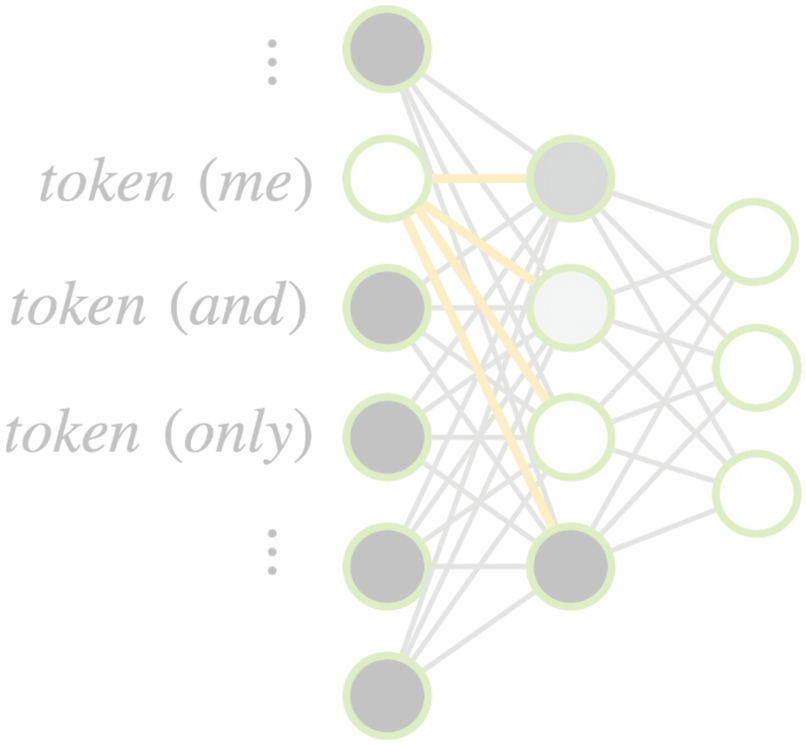
Ordered Sequence of **Classes**



Dealing With Classes

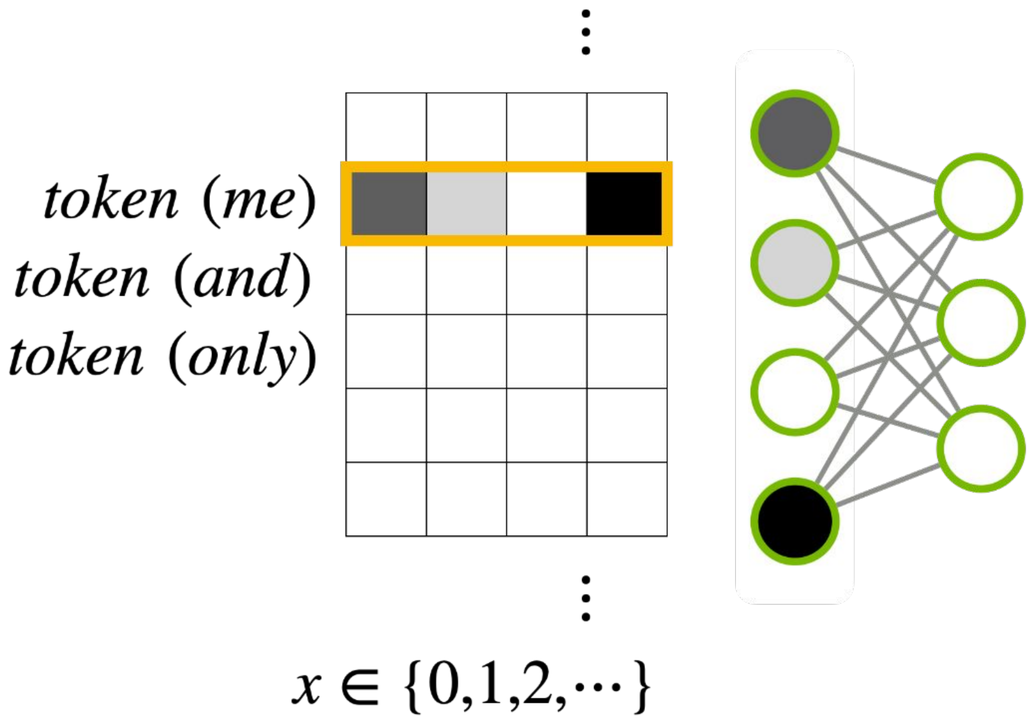
Ordered Sequence of **Classes**

Dense Layers on One-Hots



$x \in \mathbb{R}^n$

Embedding Layers



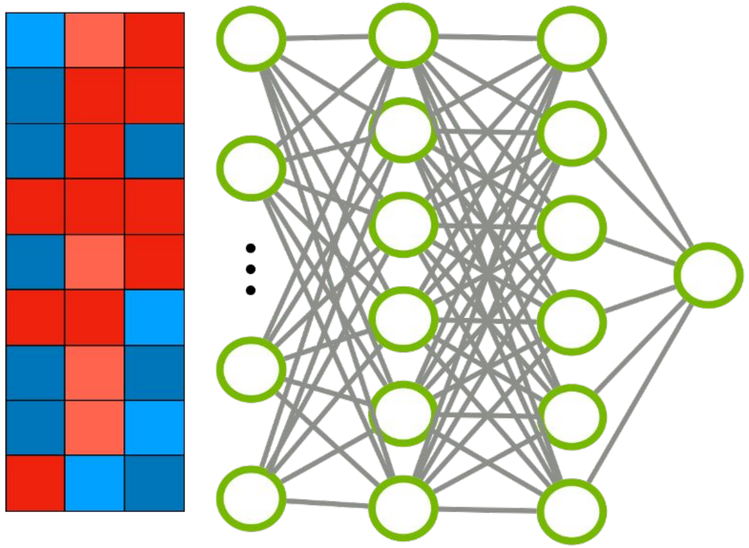
$x \in \{0,1,2,\cdots\}$

$\text{Embed}[x] \in \mathbb{R}^k$

Dealing With Classes

Ordered Sequence of **Classes**

Reason About It
All At Once

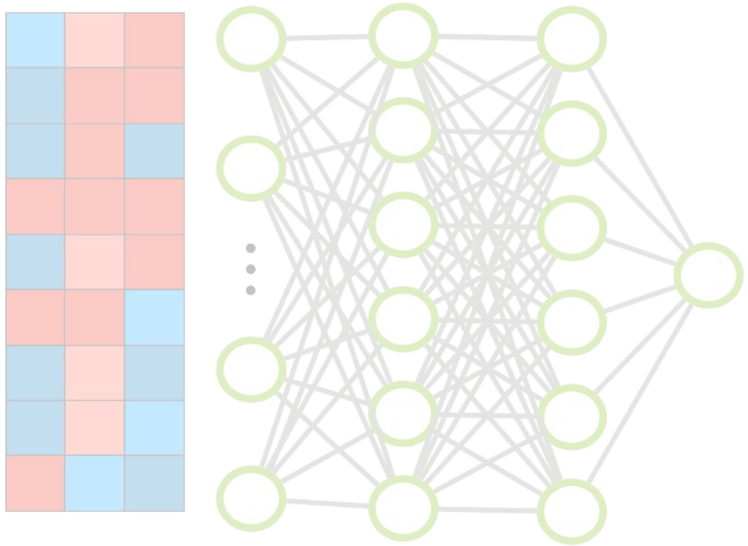


Fully-Connected Neural Network

Dealing With Classes

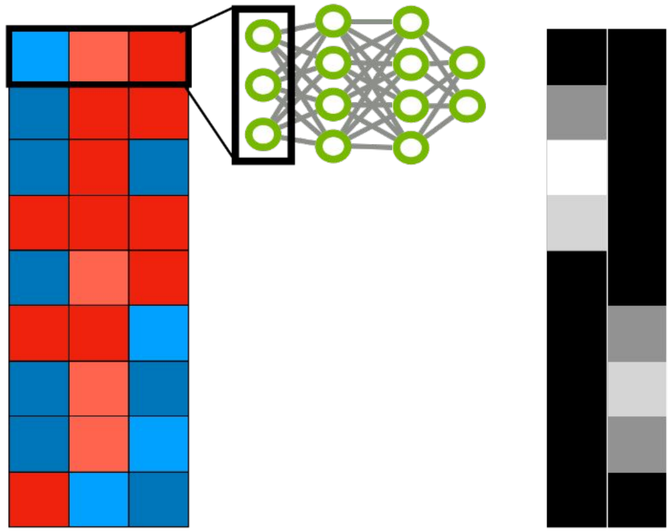
Ordered Sequence of **Classes**

Reason About It
All At Once



Fully-Connected
Neural Network

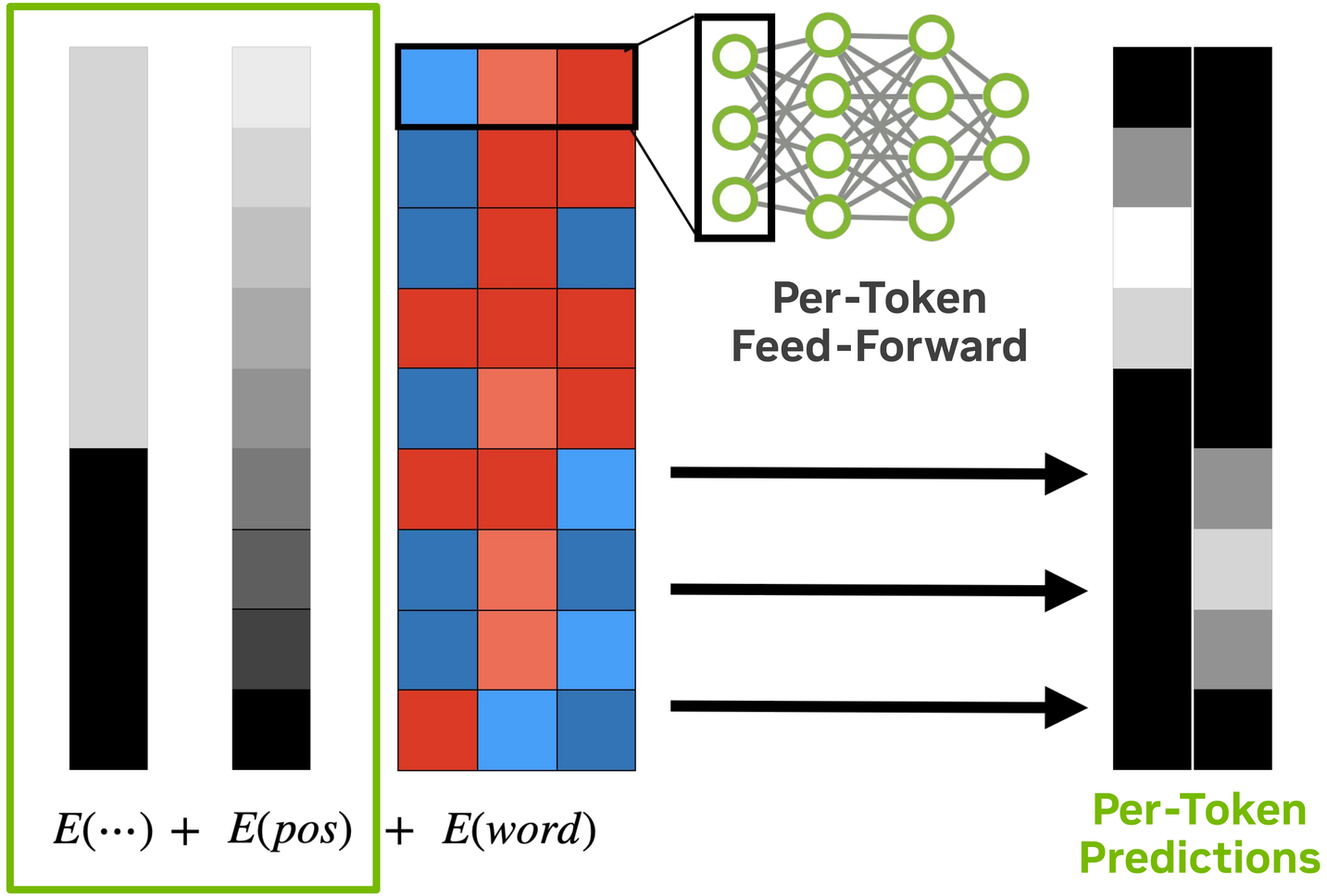
Reason One Sequence
Entry At A Time



Position-Wise
Feed-Forward NN

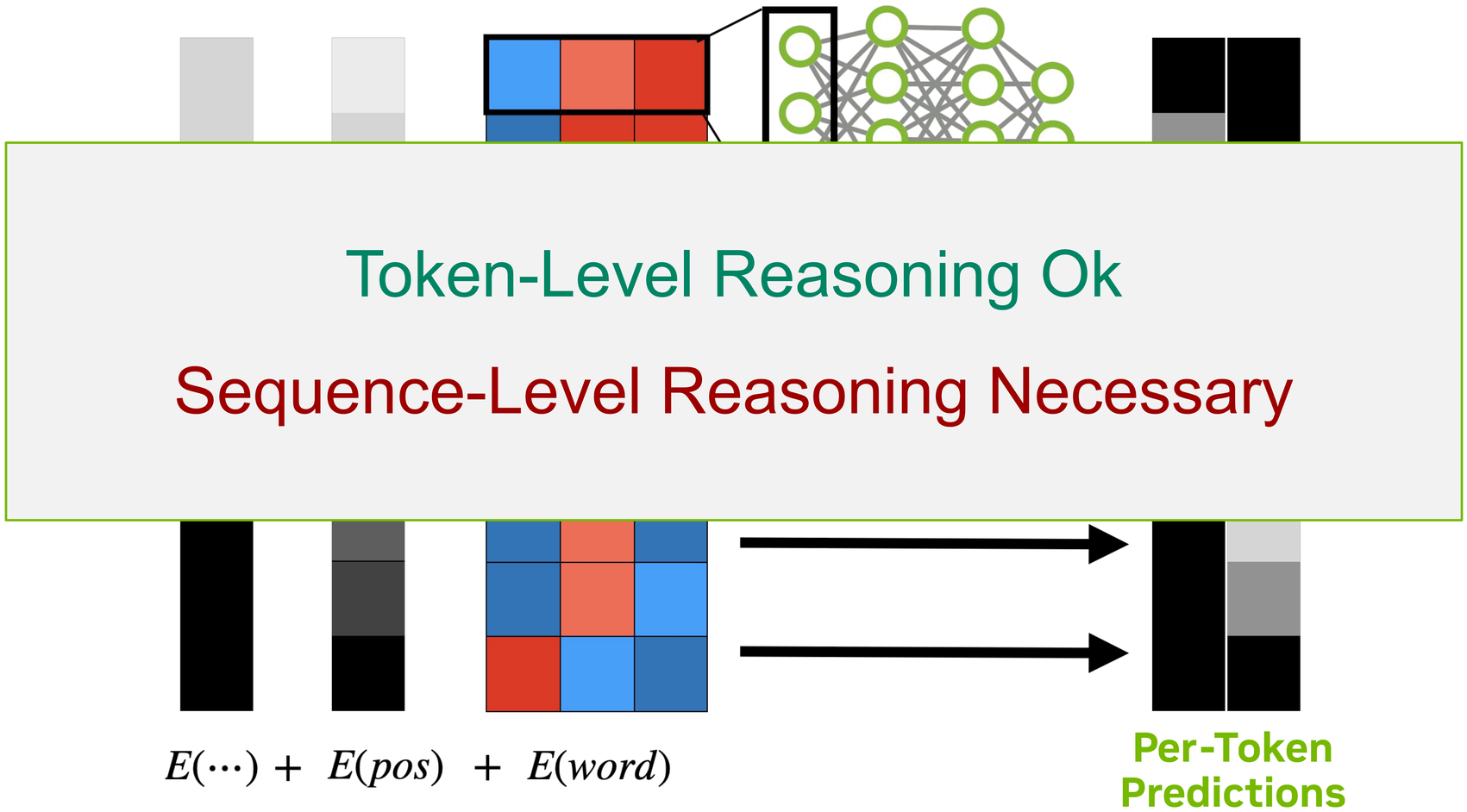
Dealing With Ordered Classes

Ordered Sequence of **Classes**



Dealing With Ordered Classes

Ordered Sequence of **Classes**

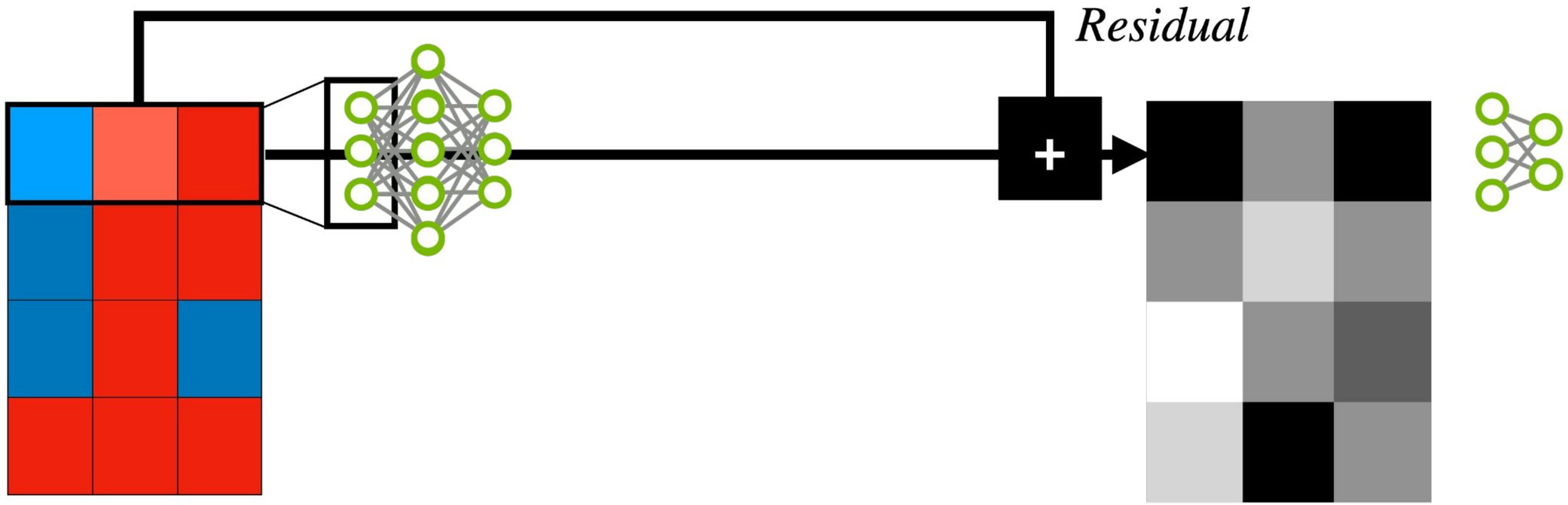


Transformers for Sequence Reasonings

Ordered **Sequence** of Classes

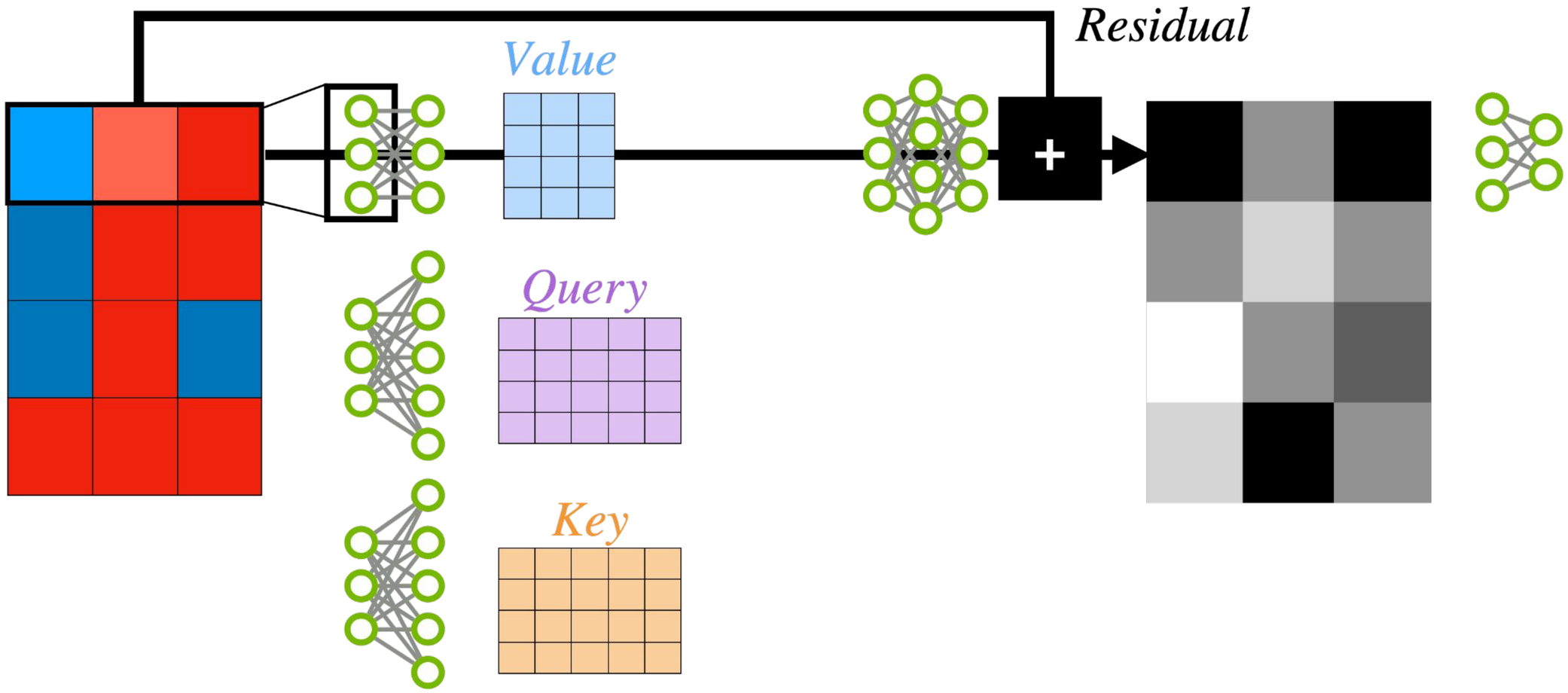
Transformers for Sequence Reasonings

Ordered **Sequence** of Classes



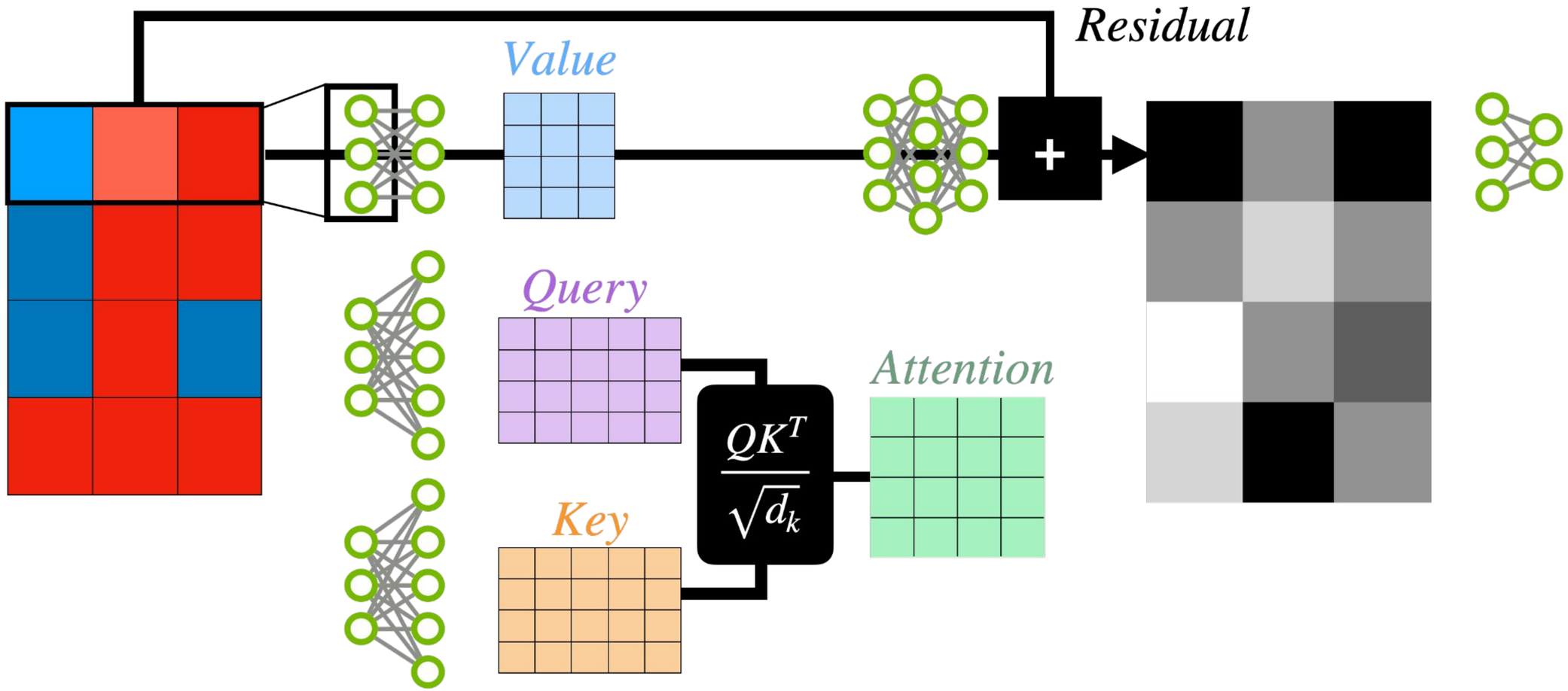
Transformers for Sequence Reasonings

Ordered **Sequence** of Classes



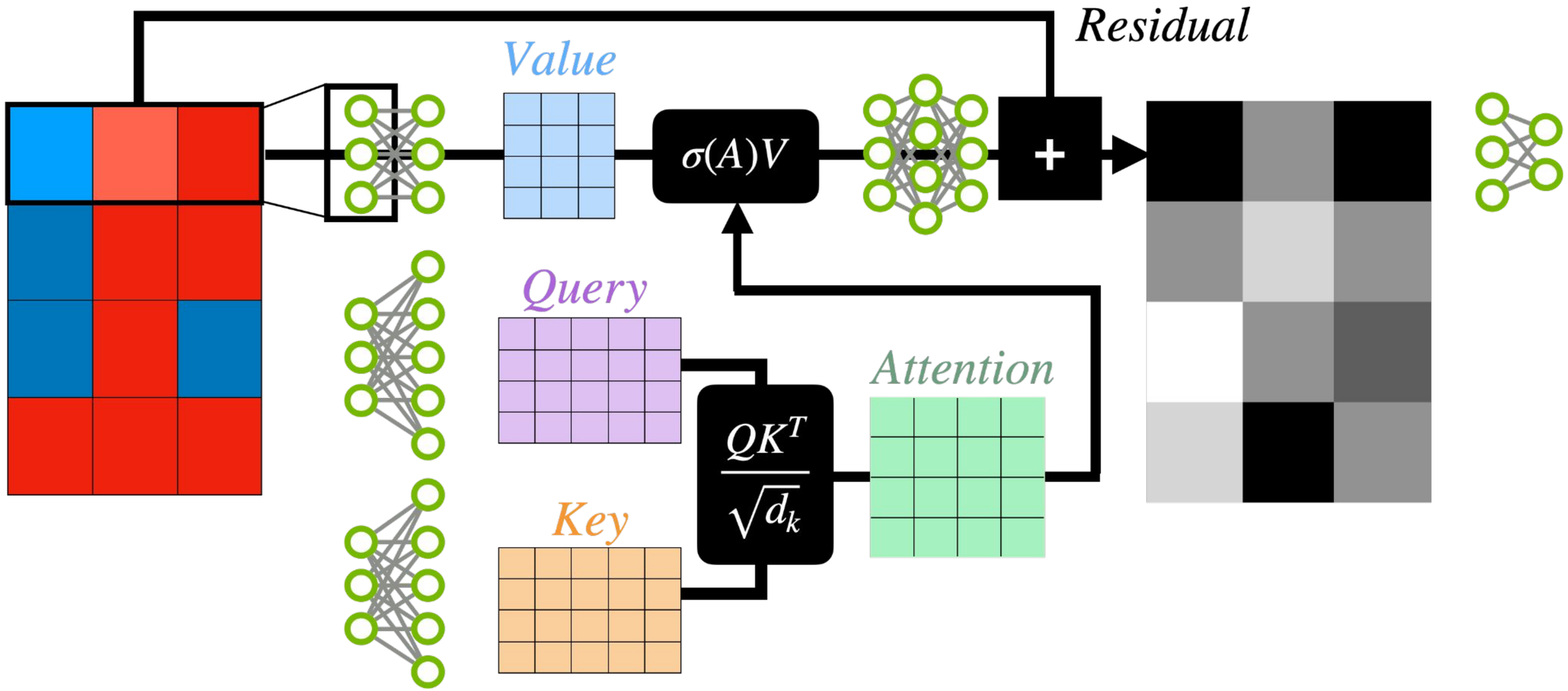
Transformers for Sequence Reasonings

Ordered **Sequence** of Classes



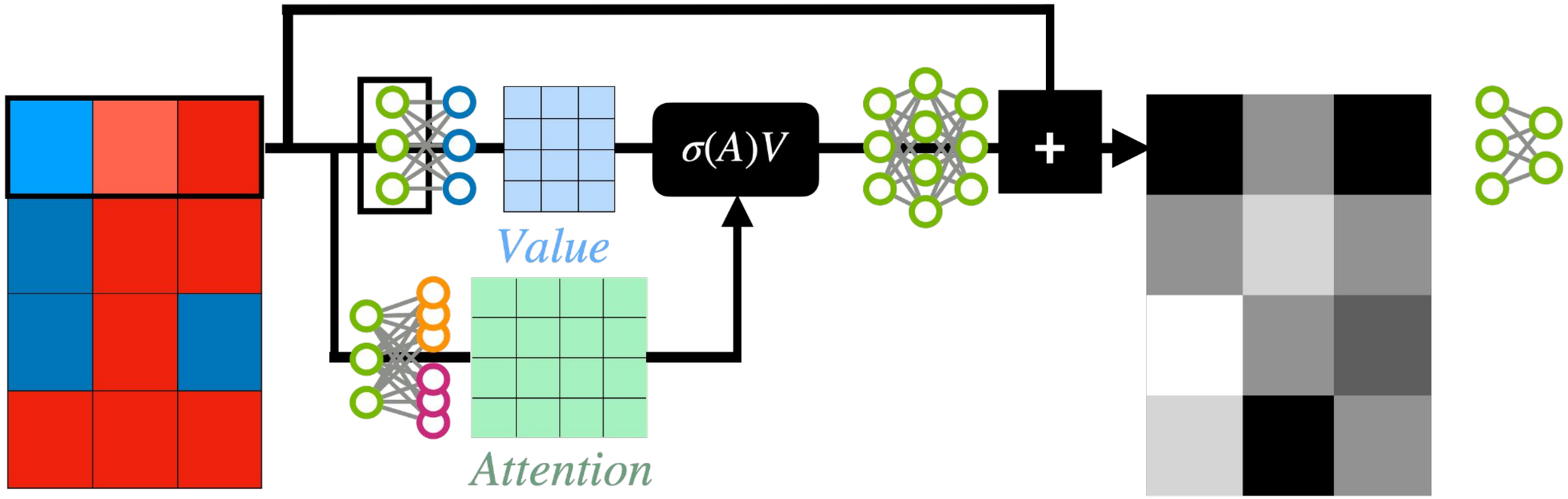
Transformers for Sequence Reasonings

Ordered **Sequence** of Classes



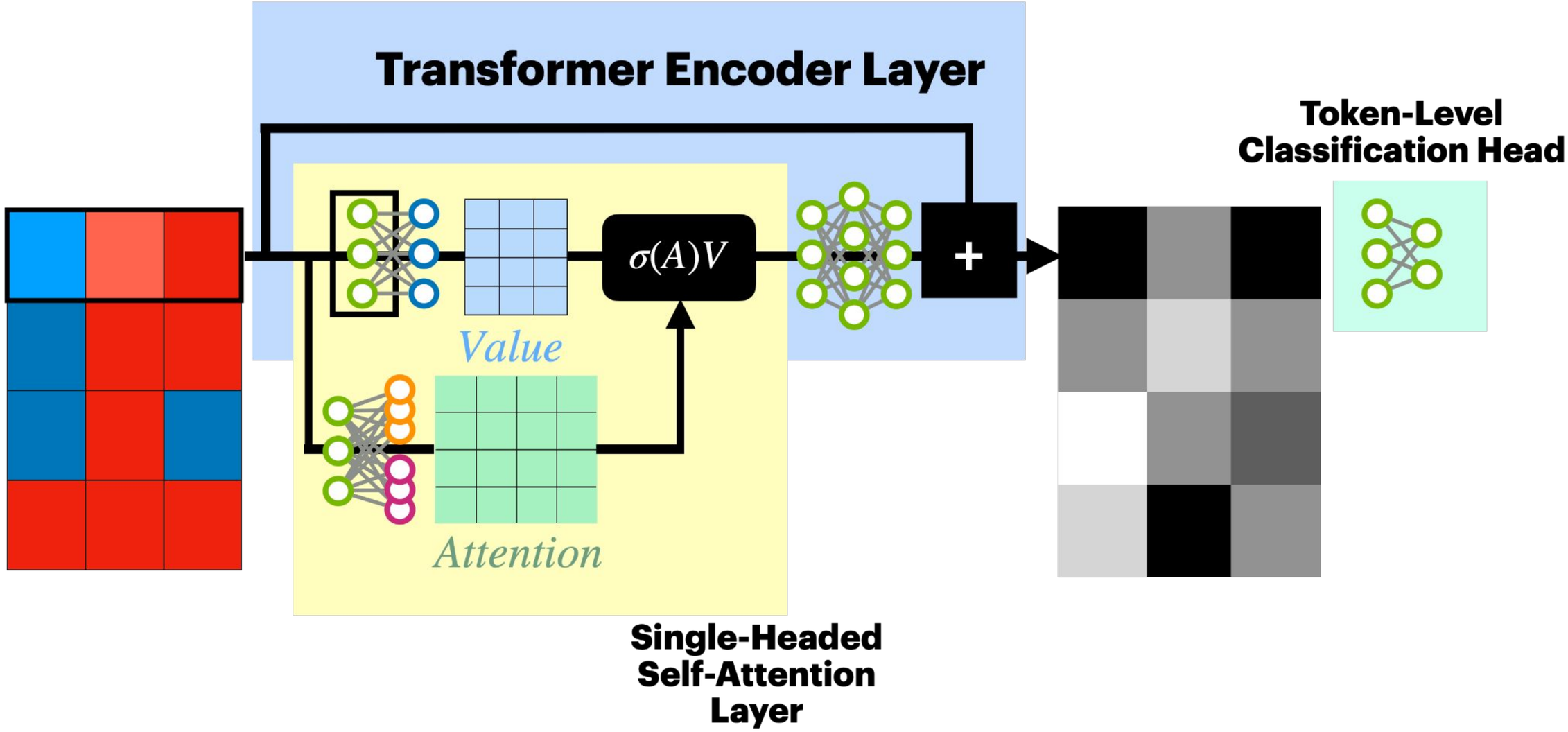
Transformers for Sequence Reasonings

Ordered **Sequence** of Classes



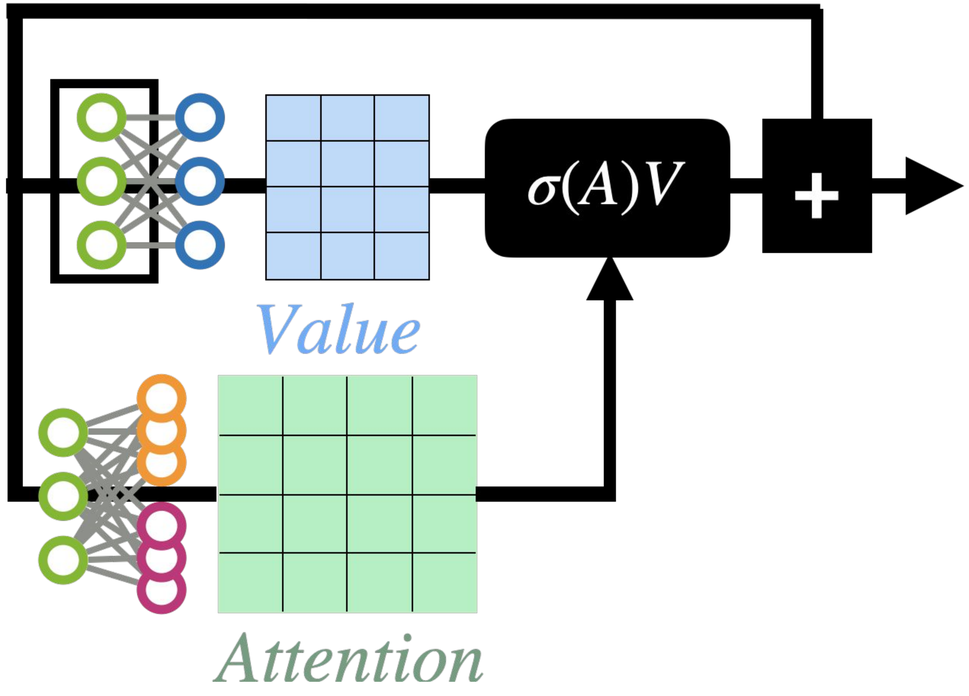
Transformers for Sequence Reasonings

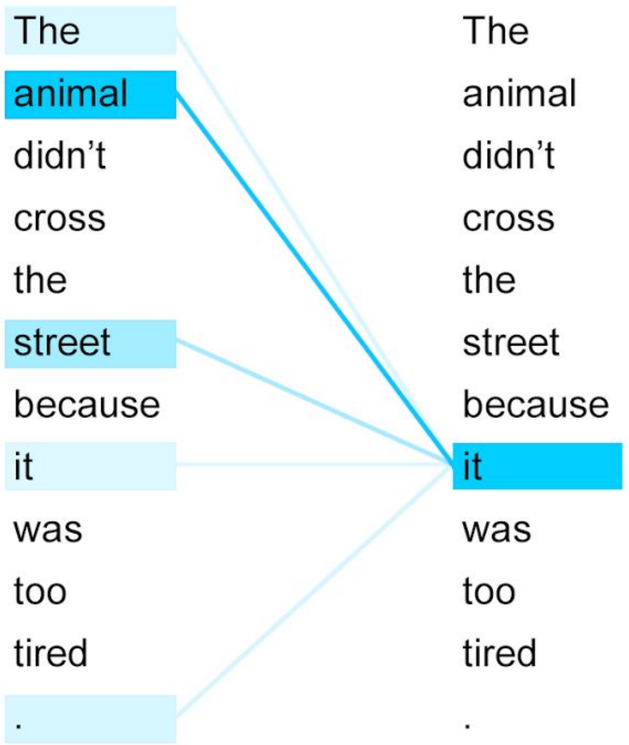
Ordered **Sequence** of Classes



Transformers Intuition

Attention Is All You Need



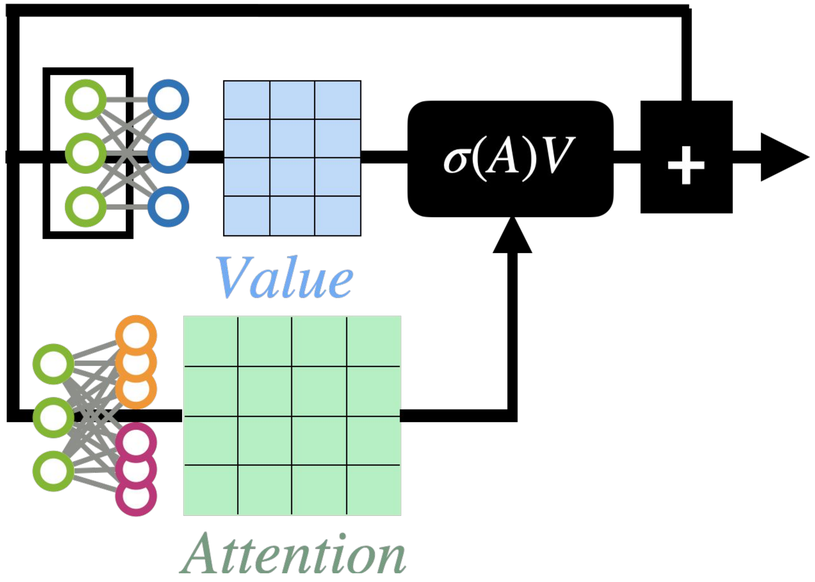


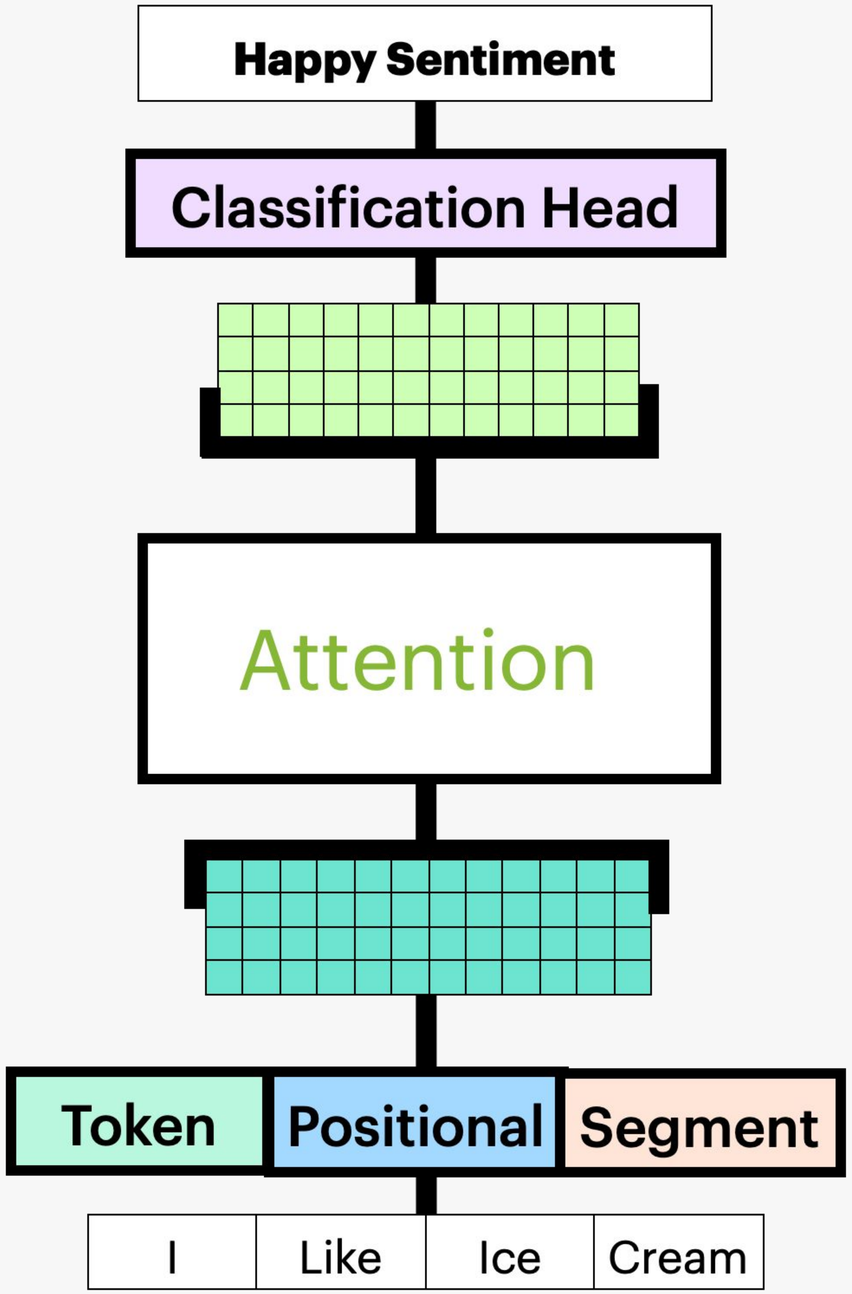
https://blog.research.google/2017/08/transformer-novel-neural-network.html?m=1

Transformer Value Add

Natural Sequence Reasoners

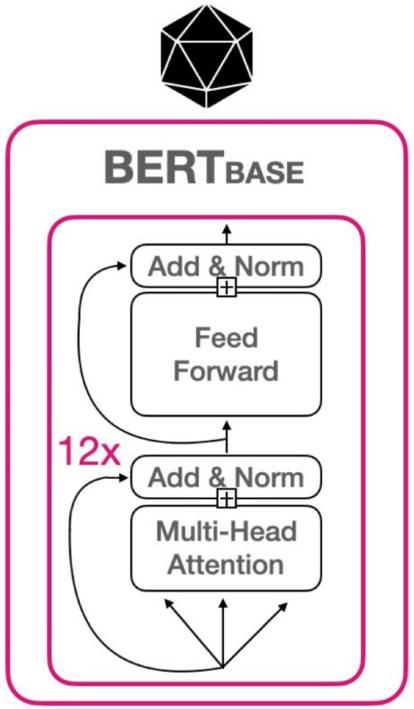
- Lightweight Entry Interface via Attention
- Residual Connection To Enforce Ordering
- Interpretable and Transferable Representation



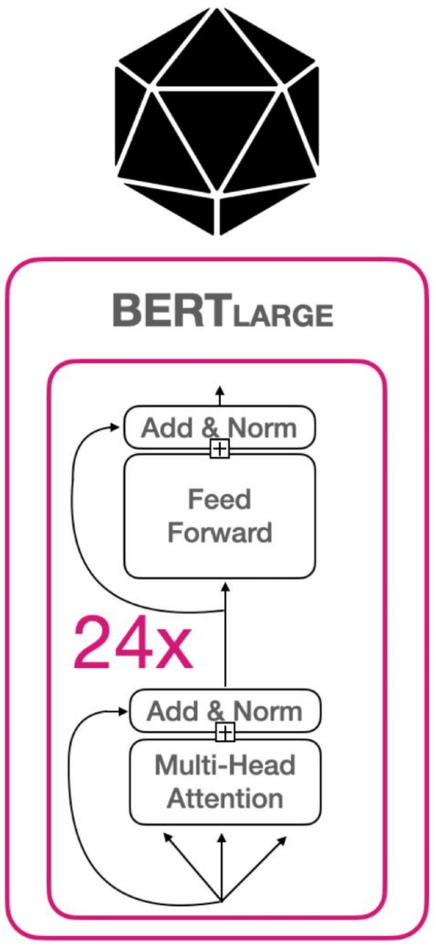


“Encoder” Transformers In Practice

What Does SOTA Look Like?



110M Parameters



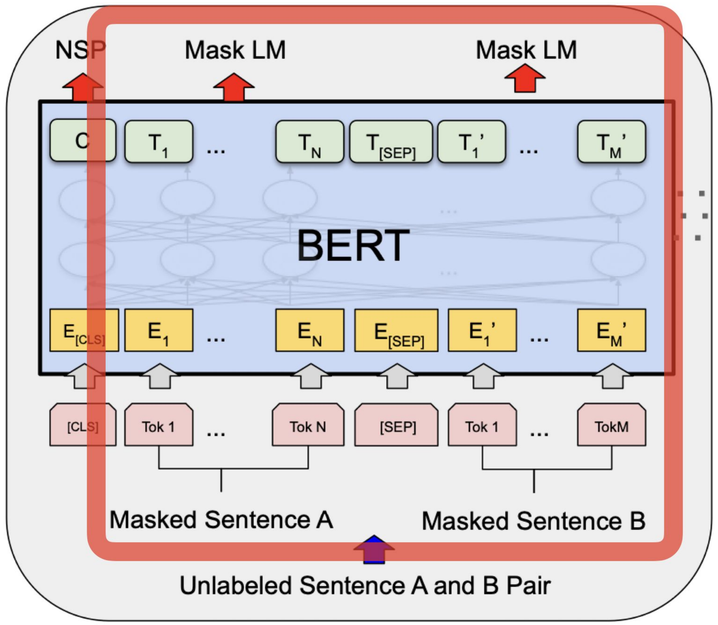
340M Parameters

https://huggingface.co/blog/bert-101

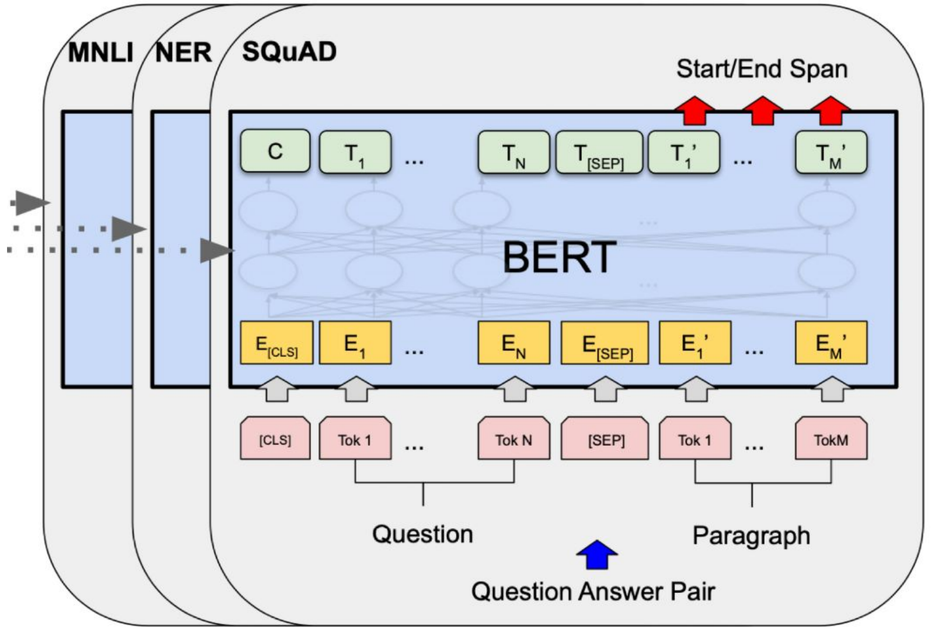
BERT Model

Powerful Encoder Model

- **Encoder:** Encodes ***n***-sequence into ***n***-encoding.
- Trained on Masked Language Modeling (MLM).
- Internet-Scale Pre-Training.
- Just embeddings and transformer blocks.
- **Used as language-reasoning backbone.**



Pretraining

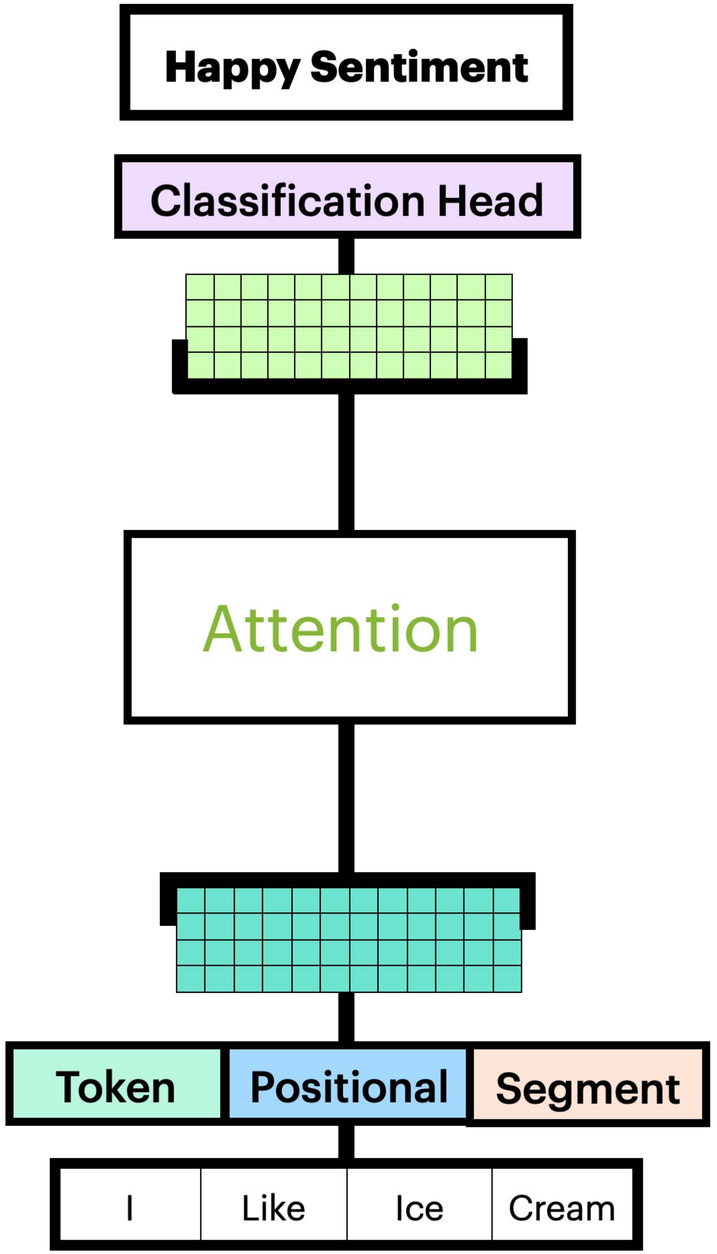


Fine-tuning

https://arxiv.org/pdf/1810.04805v2.pdf

From Transformer to BERT

Necessary Modifications



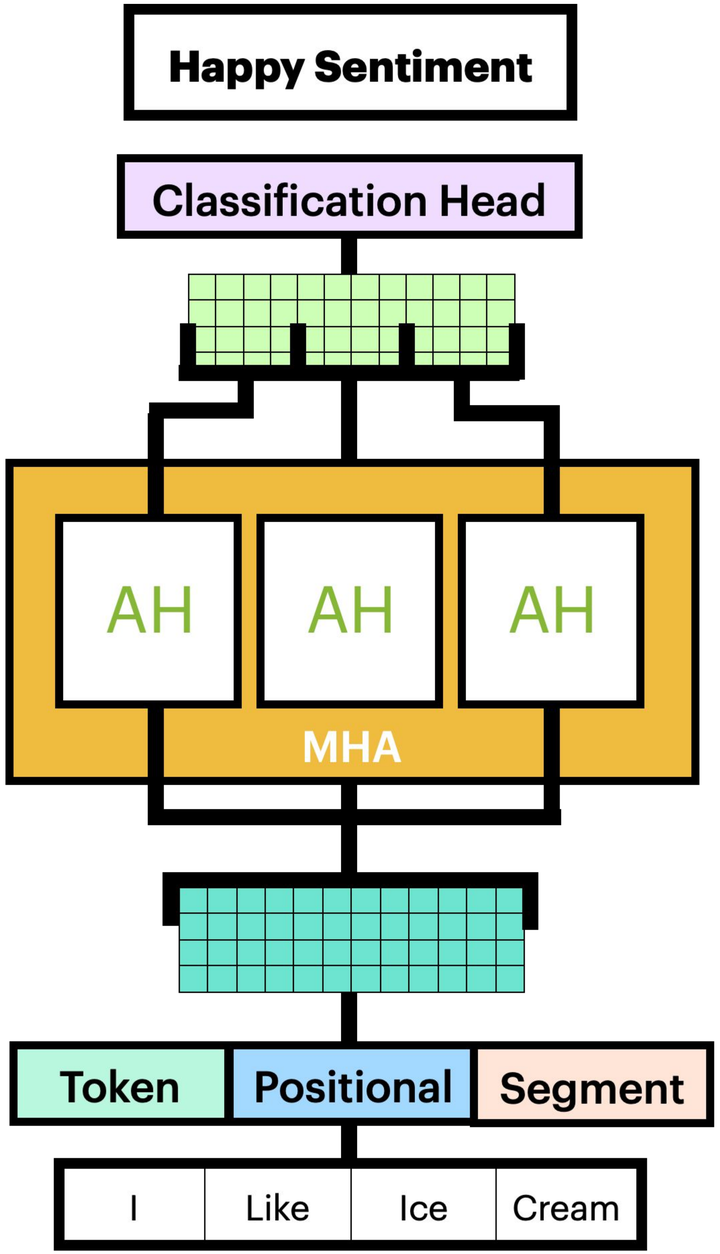
https://arxiv.org/pdf/1810.04805v2.pdf

From Transformer to BERT

Necessary Modifications

Problem: Need Deeper Reasoning

- On same data representation? **Multi-Headed Attention**



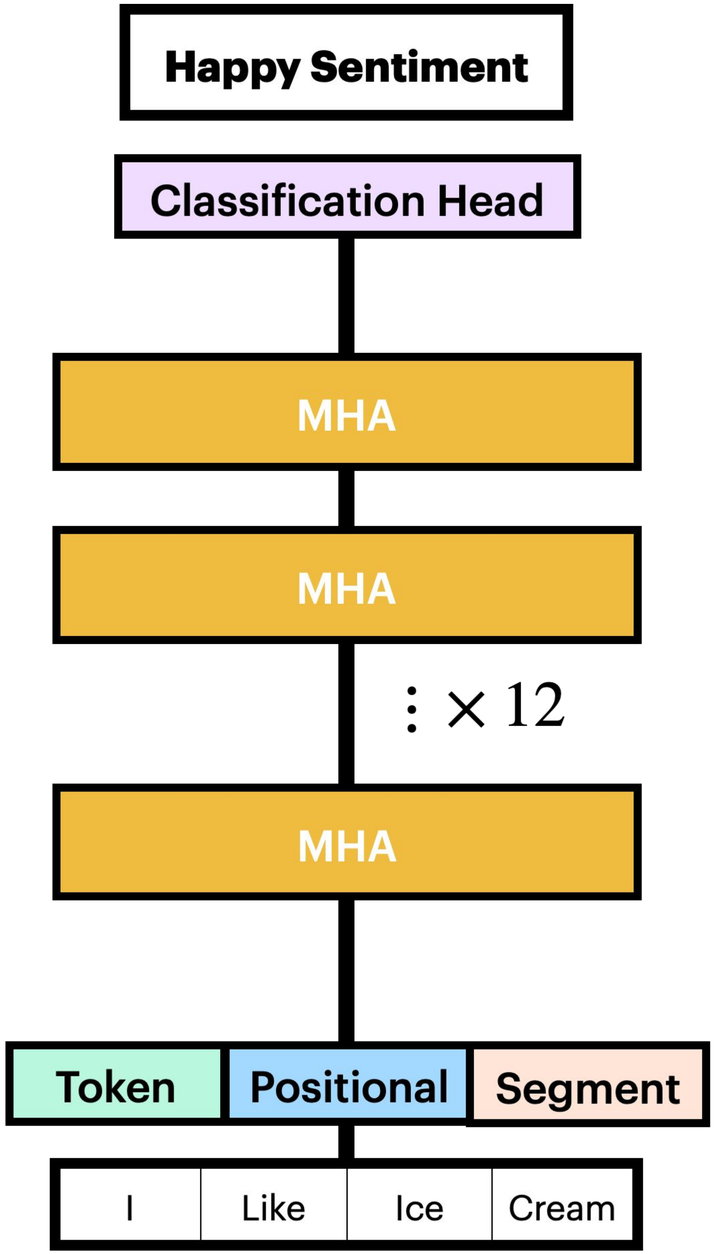
https://arxiv.org/pdf/1810.04805v2.pdf

From Transformer to BERT

Necessary Modifications

Problem: Need Deeper Reasoning

- On same data representation? **Multi-Headed Attention**
- Need more representations? **Multiple Transformer Layers**



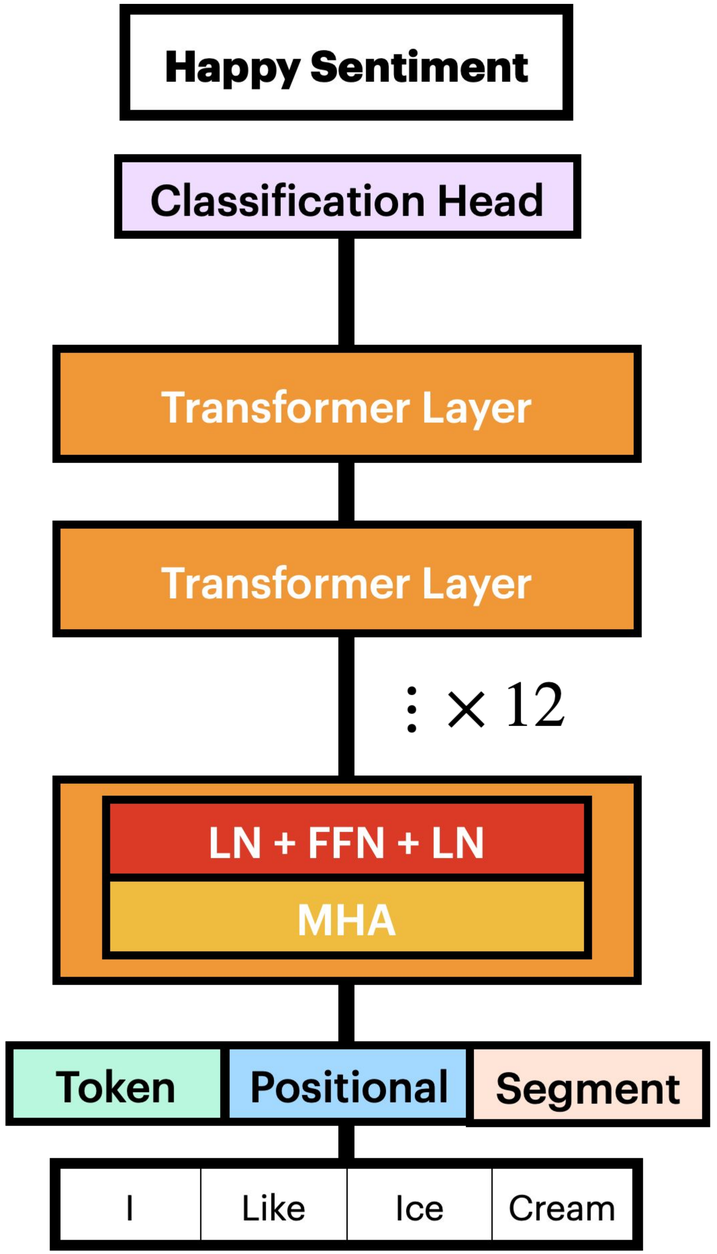
https://arxiv.org/pdf/1810.04805v2.pdf

From Transformer to BERT

Necessary Modifications

Problem: Need Deeper Reasoning
- On same data representation? **Multi-Headed Attention**
- Need more representations? **Multiple Transformer Layers**

Problem: Volatile gradients behavior, distribution shifts.
- **Layer Normalization** *(as seen in BERT)*
- **Gaussian Error Linear Unit (GELU)** *(as seen in BERT)*
- **Dropout During Training** *(as seen in BERT)*



https://arxiv.org/pdf/1810.04805v2.pdf

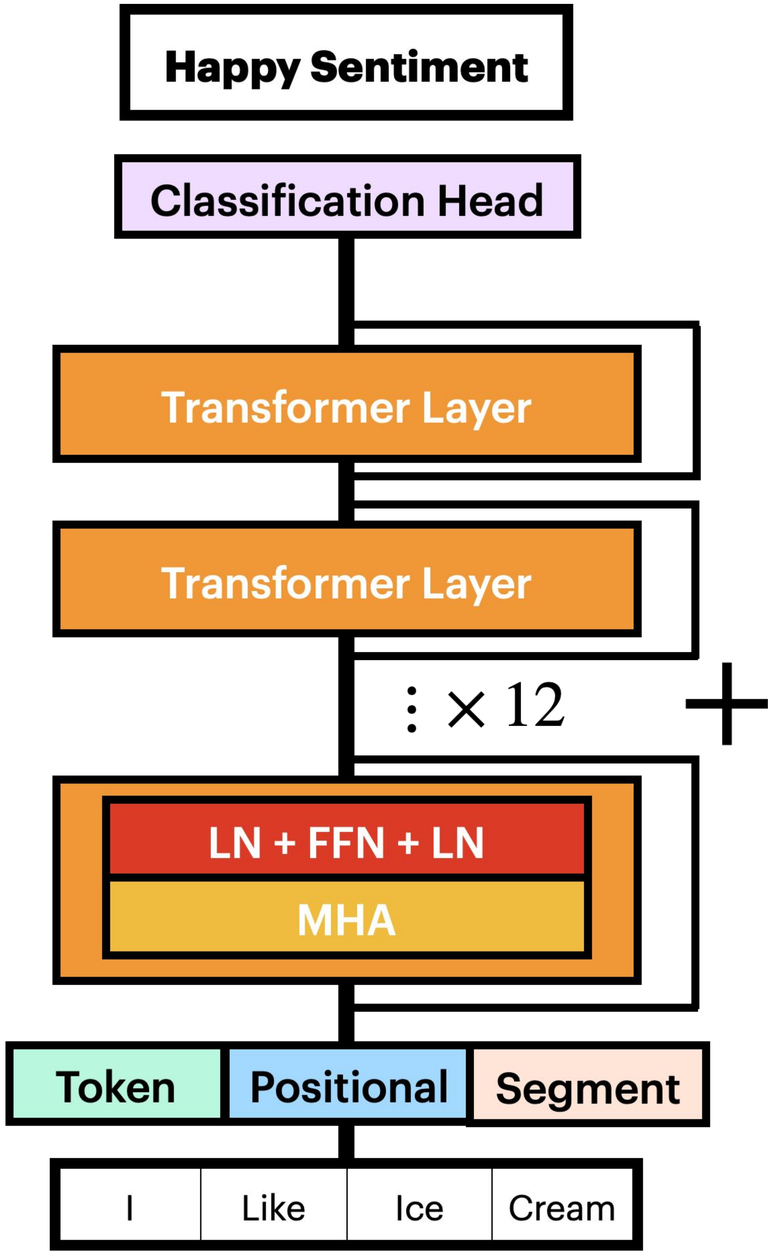
From Transformer to BERT

Necessary Modifications

Problem: Need Deeper Reasoning
- On same data representation? **Multi-Headed Attention**
- Need more representations? **Multiple Transformer Layers**

Problem: Volatile gradients behavior, distribution shifts.
- **Layer Normalization** *(as seen in BERT)*
- **Gaussian Error Linear Unit (GELU)** *(as seen in BERT)*
- **Dropout During Training** *(as seen in BERT)*

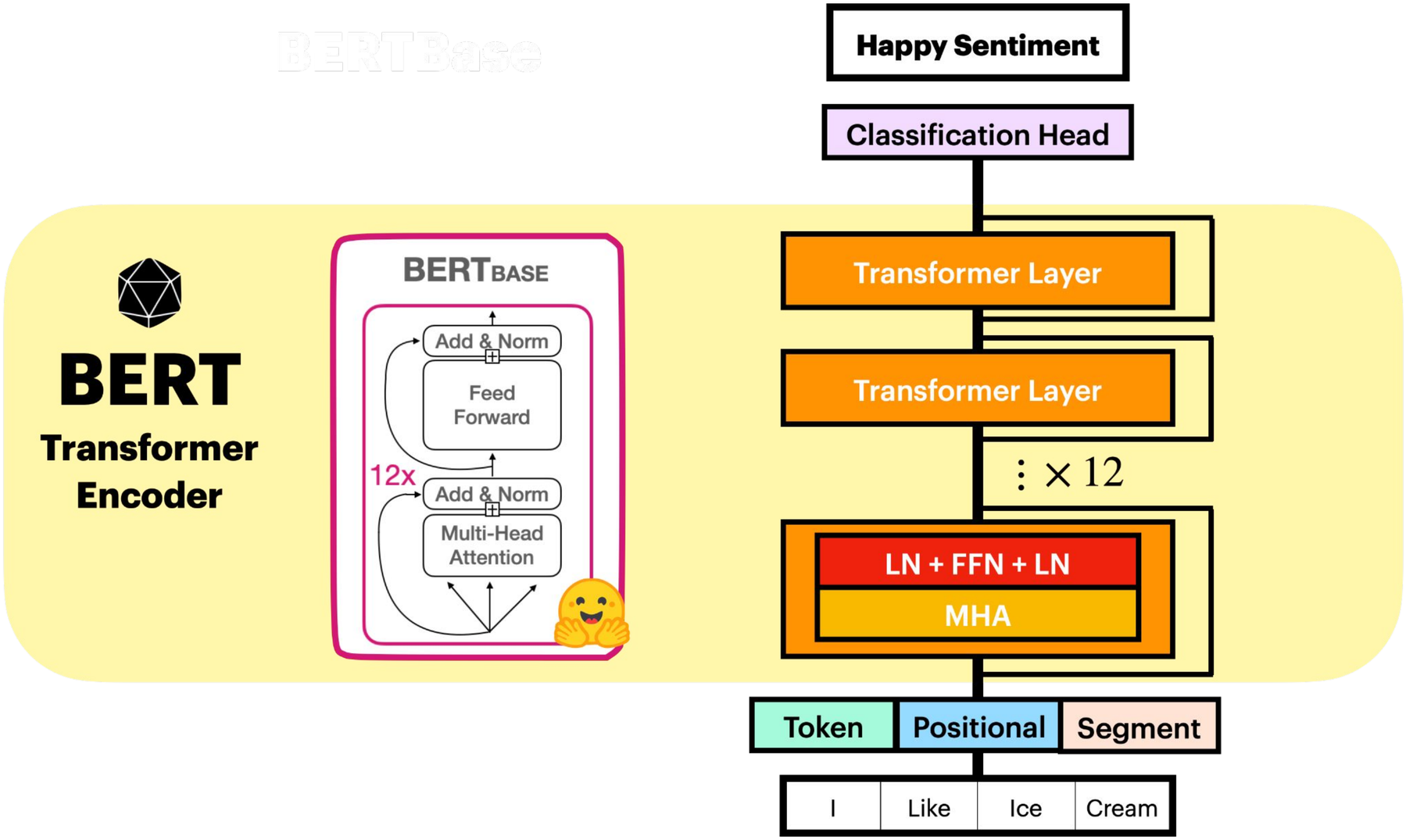
Problem: Network too deep; Gradients are vanishing
- **Residual Connections**



https://arxiv.org/pdf/1810.04805v2.pdf

BERT Constructed

From Transformer to BERT



https://huggingface.co/blog/bert-101

Part 1: From DL To LLMs

- From Deep Learning to Language Modeling
- Pulling In Our First Model
- Model Architecture Intuitions
- Using Encoders For Encoding

HF Pipeline Options

Automatic End-to-End Pipelines

NLP Classification	feature-extraction text-classification token-classification fill-mask
NLP Response	question-answering summarization translation text2text-generation zero-shot-classification
	table-qa document-qa
NLP Generative	text-generation conversational
Audio	audio-classification automatic-speech-recognition text-to-audio zero-shot-audio-classification
Vision	visual-question-answering image-classification image-segmentation image-to-text object-detection depth-estimation video-classification mask-generation zero-shot-image-classification zero-shot-object-detection

Natural Encoder Tasks

From n-sequence Encodings

feature extraction
token classification
mask filling
text classification

Feature Extraction

Feature extraction refers to the process of transforming raw data into numerical features that can be processed while preserving the information in the original dataset.

Inputs

Input

India, officially the Republic of India, is a country in South Asia.

Feature Extraction Model

Output

Dim1	Dim2	Dim3
2.58	2.75	0.90
8.29	1.10	2.03
-0.77	-1.64	-0.61
0.07	1.59	1.46

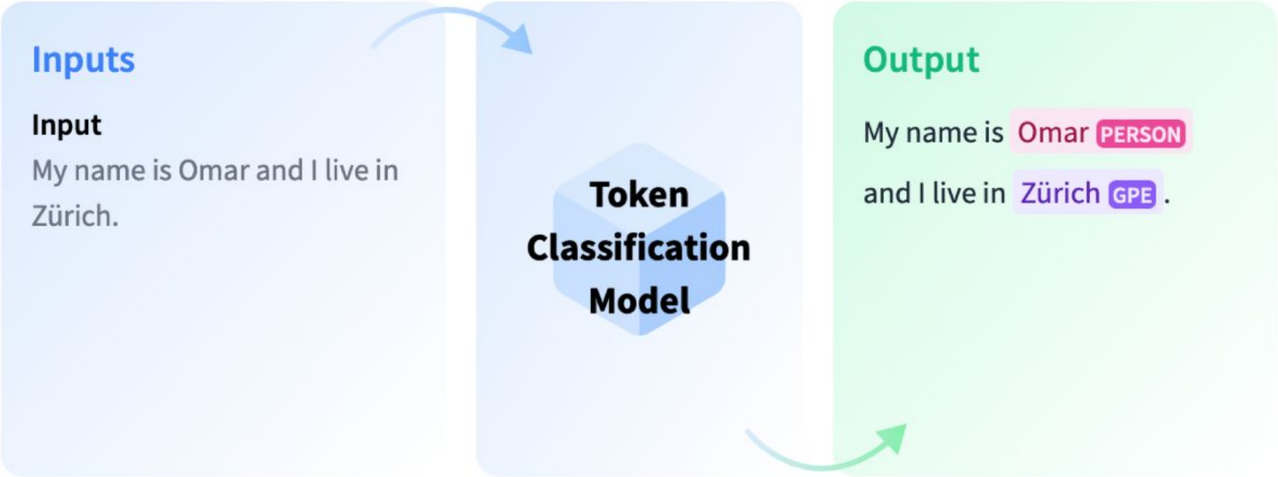
Natural Encoder Tasks

From n-sequence Encodings

feature extraction
token classification
mask filling
text classification

Token Classification

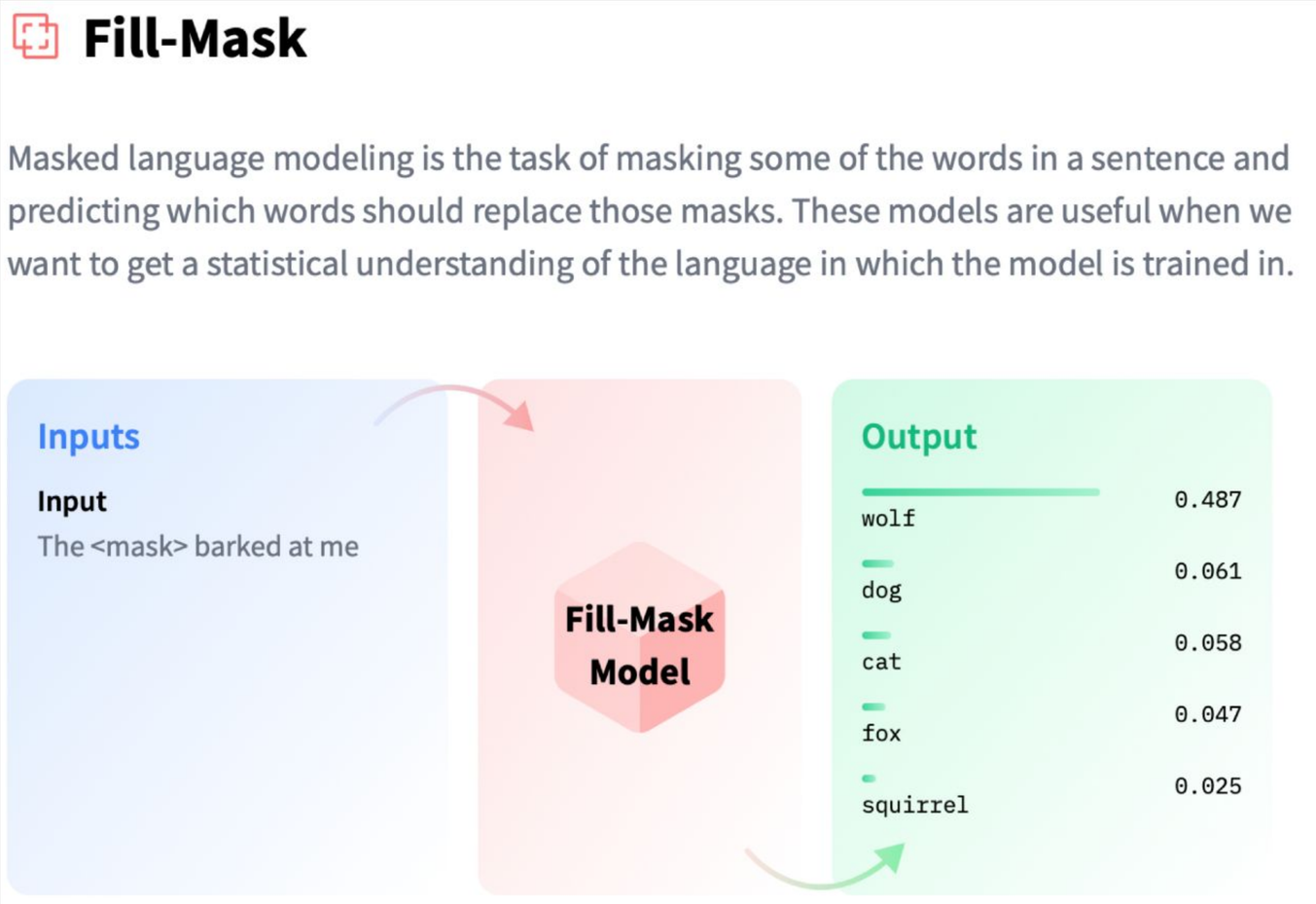
Token classification is a natural language understanding task in which a label is assigned to some tokens in a text. Some popular token classification subtasks are Named Entity Recognition (NER) and Part-of-Speech (PoS) tagging. NER models could be trained to identify specific entities in a text, such as dates, individuals and places; and PoS tagging would identify, for example, which words in a text are verbs, nouns, and punctuation marks.



Natural Encoder Tasks

From n-sequence Encodings

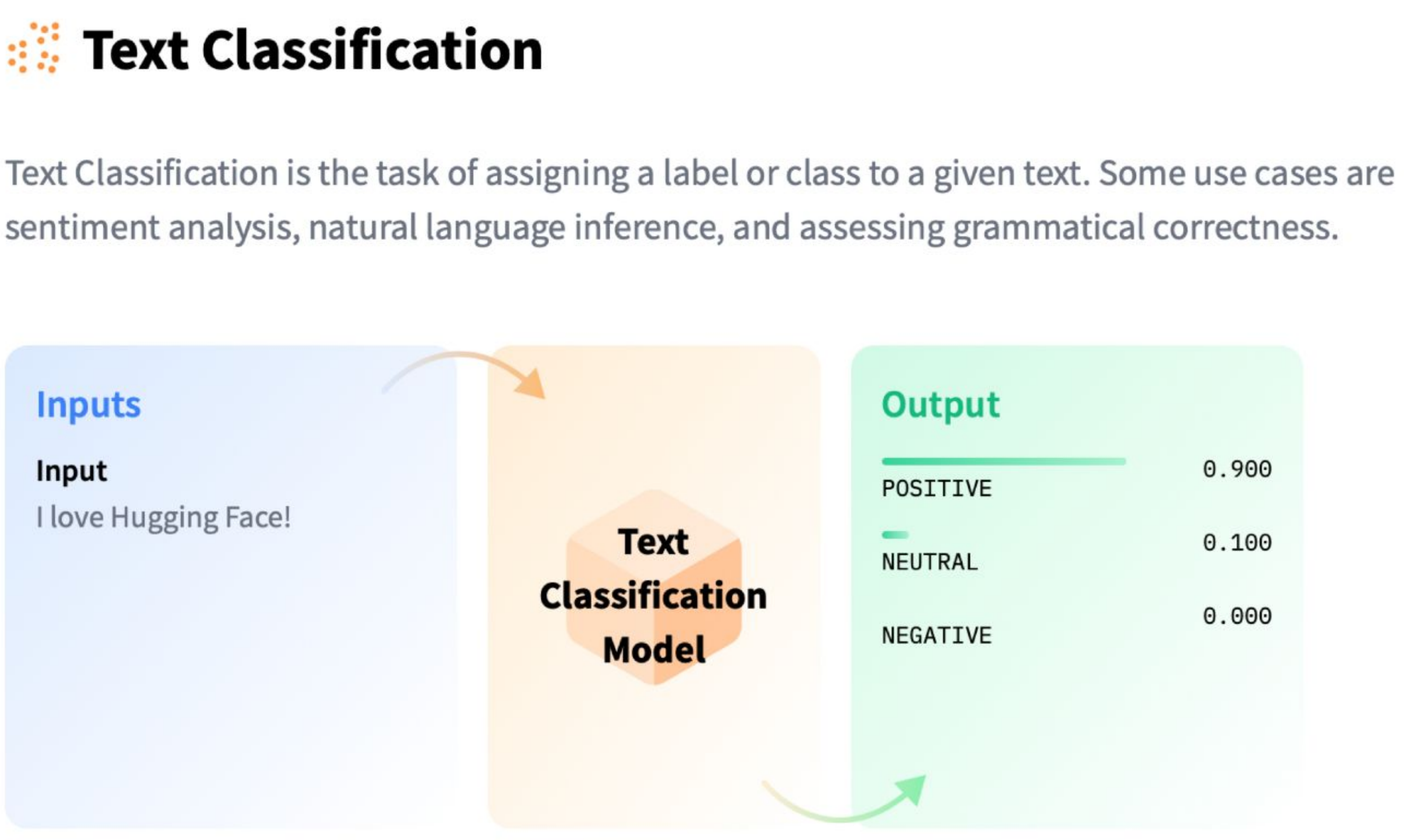
feature extraction
token classification
mask filling
text classification



Natural Encoder Tasks

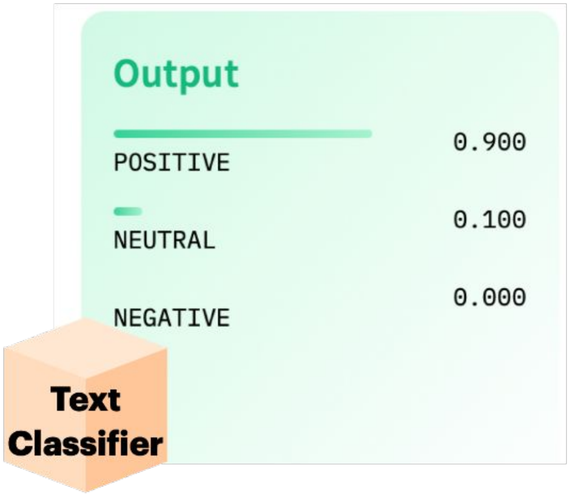
From n-sequence Encodings

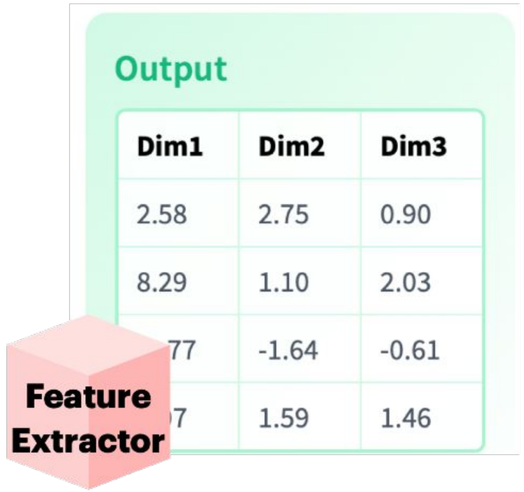
feature extraction
token classification
mask filling
text classification

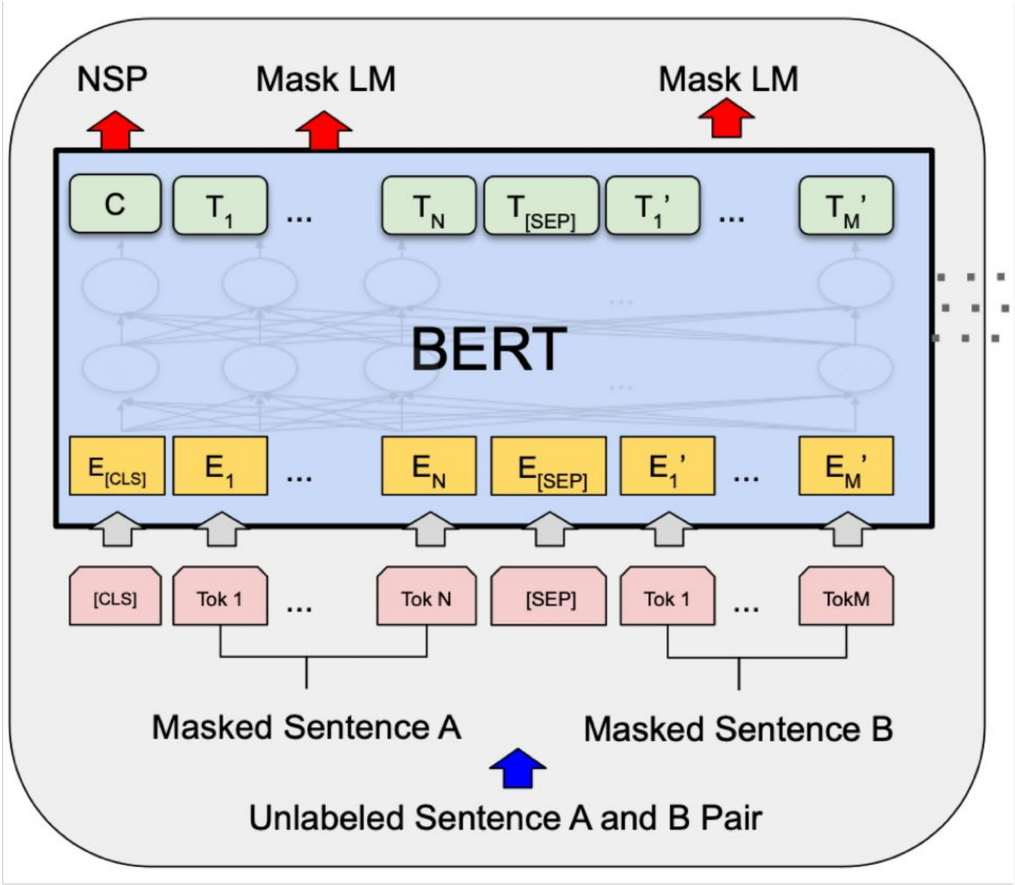


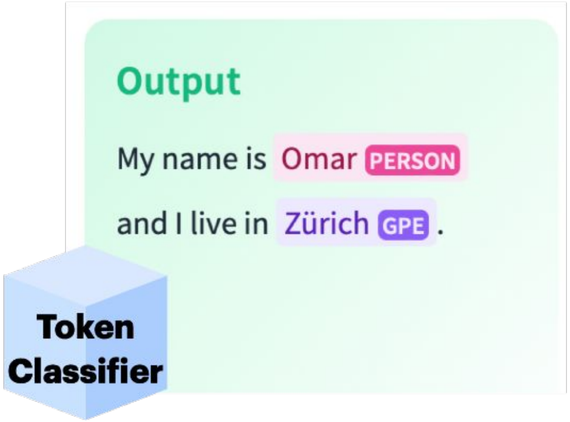
Common Encoder Backbone

From n-sequence Encodings



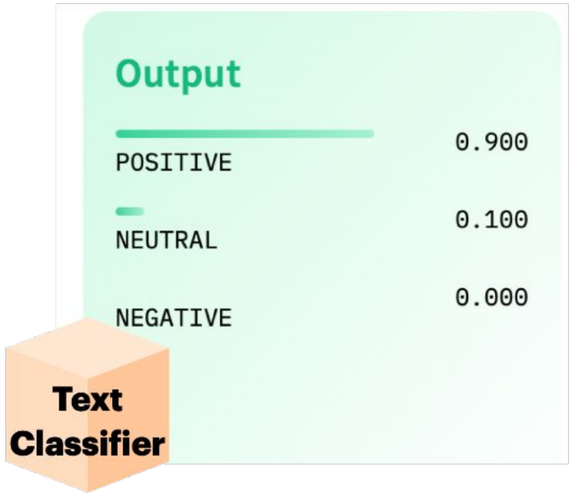


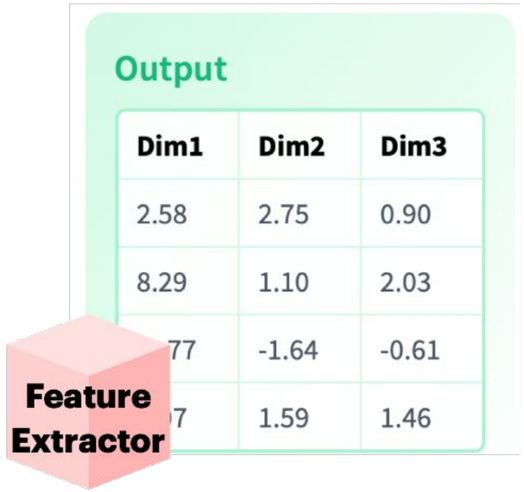


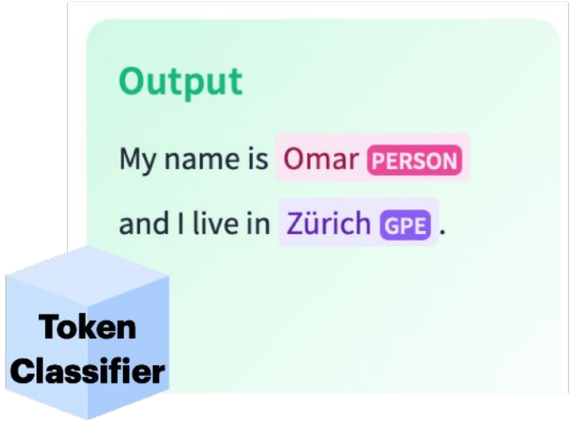


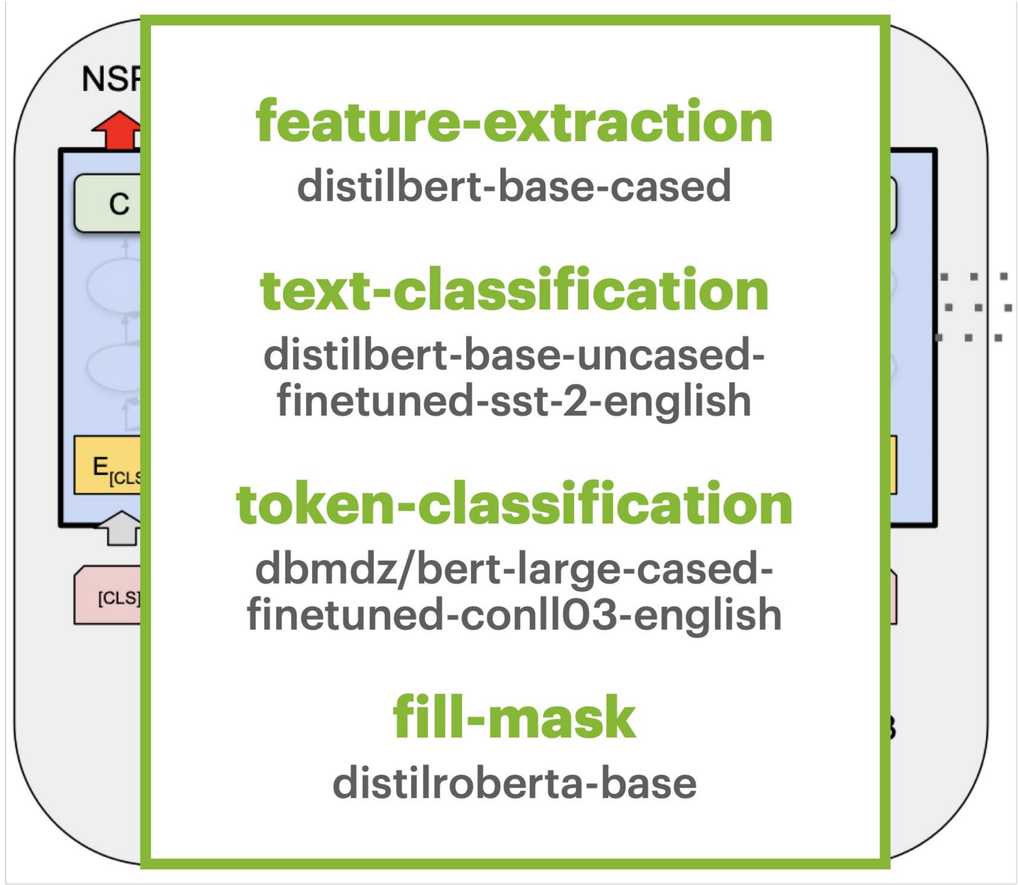
Common Encoder Backbone

From n-sequence Encodings









Encoder Model Options

Common Derivatives/Flavors

Default Base Models

feature-extraction
distilbert-base-cased

text-classification
distilbert-base-uncased-finetuned-sst-2-english

token-classification
dbmdz/bert-large-cased-finetuned-conll03-english

fill-mask
distilroberta-base

Pre-Training

BERT:
Uses Masked-LM and Next-Sentence Prediction

RoBERTa:
10x as much data, dynamic masking, no NSP

XML-RoBERTa:
Much more data trained across 100 languages

Model Size/Scope

Size/Distillation/Casing

Fine-Tuning

SST2, CONLL, SQUaD

Encoder Model Options

Common Derivatives/Flavors

Default Base Models

feature-extraction
distilbert-base-cased

text-classification
distilbert-base-uncased-finetuned-sst-2-english

token-classification
dbmdz/bert-large-cased-finetuned-conll03-english

fill-mask
distilroberta-base

Pre-Training

BERT, RoBERTa, XML

Model Size/Scope

small/med/large/xl/etc:
Different architectures with different param #

DistilModel:
Smaller model trained to mimic a larger model

Cased/Uncased/etc:
Restrictions on tokenization strategy

Fine-Tuning

SST2, CONLL, SQUaD

Encoder Model Options

Common Derivatives/Flavors

Default Base Models

feature-extraction
distilbert-base-cased

text-classification
distilbert-base-uncased-finetuned-sst-2-english

token-classification
dbmdz/bert-large-cased-finetuned-conll03-english

fill-mask
distilroberta-base

BERT, RoBERTa, XML	Pre-Training
Size/Distillation/Casing	Model Size/Scope

Fine-Tuning

SST2:
~215K positive/negative-sentiment passages

CONLL:
~20K named entity recognition (NER) examples

SQUaD2.0:
~100K question/answer pairs (+50K Q-only)

BERT Benefits

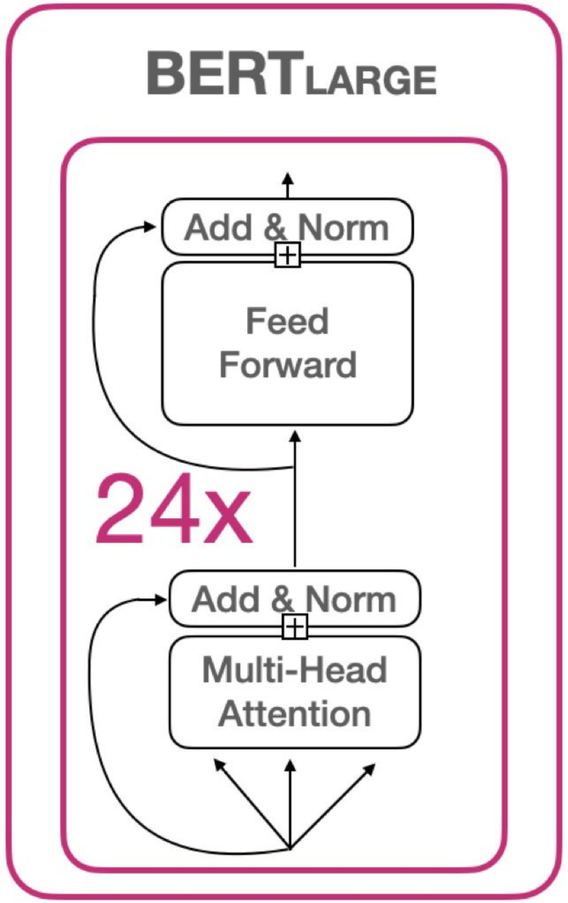
When BERT-Style Encoders Shine

Key Features: Simple and Small, But Powerful

- Simple input-output schema; strong supervision.
- Top-performing models usually <1B parameters.
- Transferable embeddings/attention interfaces.
- Constant bidirectional enforcement

BEST WHEN

- When n->n, n->1, or subsetting is sufficient.



340M Parameters

Multiple-Query Pipelines

Towards Zero-Shot Classification for Any Class

```
<s>Dune is the best movie ever</s>
This is about Cinema  -> 0.6
This is about Art     -> 0.3
This is about Music   -> 0.1

Contradiction/Entailment Task
```

Zero-Shot Classification

Zero-shot text classification is a task in natural language processing where a model is trained on a set of labeled examples but is then able to classify new examples from previously unseen classes.

BERT Benefits

When BERT-Style Encoders Shine

Key Features: Simple and Small, But Powerful

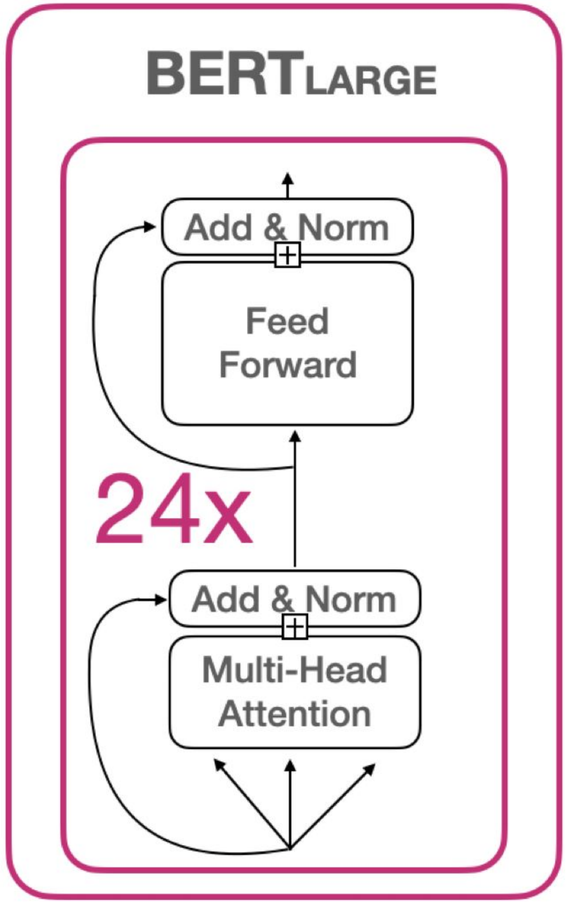
- Simple input-output schema; strong supervision.
- Top-performing models usually <1B parameters.
- Transferable embeddings/attention interfaces.
- Constant bidirectional enforcement

BEST WHEN

- n->n, n->1, n->(1,1, ...).
-

BUT NOT GOOD WHEN

- Need to generate new tokens/sequences.



340M Parameters

Variable-Length Outputs

Range Selection Sufficient, But Limited

Question Answering

Question Answering models can retrieve the answer to a question from a given text, which is useful for searching for an answer in a document. Some question answering models can generate answers without context!

Inputs

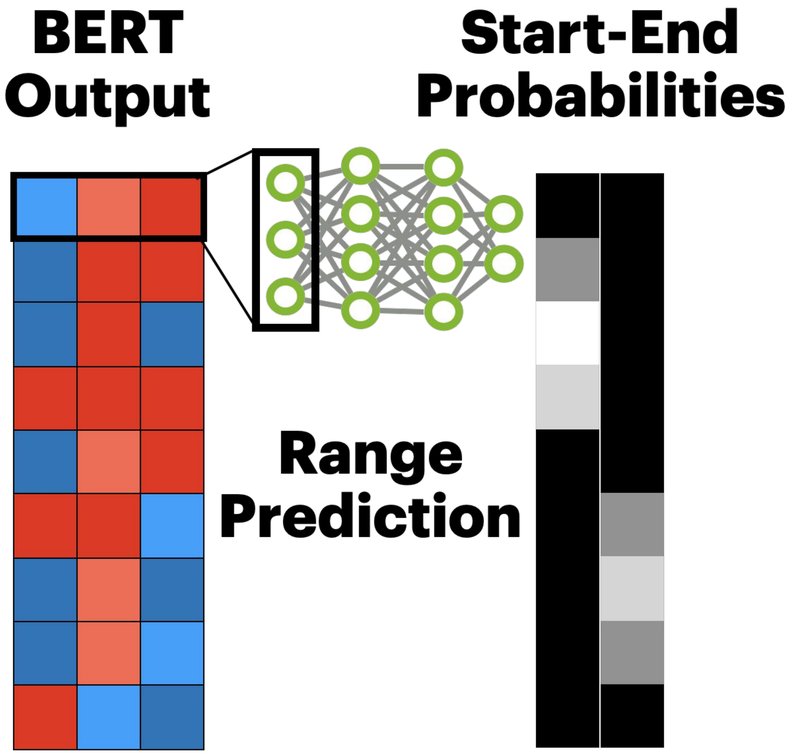
Question
Which name is also used to describe the Amazon rainforest in English?

Context
The Amazon rainforest, also known in English as Amazonia or the Amazon Jungle

Question Answer Model

Output

Answer
Amazonia



BERT Benefits

When BERT-Style Encoders Shine

Key Features: Simple and Small, But Powerful

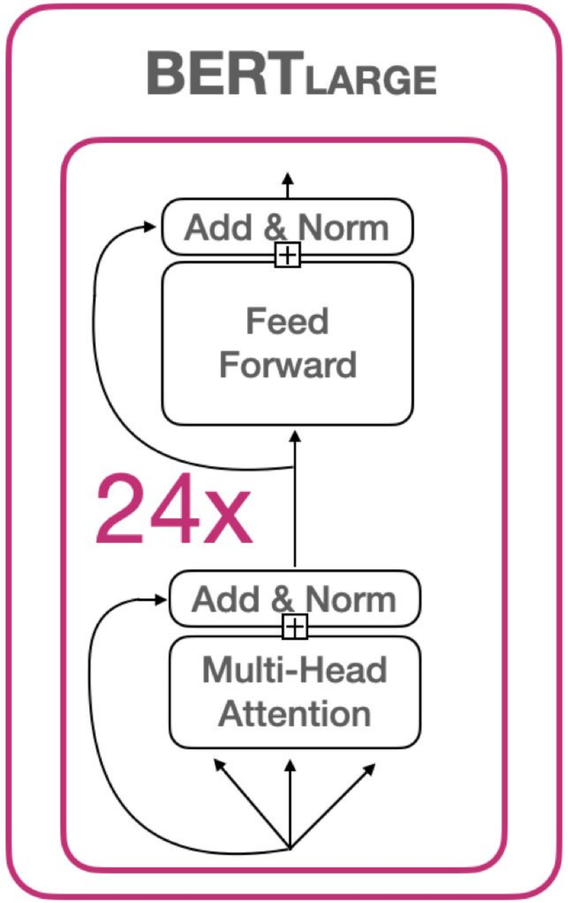
- Simple input-output schema; strong supervision.
- Top-performing models usually <1B parameters.
- Transferable embeddings/attention interfaces.
- Constant bidirectional enforcement

BEST WHEN

- n->n, n->1, n->(1,1, ...), or subsetting is sufficient.

BUT NOT GOOD WHEN

- Need to generate new tokens/sequences.



340M Parameters

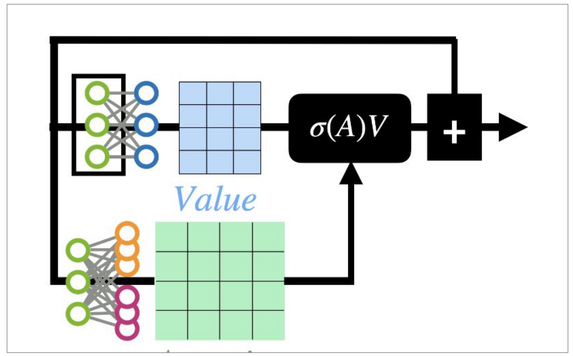
Part 2: Decoding Models

- **Encoder-Decoders for Seq2Seq**
- **Cross-Attention for Multimodal Models**
- **Large Decoder Models for Text Gen**

Course Schedule

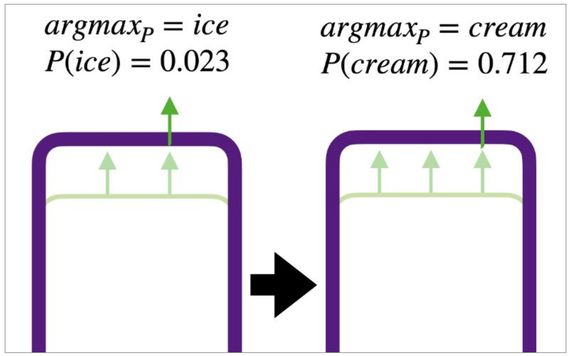
What's The Plan?

Getting Started

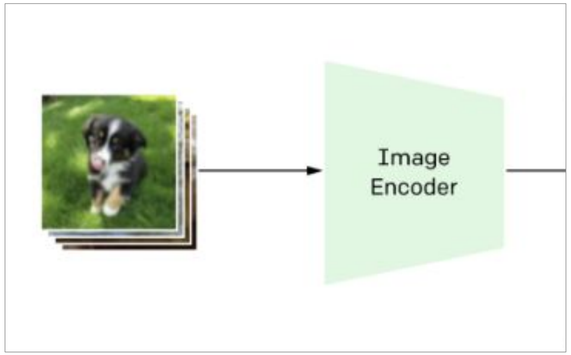


Transformer Intuitions

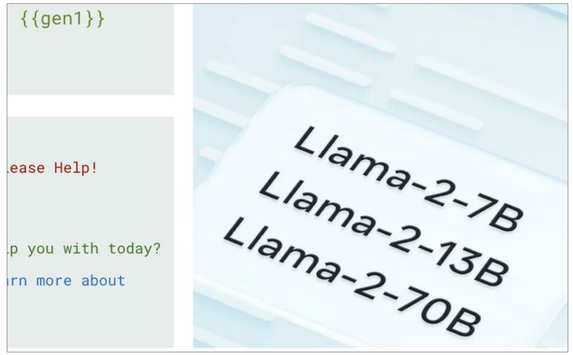
Encoder Tasks



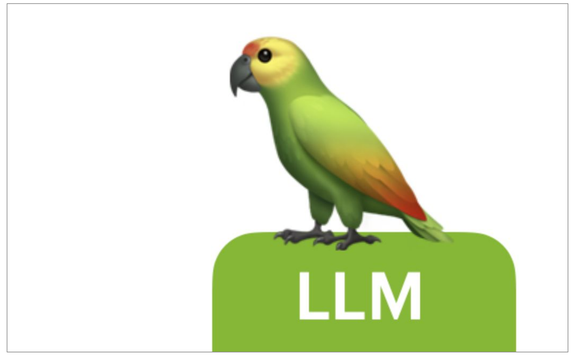
Seq2Seq



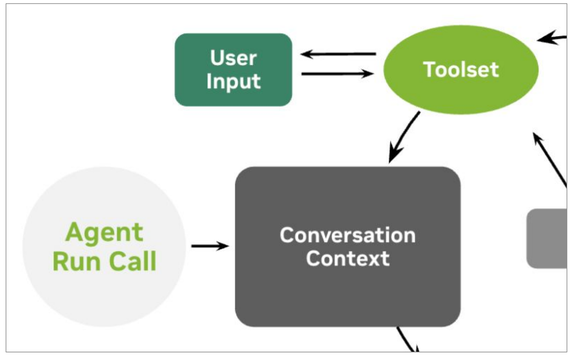
Multimodality



Text Generation



Stateful LLMs



Agent Assessment

HF Pipeline Options

Automatic End-to-End Pipelines

NLP Classification

feature-extraction
text-classification
token-classification
fill-mask

NLP Response

question-answering
summarization
translation
text2text-generation
zero-shot-classification

table-qa
document-qa

NLP Generative

text-generation
conversational

Audio

audio-classification
automatic-speech-recognition
text-to-audio
zero-shot-audio-classification

Vision

visual-question-answering
image-classification
image-segmentation
image-to-text
object-detection
depth-estimation
video-classification
mask-generation
zero-shot-image-classification
zero-shot-object-detection

Variable-Length Outputs

Range Selection Sufficient, But Limited

Question Answering

Question Answering models can retrieve the answer to a question from a given text, which is useful for searching for an answer in a document. Some question answering models can generate answers without context!

Inputs

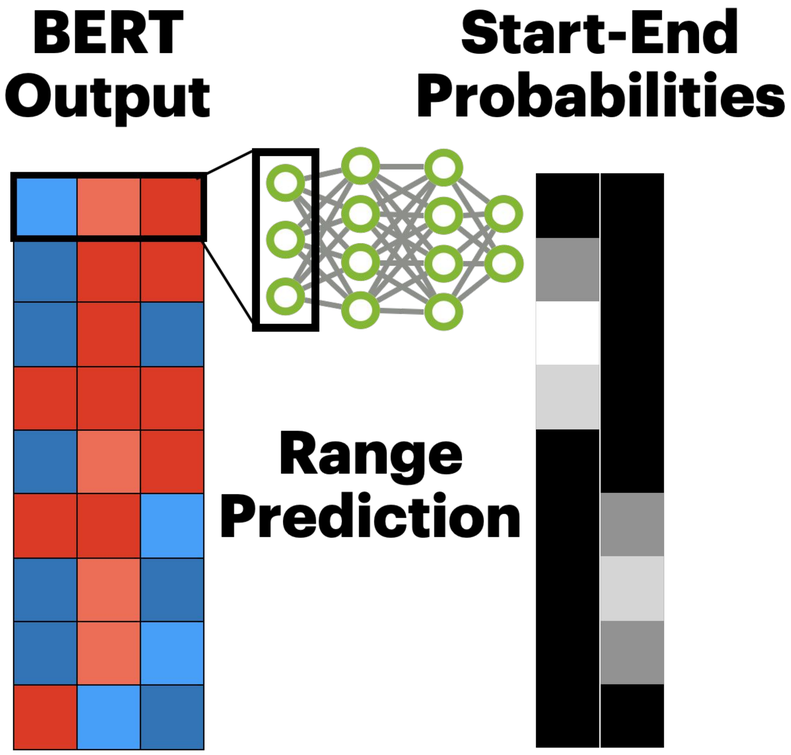
Question
Which name is also used to describe the Amazon rainforest in English?

Context
The Amazon rainforest, also known in English as Amazonia or the Amazon Jungle

Question Answer Model

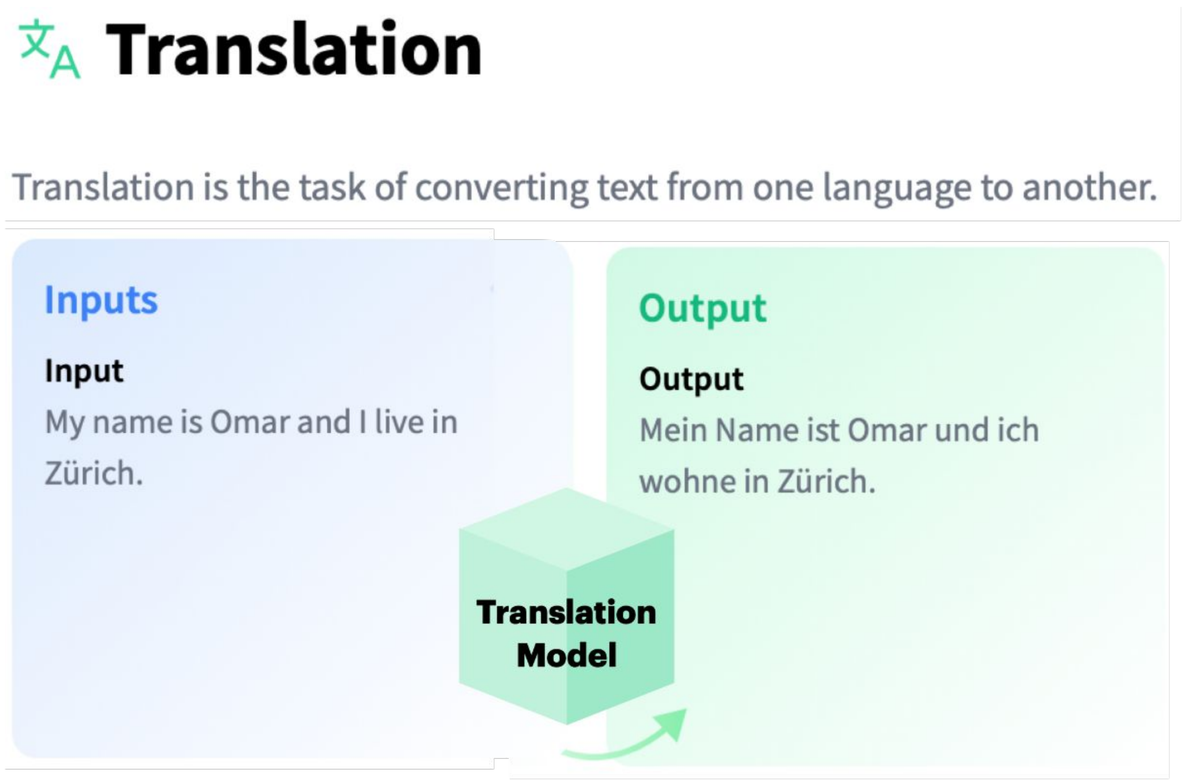
Output

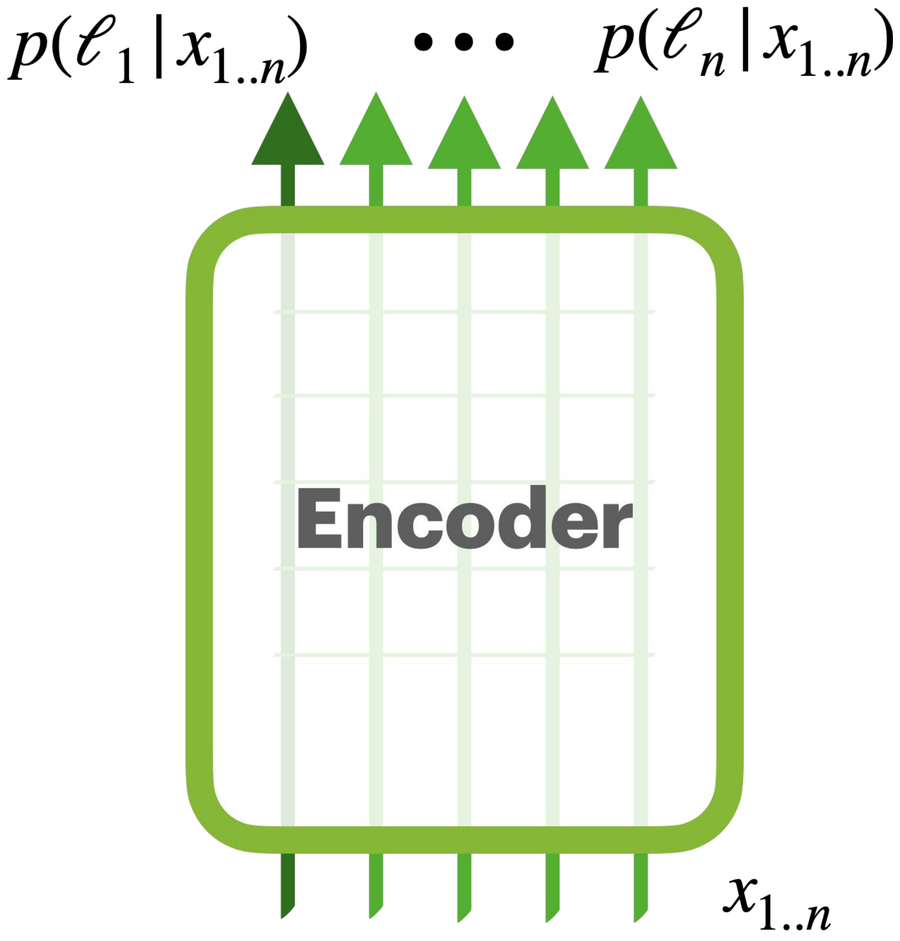
Answer
Amazonia



Translation Tasks

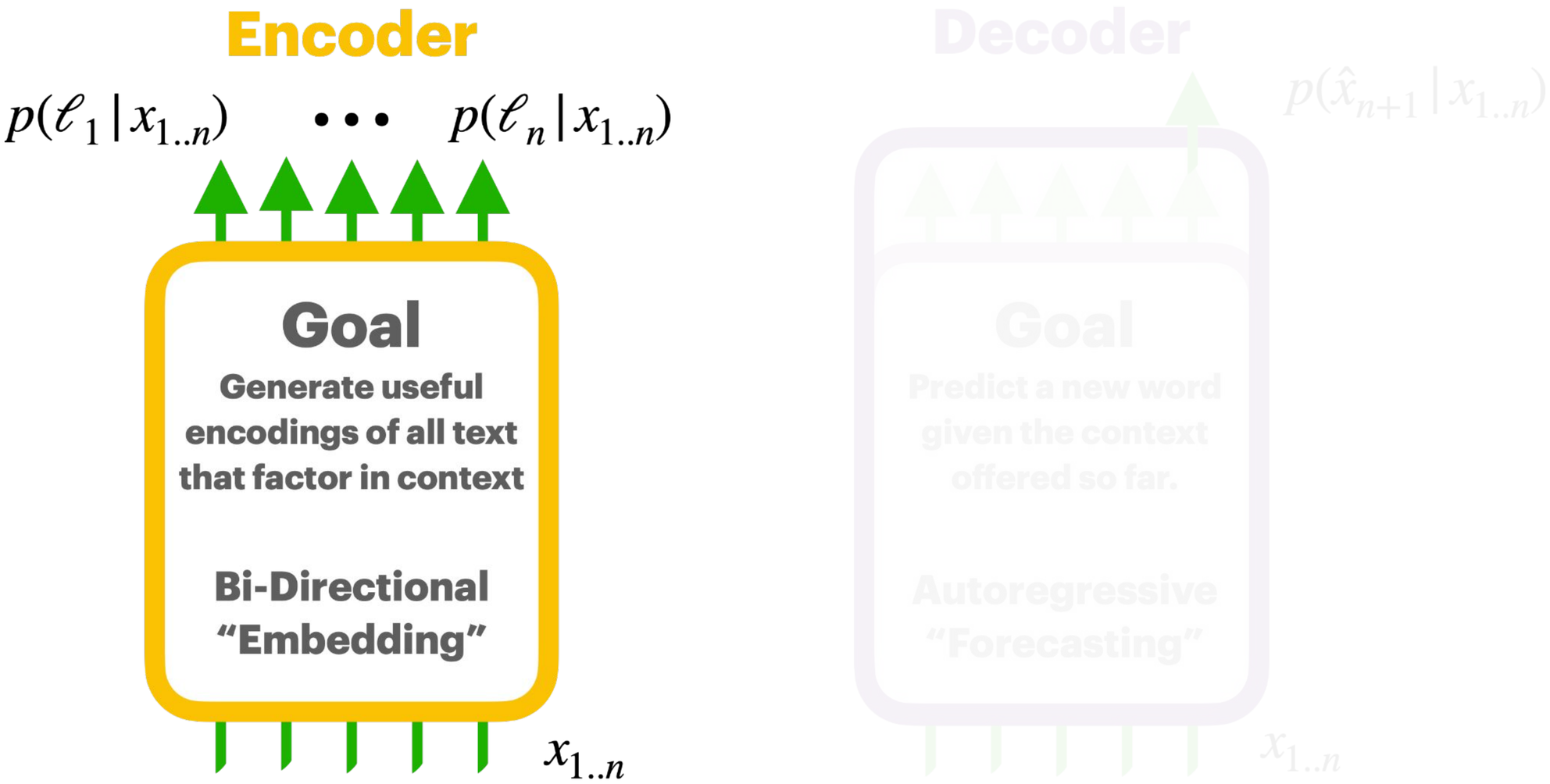
Encoder-Only Insufficient for General m->n





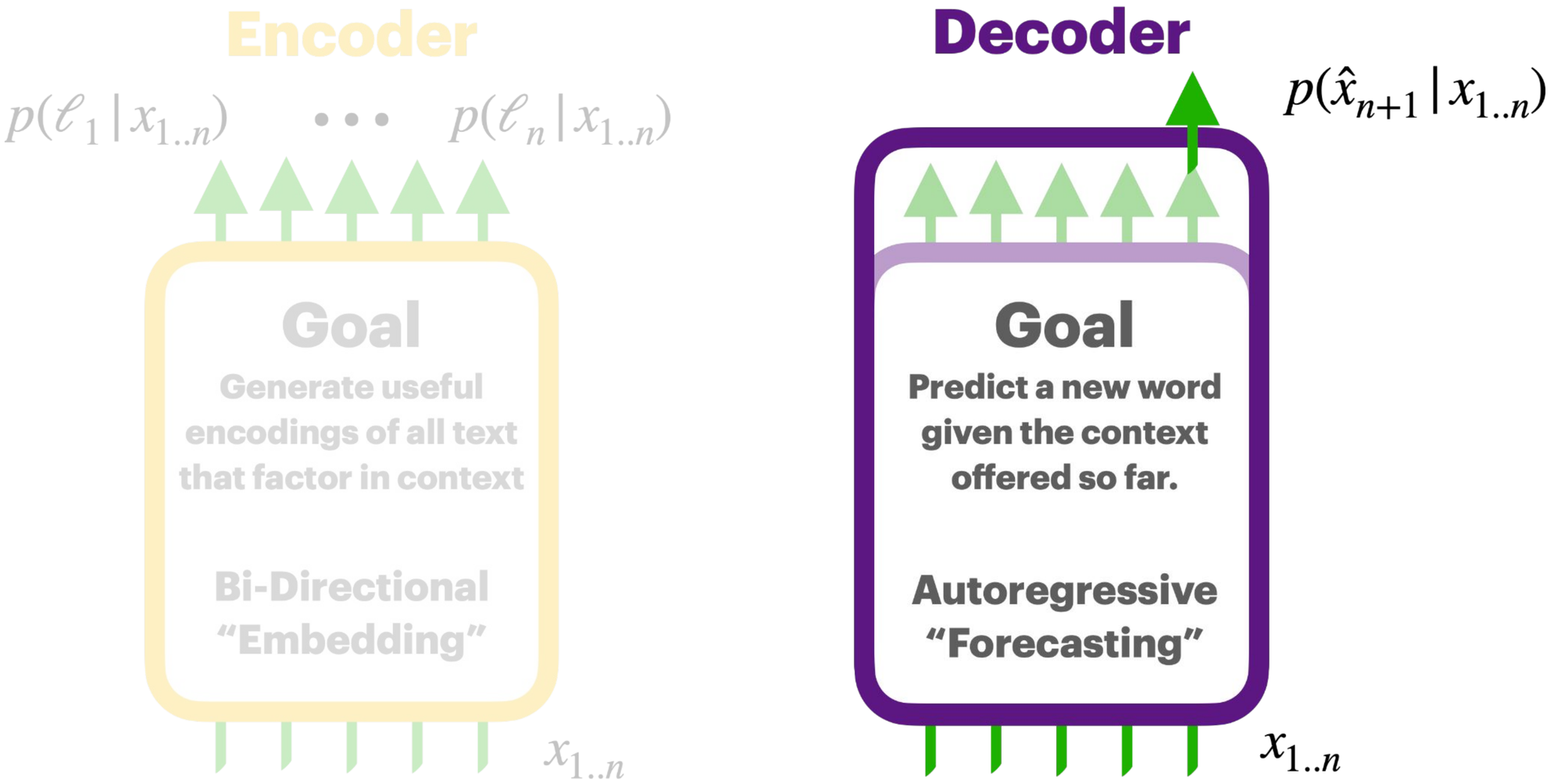
From Encoding to Decoding

Predict ?->1 To Forecast n Predictions



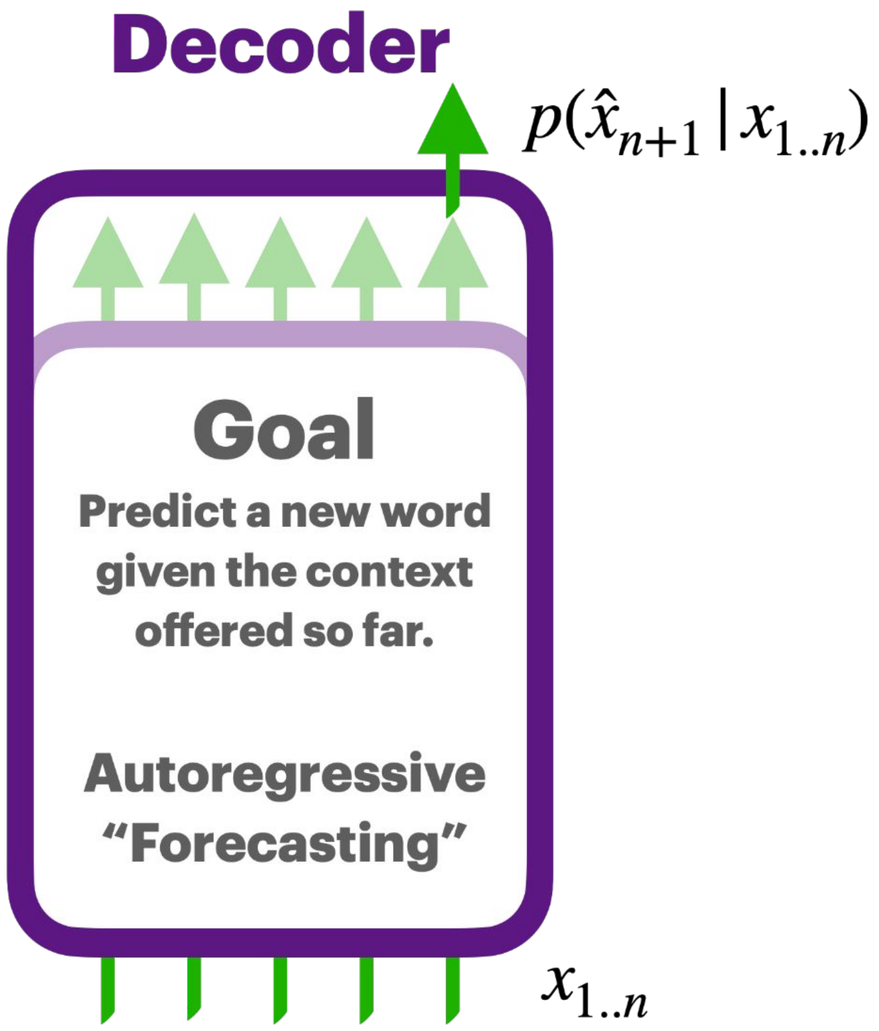
From Encoding to Decoding

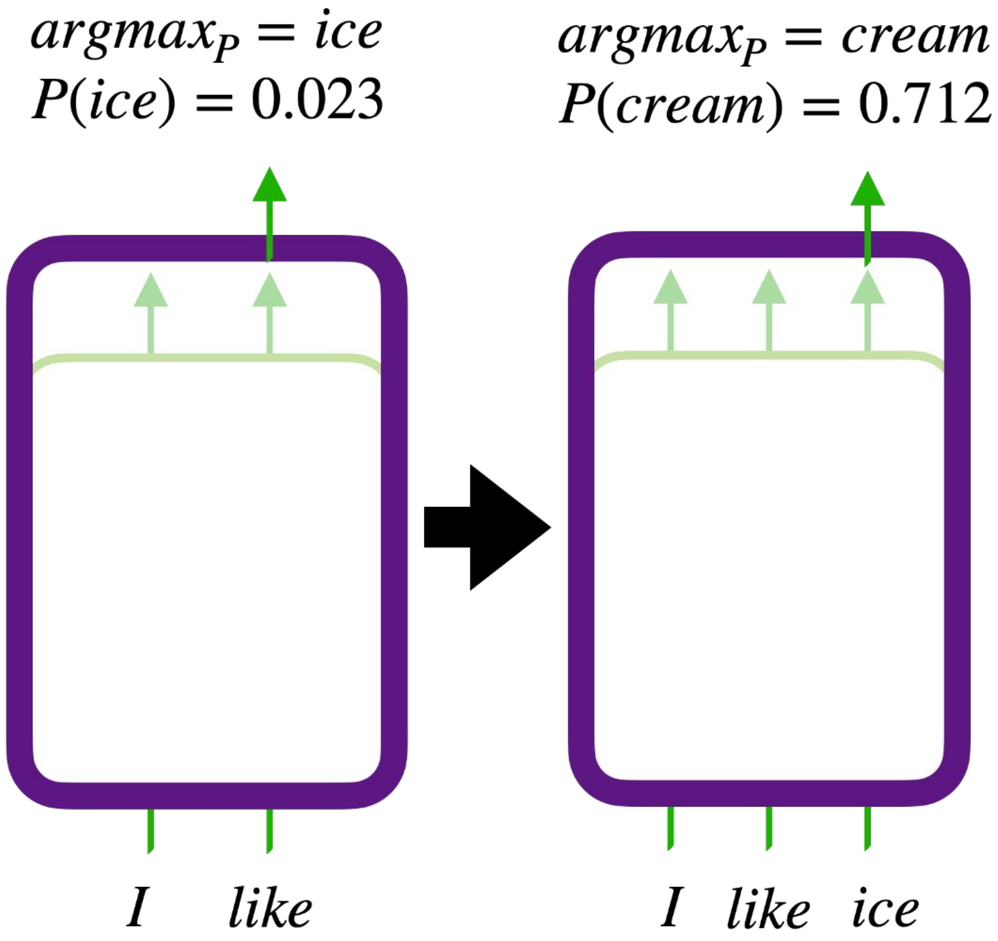
Predict ?->1 To Forecast n Predictions



From Encoding to Decoding

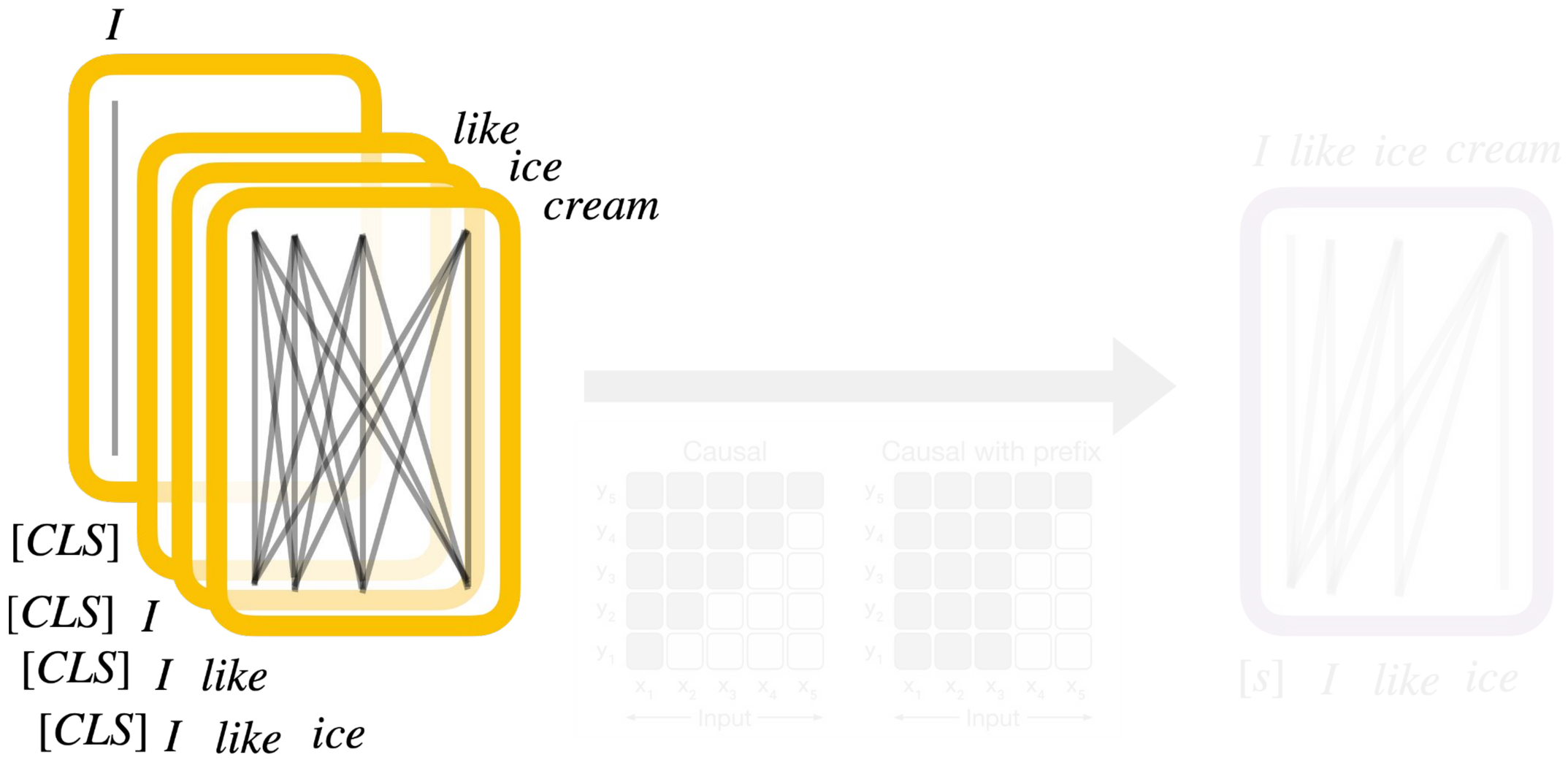
Predict ?->1 To Forecast n Predictions





Decoder Training Parallelization

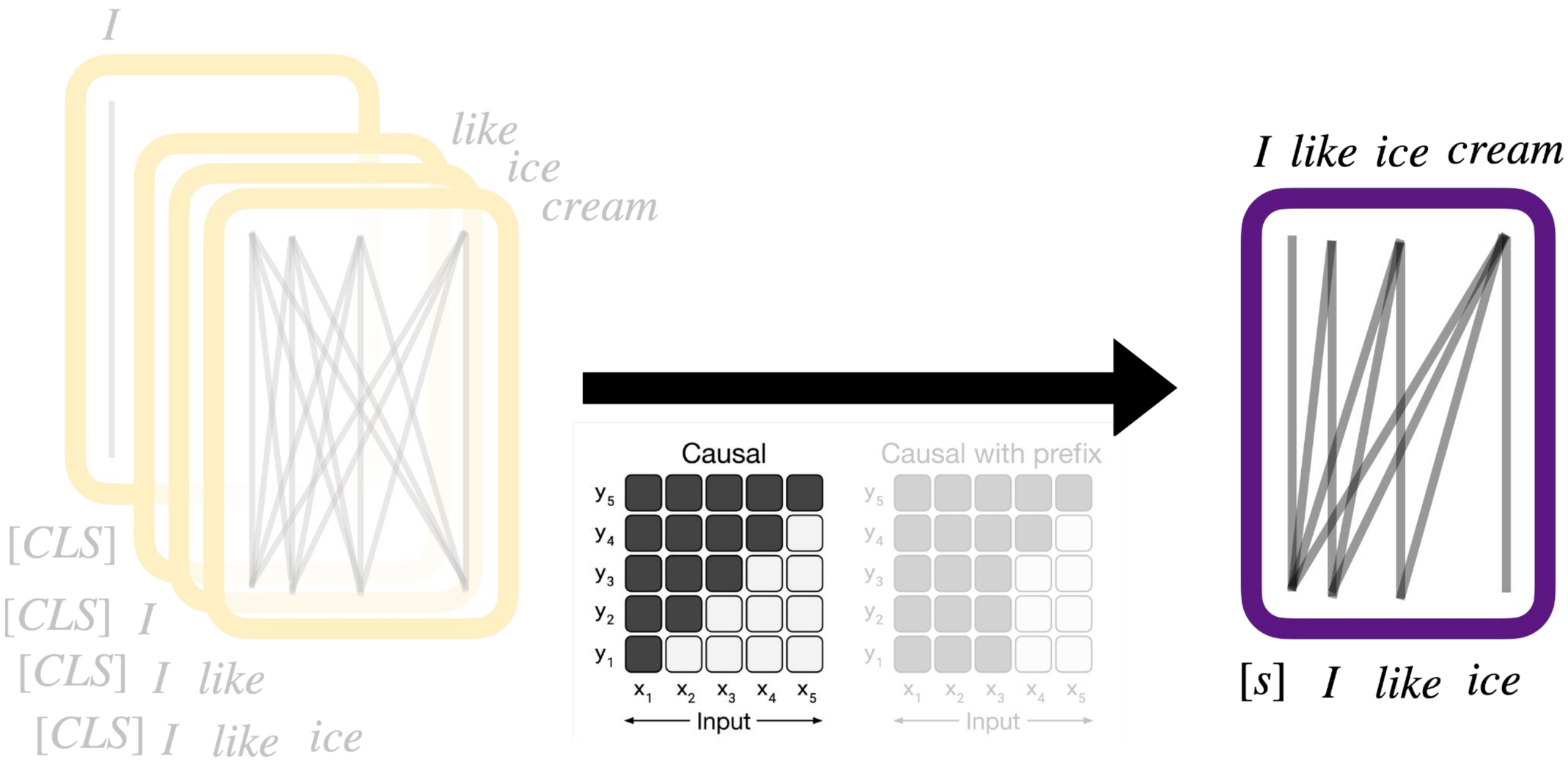
Why Unidirectional Reasoning Is Standard



https://arxiv.org/abs/1910.10683v4

Decoder Training Parallelization

Why Unidirectional Reasoning Is Standard



https://arxiv.org/abs/1910.10683v4

Context Limits

Why Autoregressing Indefinitely Is Not Great

I like ice cream and I like cookies and ⋯ I like | cupcakes and ⋯

Like Training

Out Of Distribution

[1023]

[0]

Encoders vs Decoders

Both Options Have Pros and Cons

Encoder

Core Value Prop

Strongly enforces per-token logic

Relies on bidirectional reasoning

Powerful full-context encodings

Embedding

Decoder

Core Value Prop

Used to generate novel entries.

Needs very strong one-directional reasoning for good results.

Forecasting

Encoders vs Decoders

Both Options Have Pros and Cons

Encoder

Relies on bidirectional reasoning

Powerful full-context encodings

Embedding

Decoder

Needs very strong one-directional reasoning for good results.

Forecasting

Both Have Limitations

Why Not Combine Them?

Encoder-Decoder

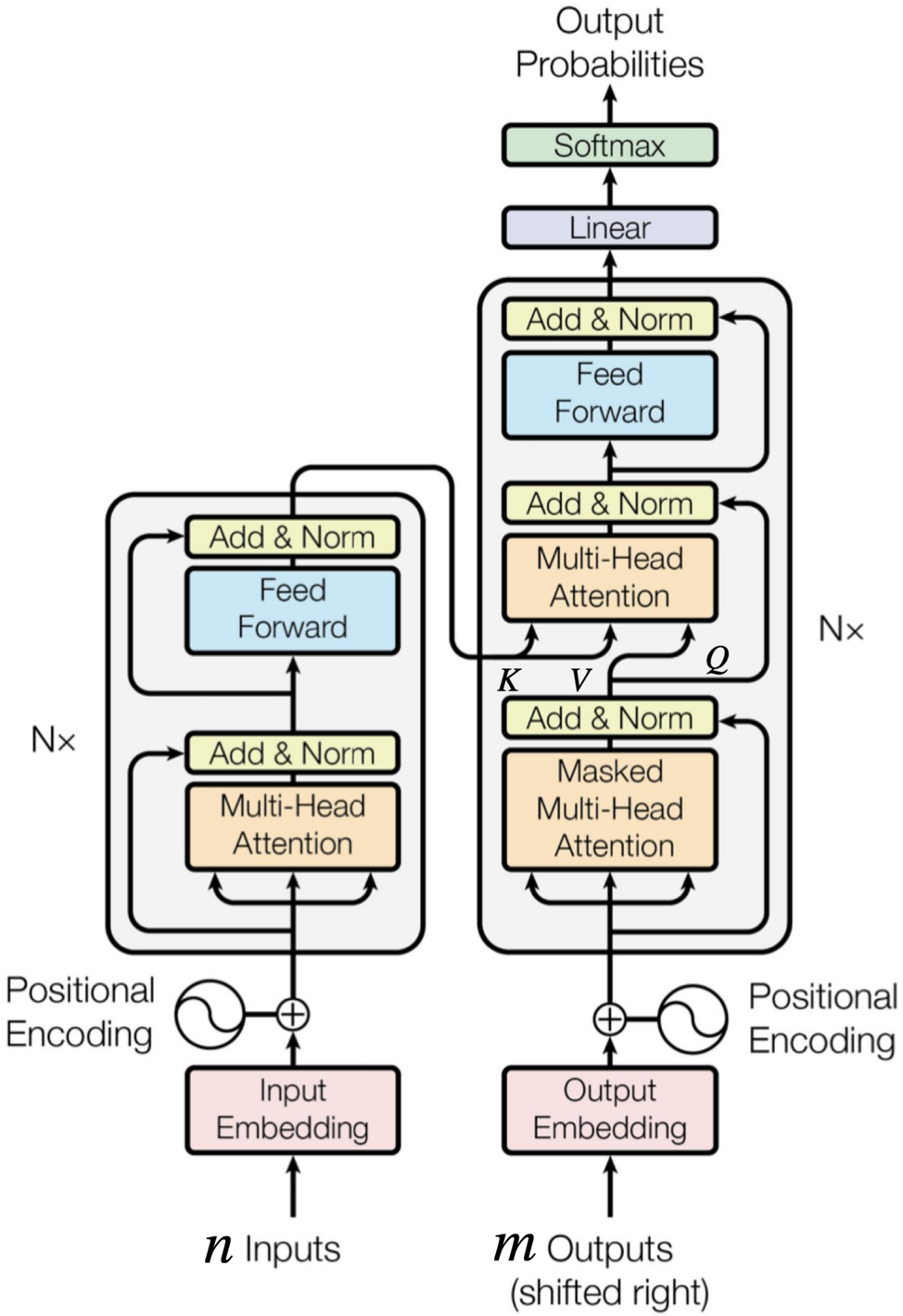
Best of Both Worlds (Possibly)

Attention

Value

$$\frac{QK^T}{\sqrt{d_k}}$$

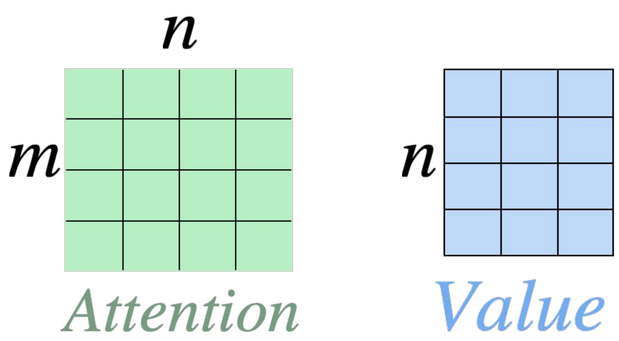
$$\sigma(A)V$$



https://arxiv.org/pdf/1706.03762.pdf

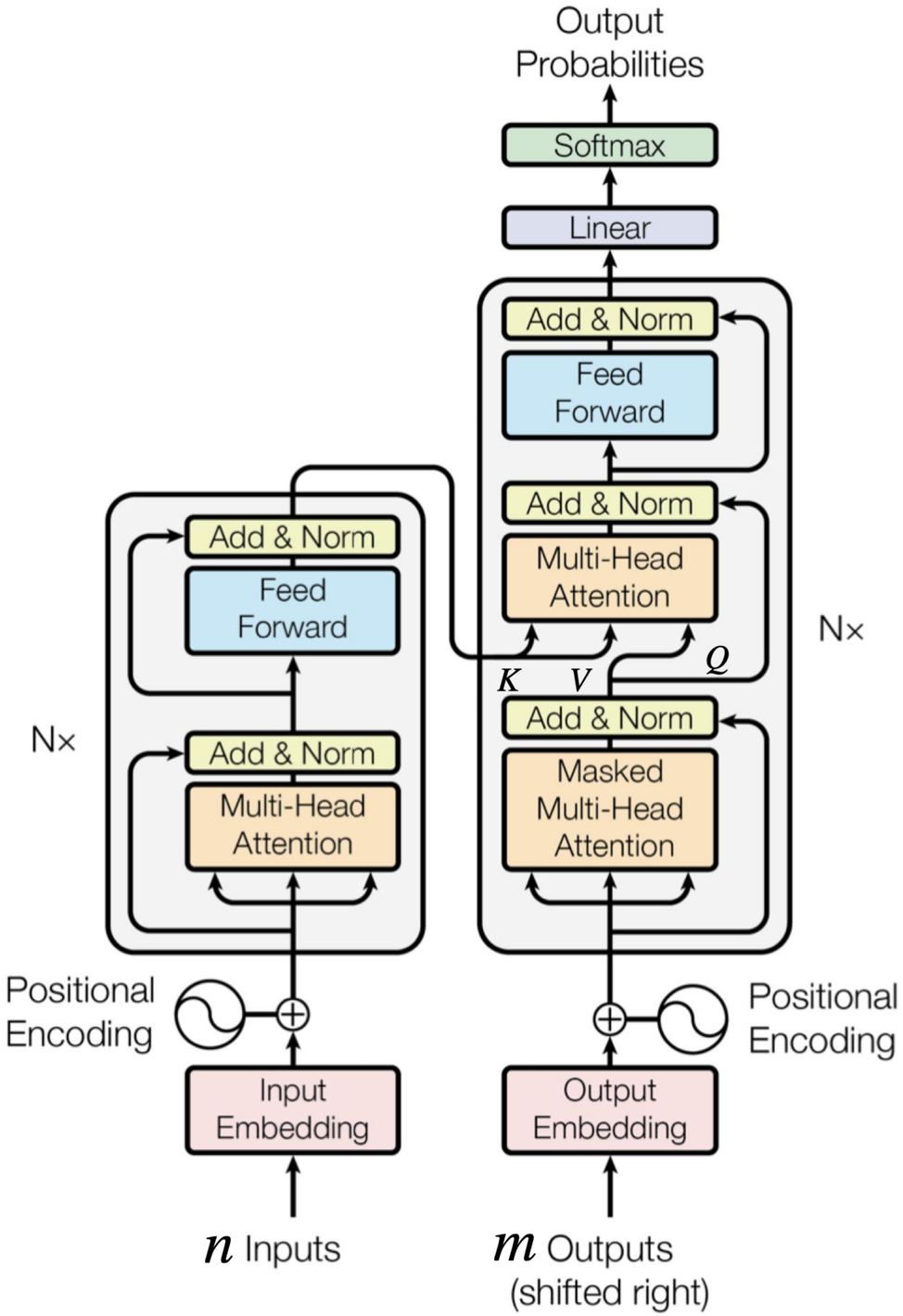
Encoder-Decoder

Best of Both Worlds (Possibly)



$$\frac{QK^T}{\sqrt{d_k}}$$

$$\sigma(A)V$$



https://arxiv.org/pdf/1706.03762.pdf

Encoder-Decoder

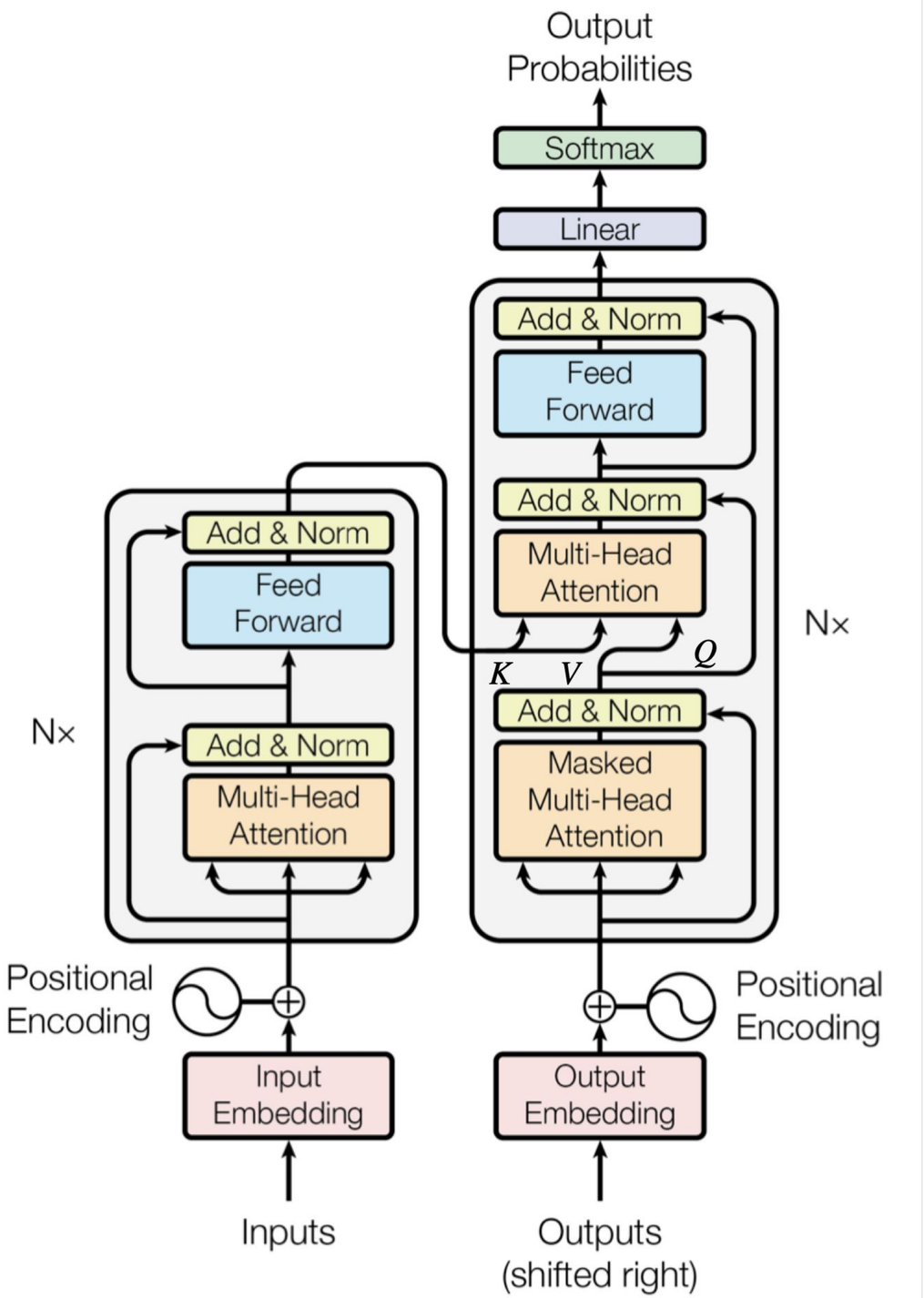
Best of Both Worlds (Possibly)

Approach Overview

- Bidirectional reasoning w/ encoder
- Help decoder by supplying embedding as context
- Generate response w/ context-primed decoder

Key Benefits

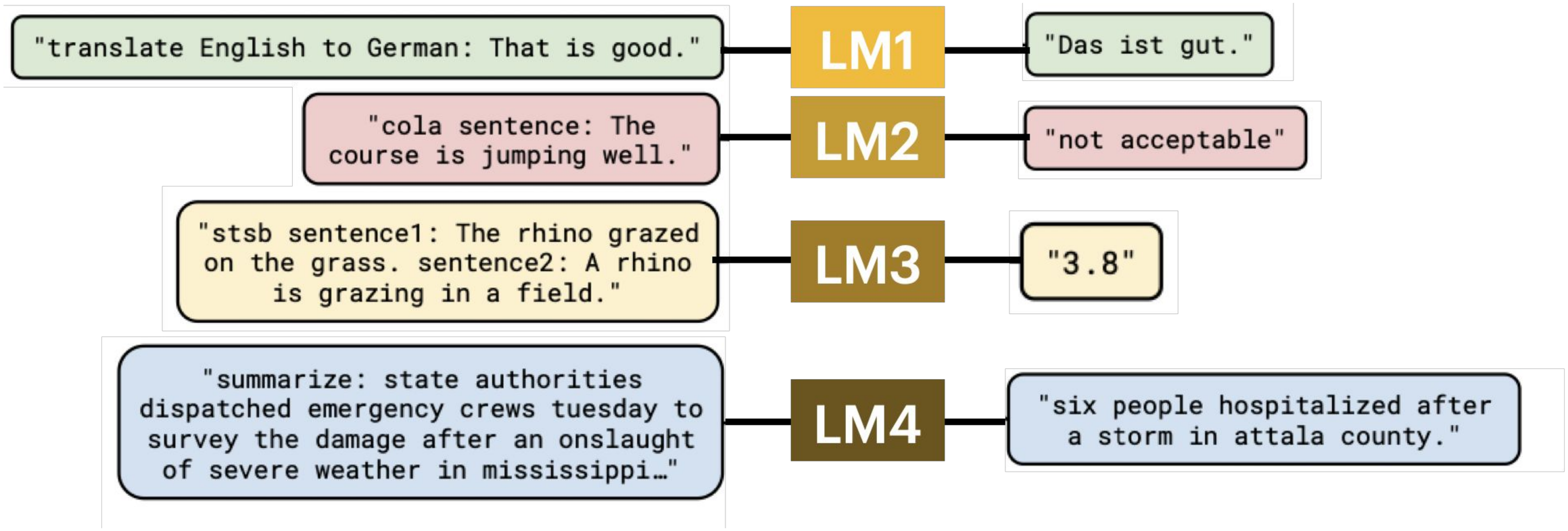
- Leverages ease of training encoders
- Grounds harder-to-train decoder with context
- Leverages decoder's forecasting formulation



https://arxiv.org/pdf/1706.03762.pdf

Single-Task Seq2Seq

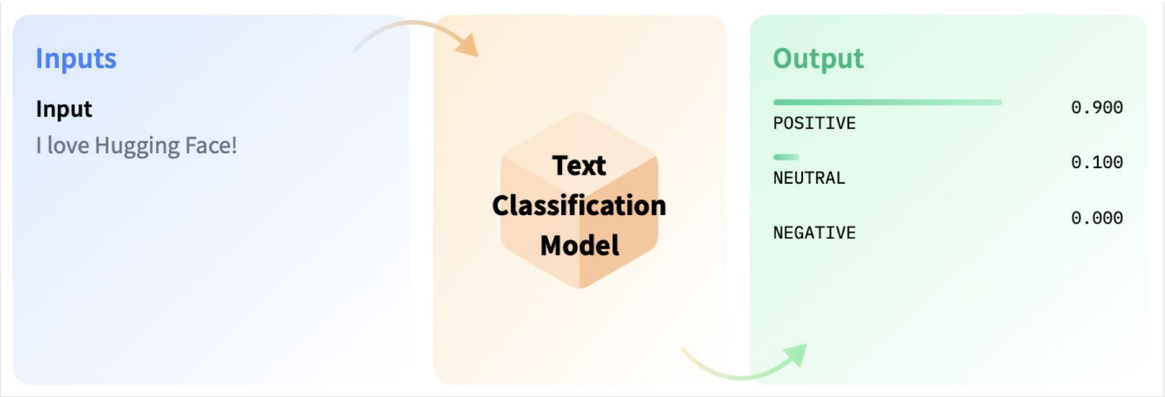
Arbitrary Translation

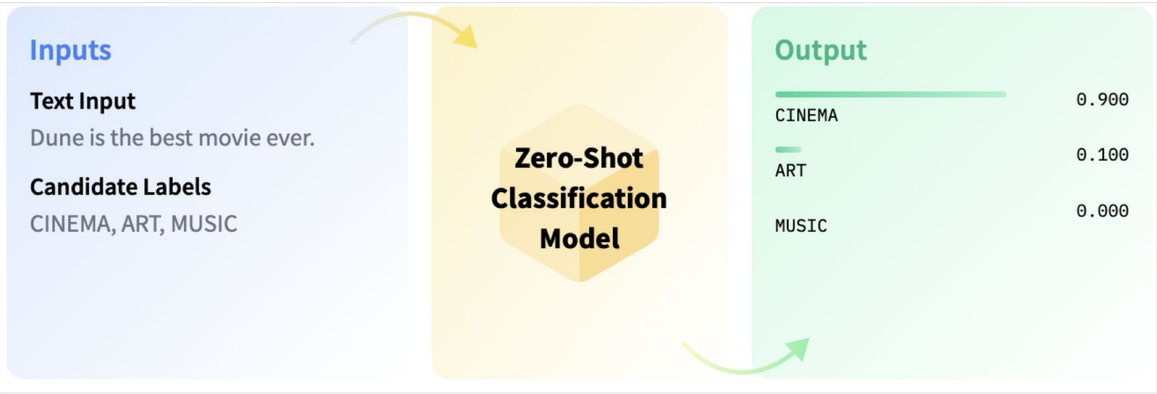


https://arxiv.org/pdf/1910.10683v3.pdf

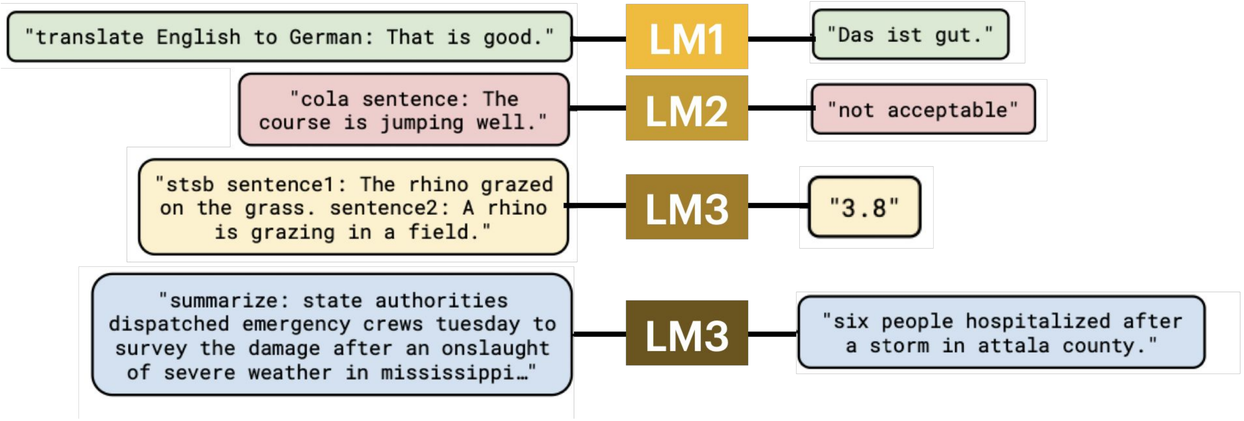
Progressing Towards Zero-Shot

Understanding Tasks Without Explicitly Training To Them

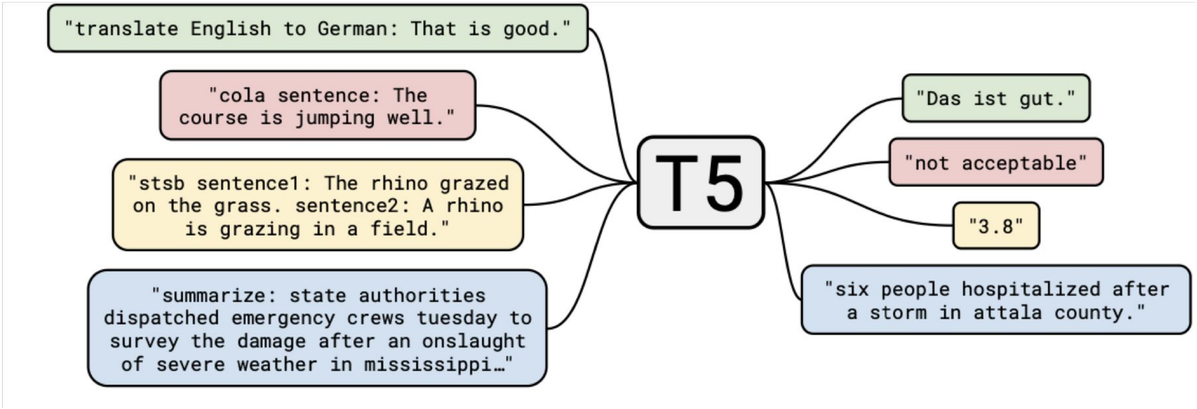




Training For My Objectives



Training For All Objectives



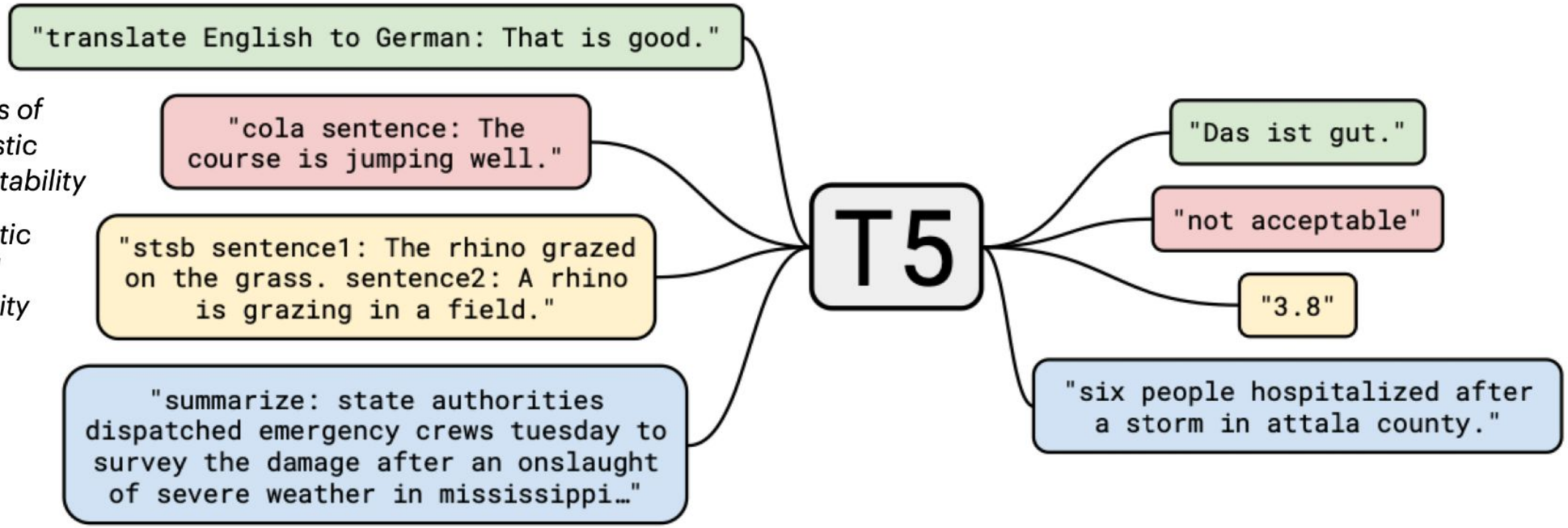
Question-Understanding Seq2Seq

Arbitrary Translation

Support for tasks that help build up logic

corpus of linguistic acceptability

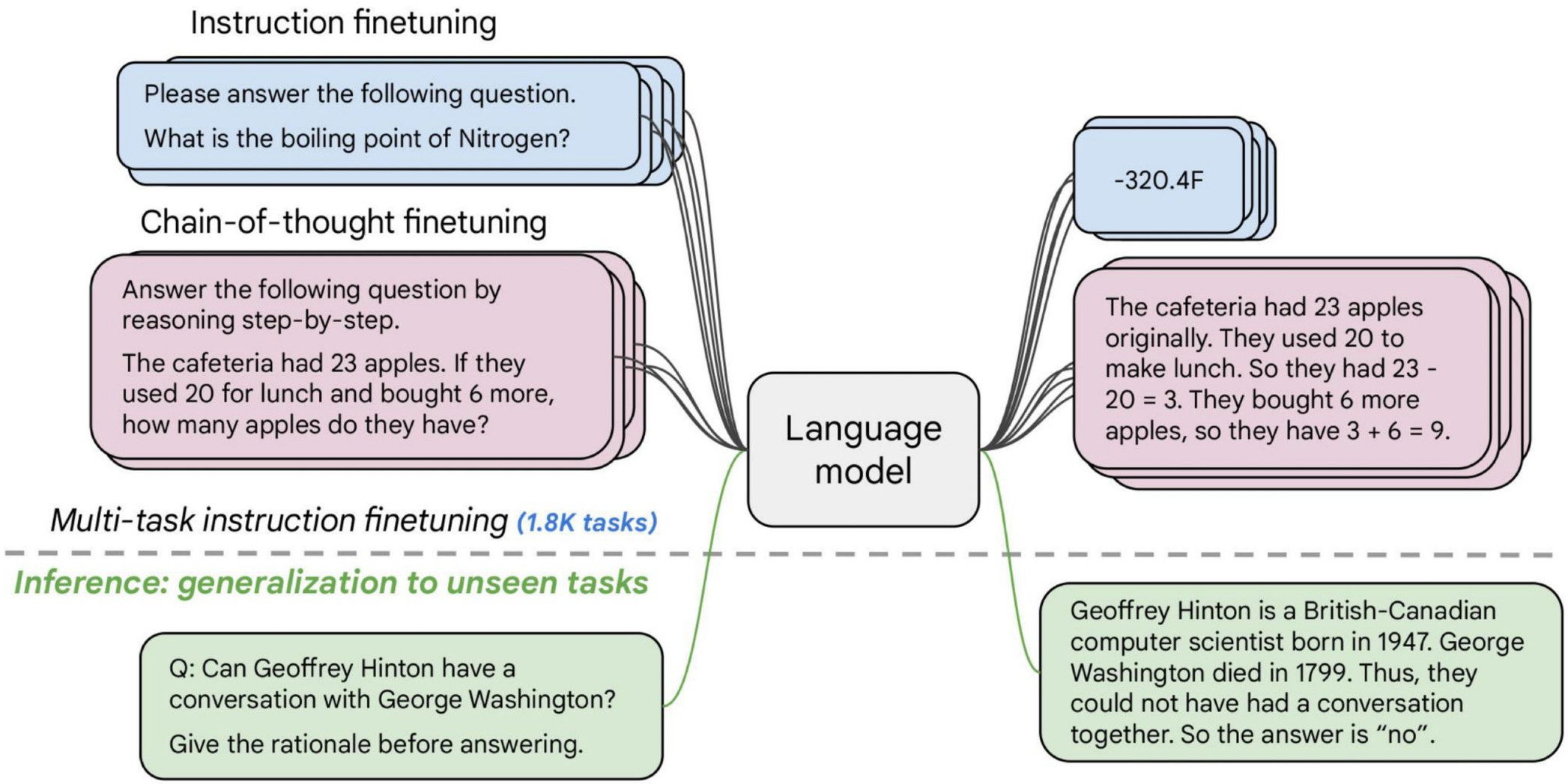
semantic textual similarity



https://arxiv.org/pdf/1910.10683v3.pdf

Instruction-Following Seq2Seq

Arbitrary Translation



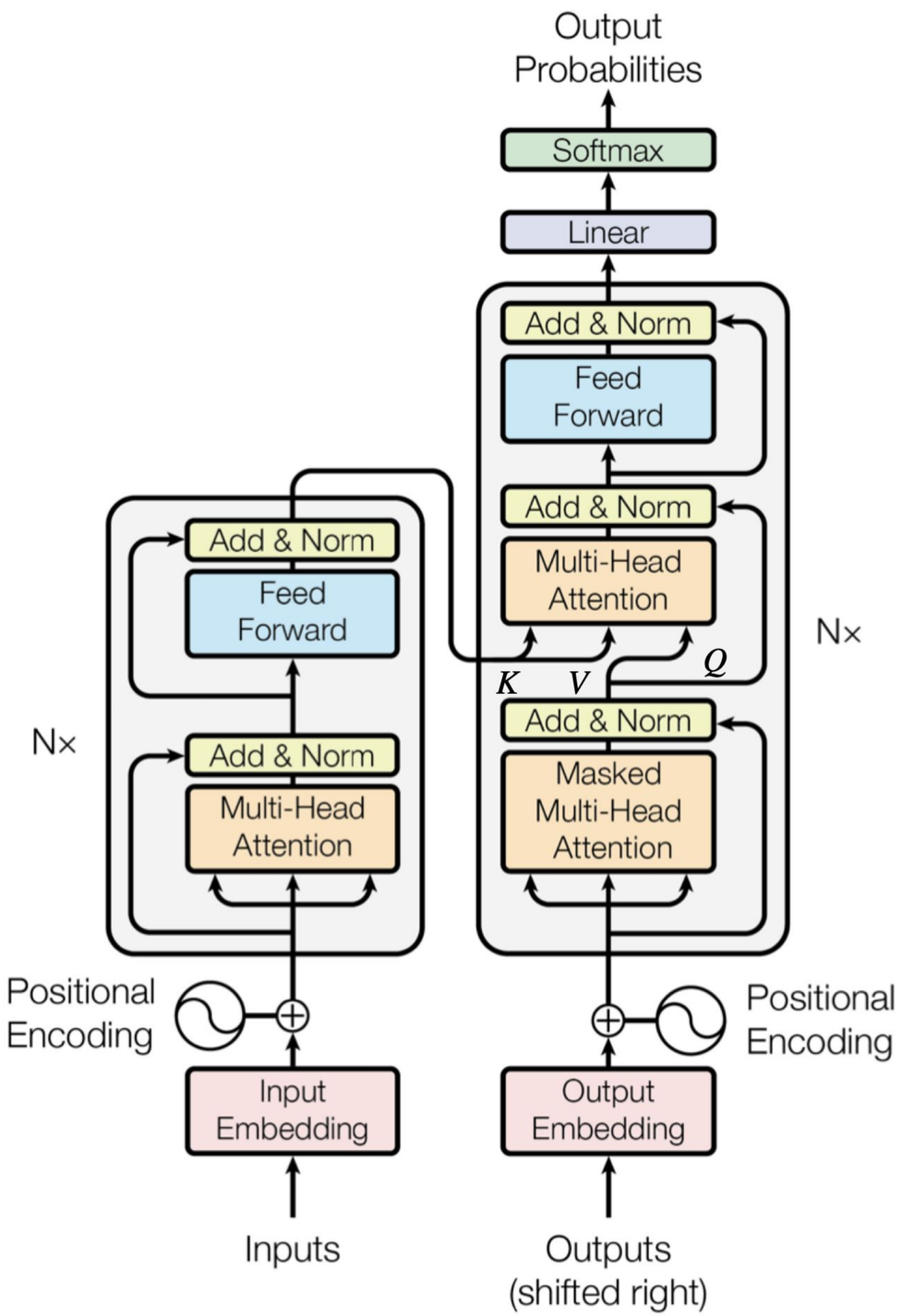
https://arxiv.org/pdf/2210.11416.pdf

Encoder-Decoder Benefits

When T5-Style Seq2Seq Shine

Key Features: Powerful and Efficient

- Grounds harder-to-train decoder with context
  - Static context also allows KV caching
- Leverages decoder's forecasting formulation



Encoder-Decoder Benefits

When T5-Style Seq2Seq Shine

Key Features: Powerful and Efficient

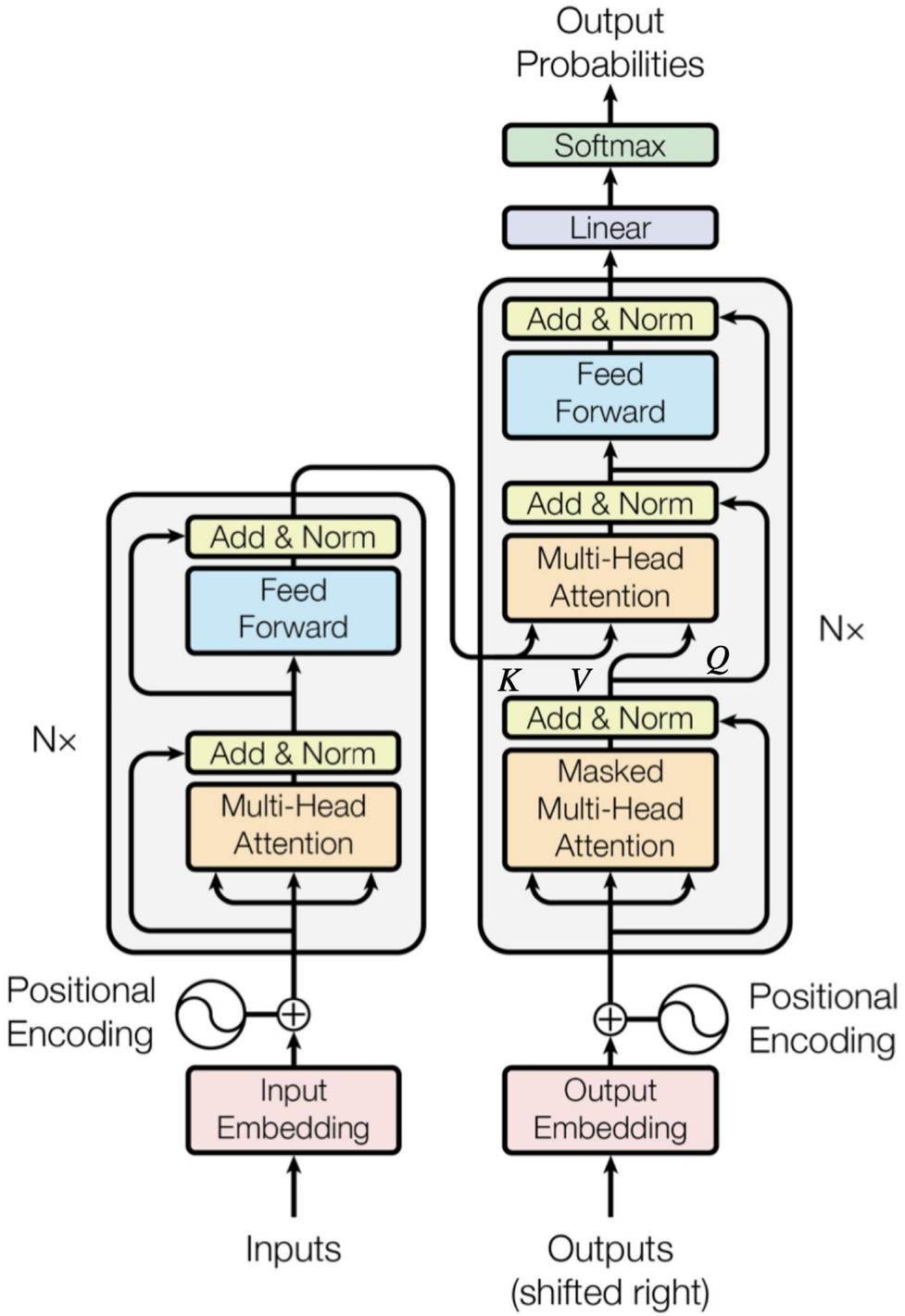
- Grounds harder-to-train decoder with context
  - Static context also allows KV caching
- Leverages decoder's forecasting formulation

BEST WHEN

- Context and response are strongly separated

BUT NOT IDEAL WHEN

- There is little-to-no context/response division



Part 2: Decoding Models

- Encoder-Decoders for Seq2Seq
- Cross-Attention for Multimodal Models
- Large Decoder Models for Text Gen

Transformer Benefits

What Are They Good For

Misconception:

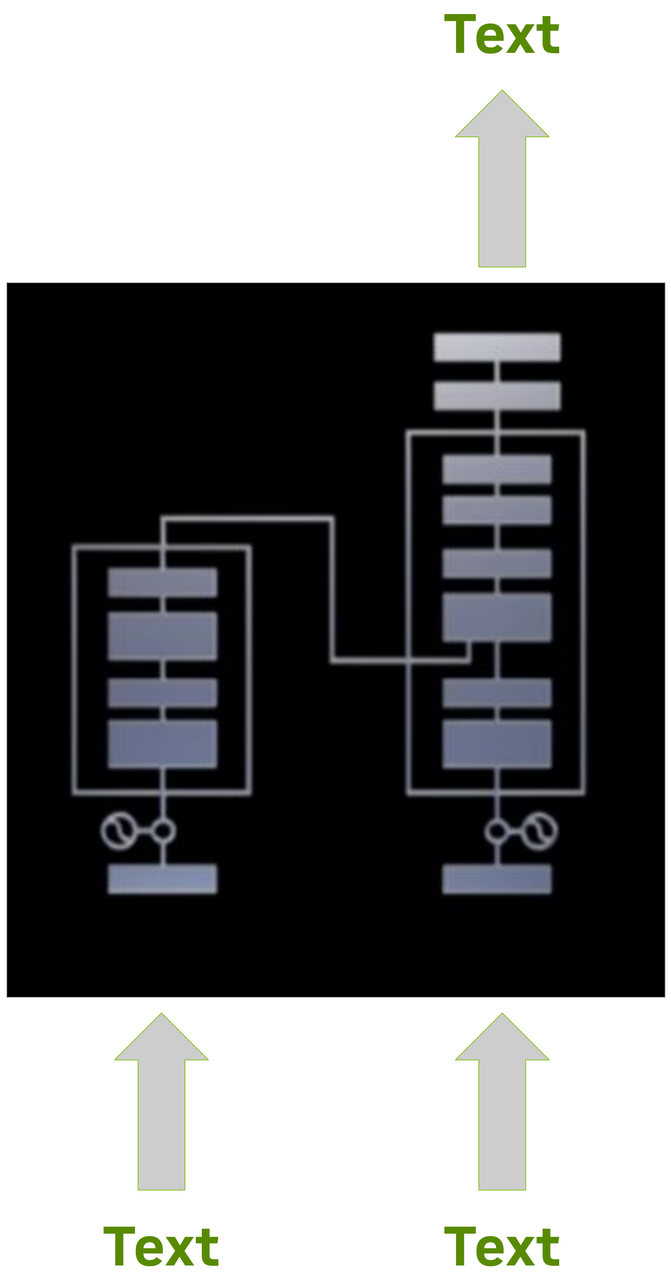
Transformers are only good for language

Recall:

Language Text = Ordered Sequence of Classes

Resolution:

Transformers are good for ordered sequences



Transformer Benefits

What Are They Good For

Misconception:

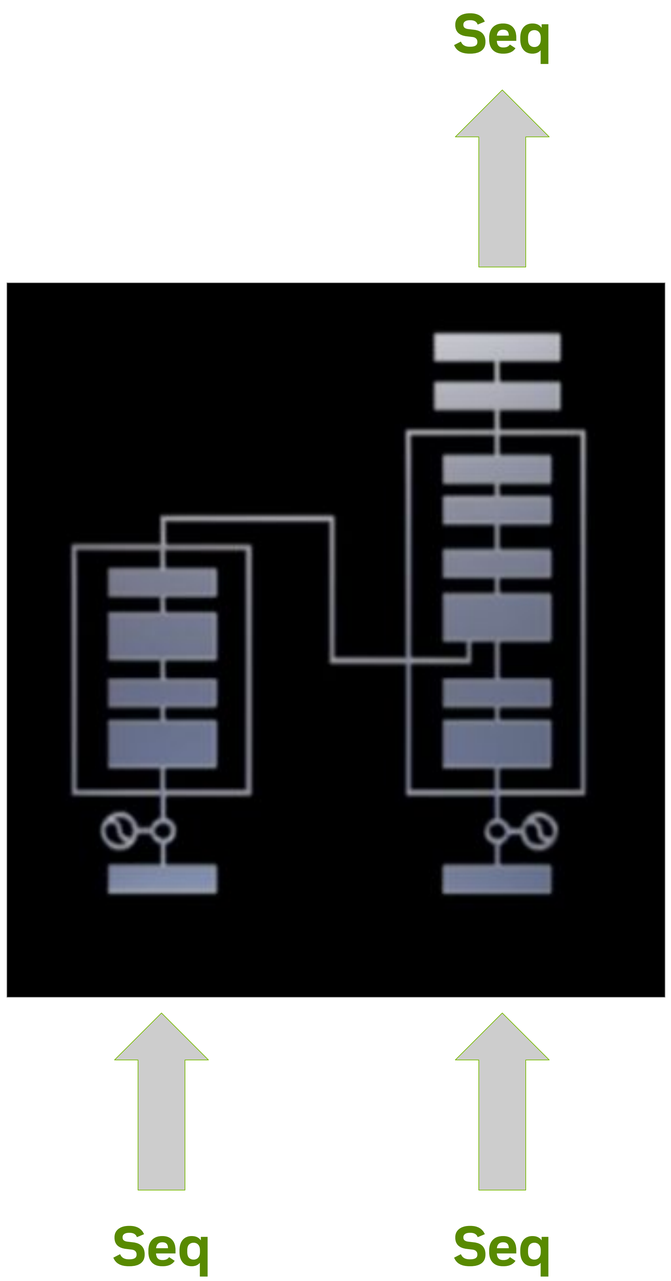
Transformers are only good for language

Recall:

Language Text = Ordered Sequence of Classes

Resolution:

Transformers are good for ordered sequences



Tabular Encoders

Just Beyond Language

Inputs

Rank	Name	No.of reig
1	lou Thesz	3
2	Ric Flair	8
3	Harley Race	7

Question

What is the number of reigns for Harley Race?

Table Question Answering Model

Output

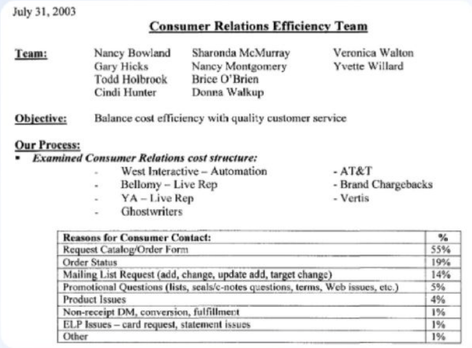
Result

7

Inputs

Question

What is the idea behind the consumer relations efficiency team?



July 31, 2003

Consumer Relations Efficiency Team

Team: Nancy Bowland, Gary Hicks, Todd Holbrook, Cindi Hunter, Sharonda McMurray, Nancy Montgomery, Brice O'Brien, Donna Walkup, Veronica Walton, Yvette Willard

Objective: Balance cost efficiency with quality customer service

Our Process:

- *Examined Consumer Relations cost structure:*
  - West Interactive – Automation
  - Bellomy – Live Rep
  - YA – Live Rep
  - Ghostwriters
  - AT&T
  - Brand Chargebacks
  - Vertis

Reasons for Consumer Contact:	%
Request Catalog/Order Form	55%
Order Status	19%
Mailing List Request (add, change, update add, target change)	14%
Promotional Questions (lists, seals/e-notes questions, terms, Web issues, etc.)	5%
Product Issues	4%
Non-receipt DM, conversion, fulfillment	1%
ELP Issues – card request, statement issues	1%
Other	1%

Document Question Answering Model

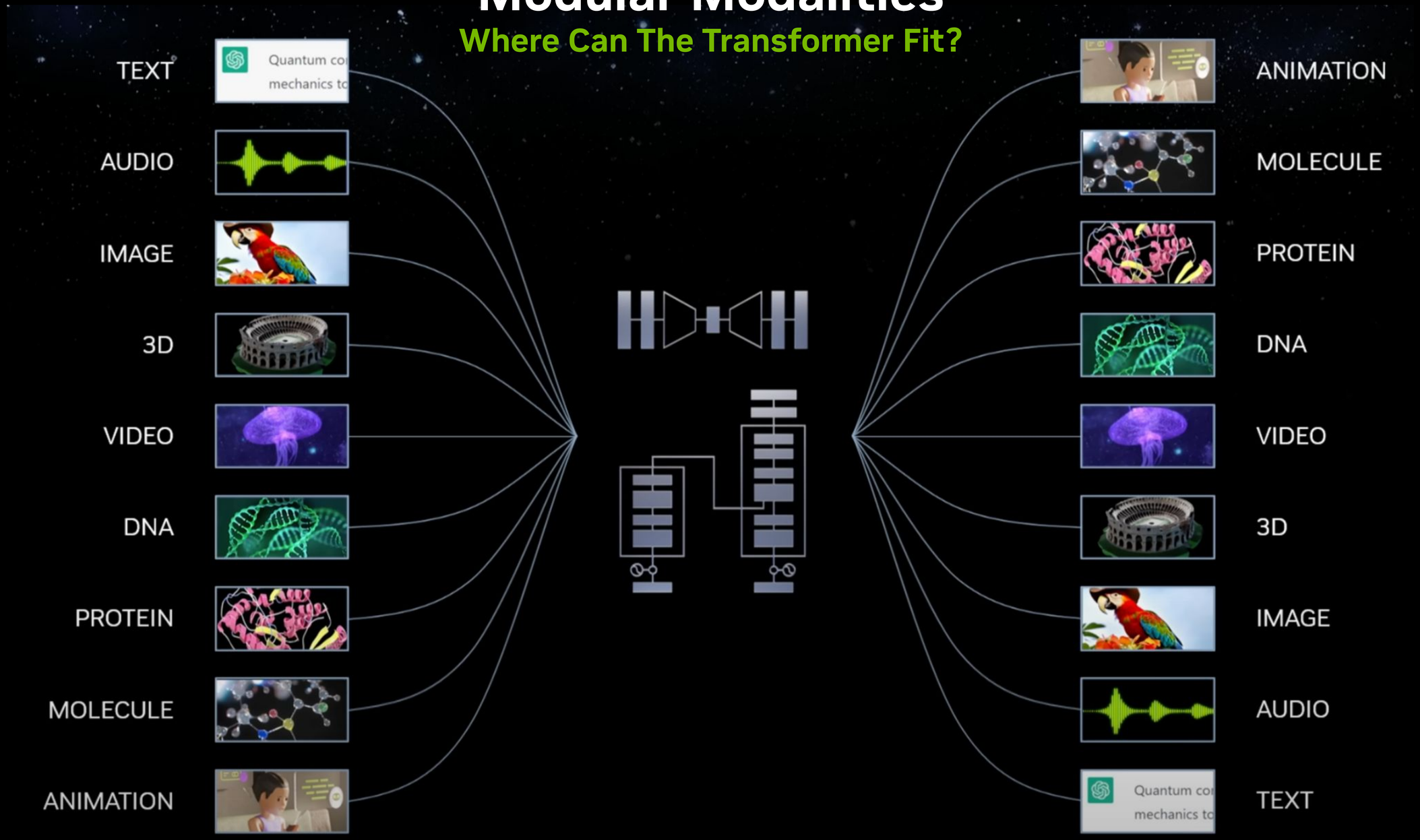
Output

Answer

Balance cost efficiency with quality customer service

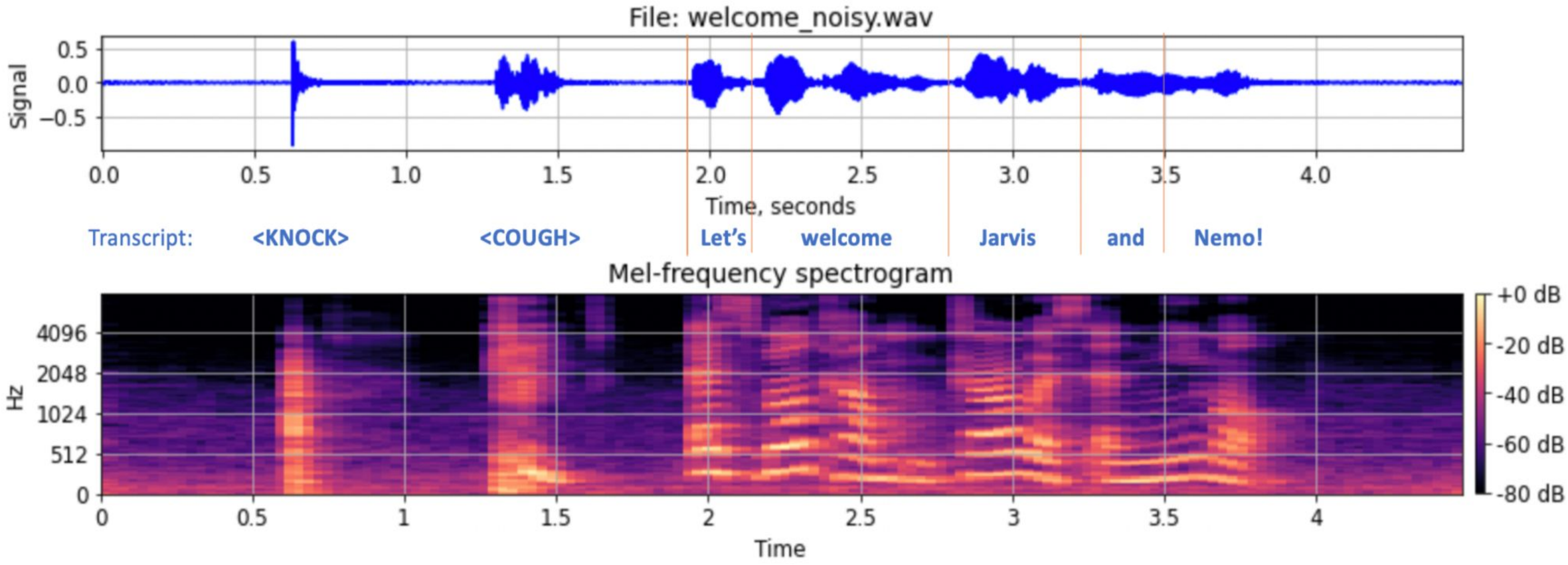
Modular Modalities

Where Can The Transformer Fit?



Speech-Guided Encoders

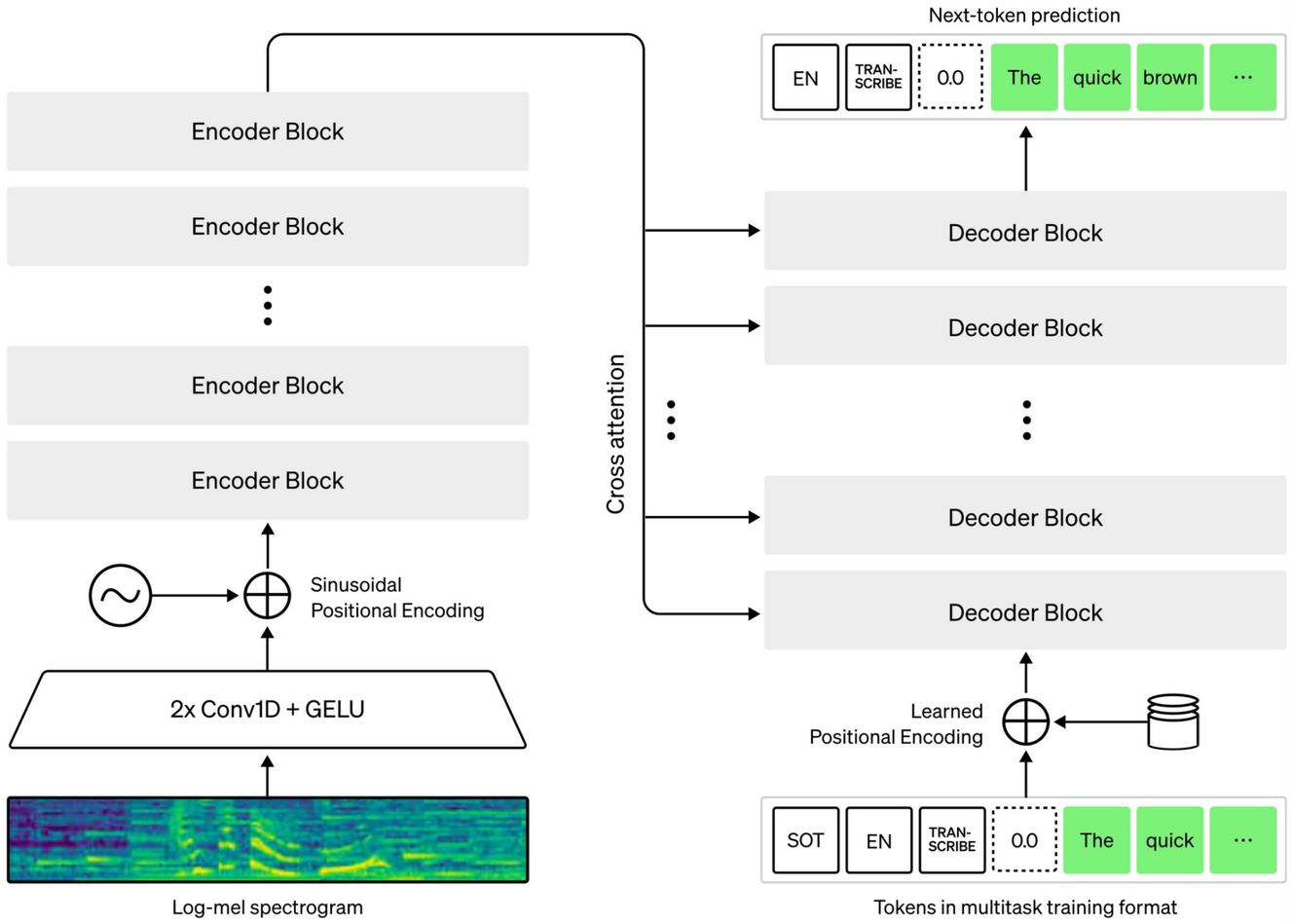
Audio-In Systems

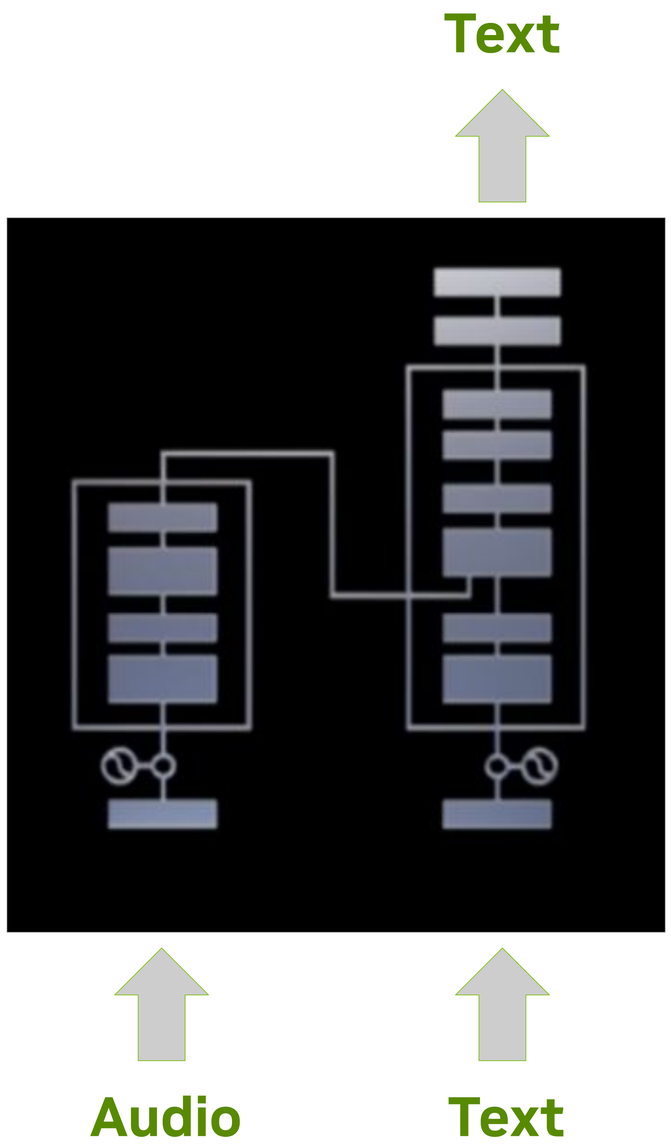


https://docs.nvidia.com/deeplearning/nemo/user-guide/docs/zh/latest/voice_activity_detection/tutorial.html

Speech-Guided Text Generation

Audio-In, Text-Out

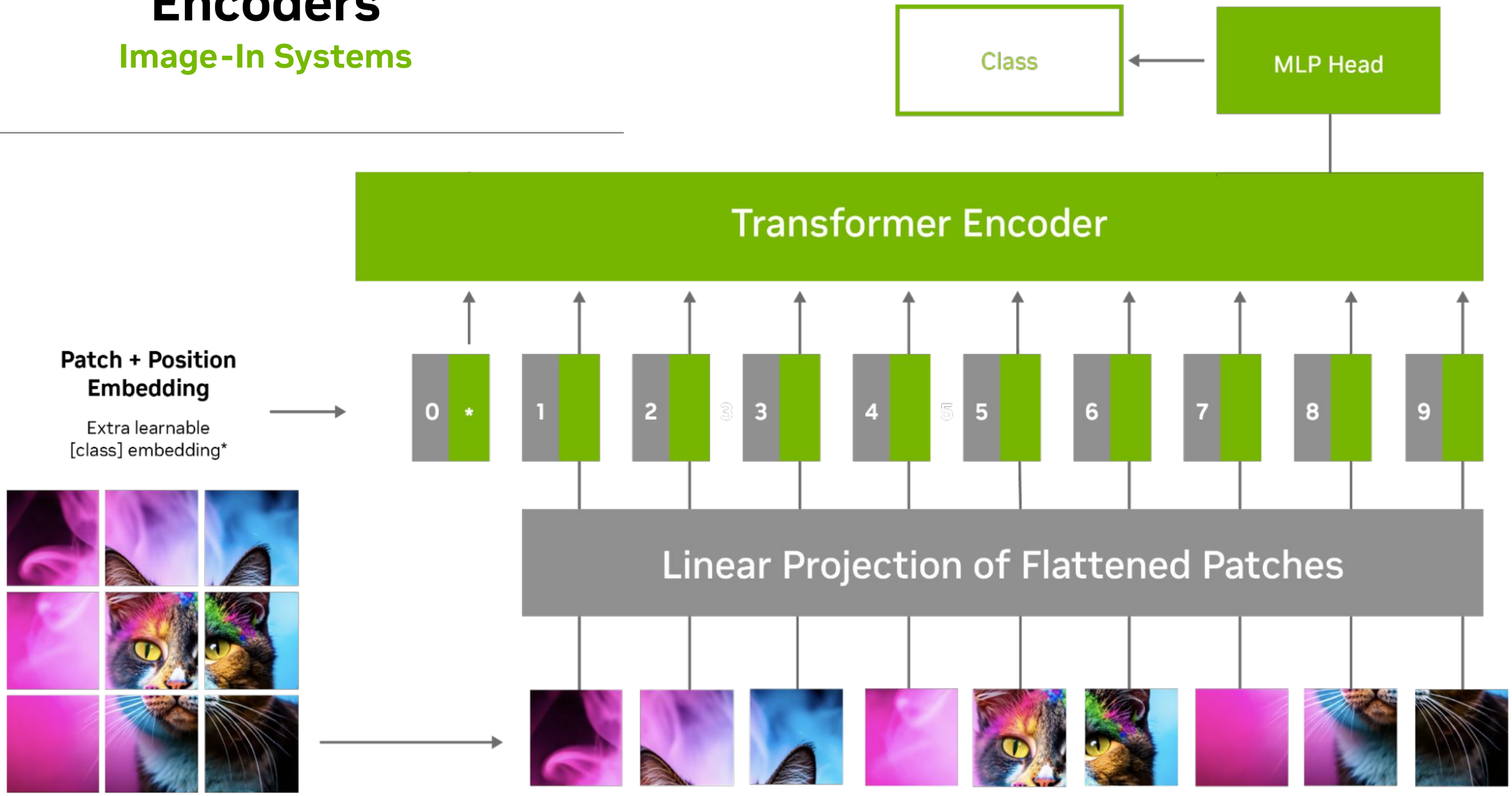




https://openai.com/research/whisper

Image-Guided Encoders

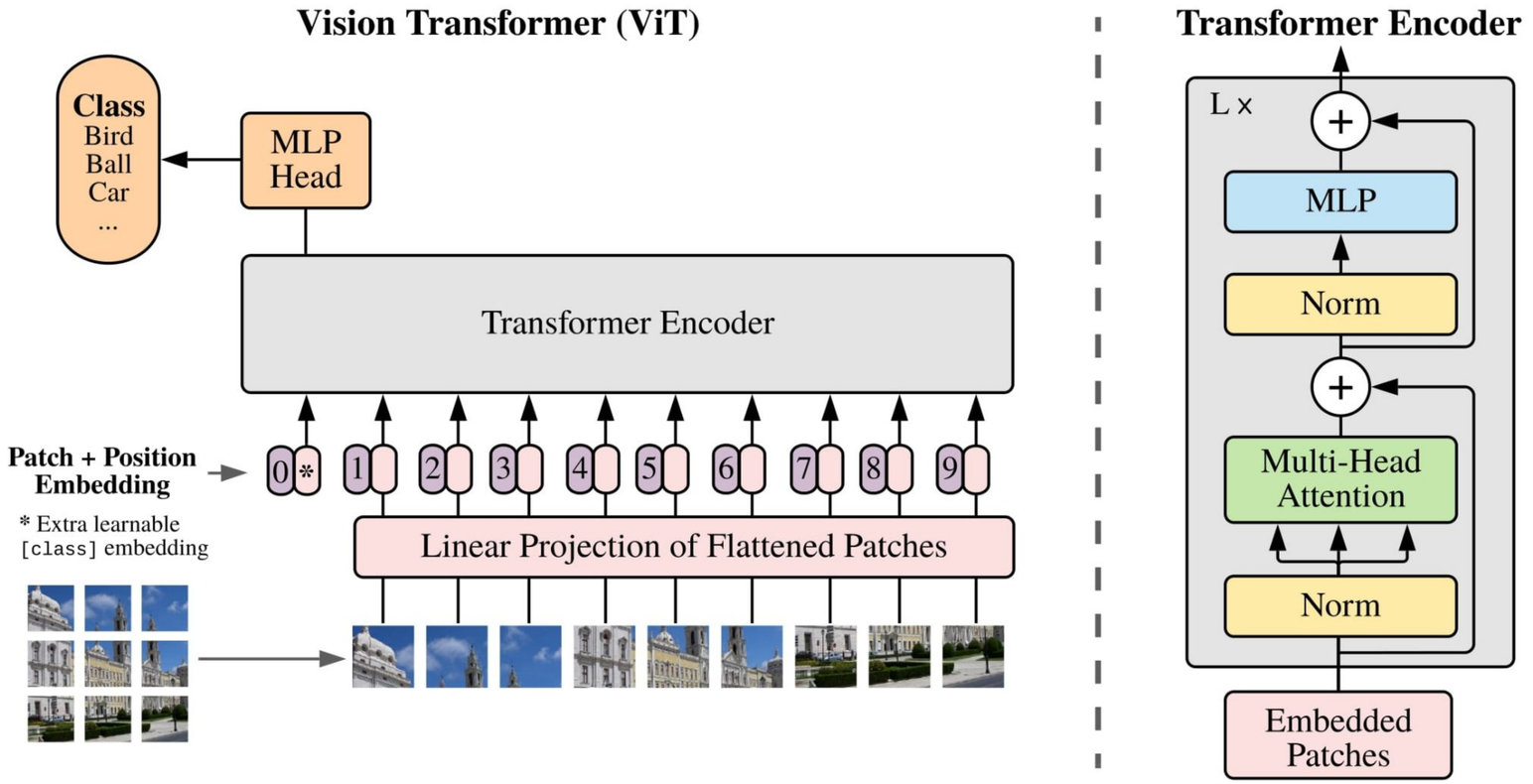
Image-In Systems

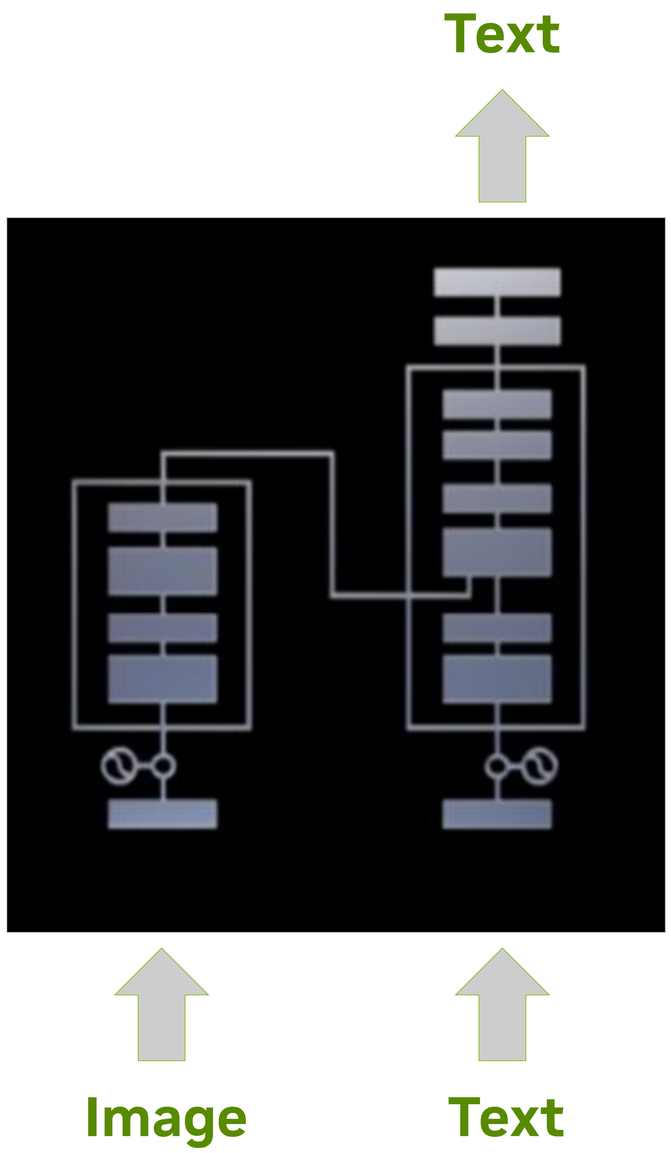


https://developer.nvidia.com/blog/improve-accuracy-and-robustness-of-vision-ai-apps-with-vision-transformers-and-nvidia-tao/

Image-Guided Text Generation

Image-In, Text-Out

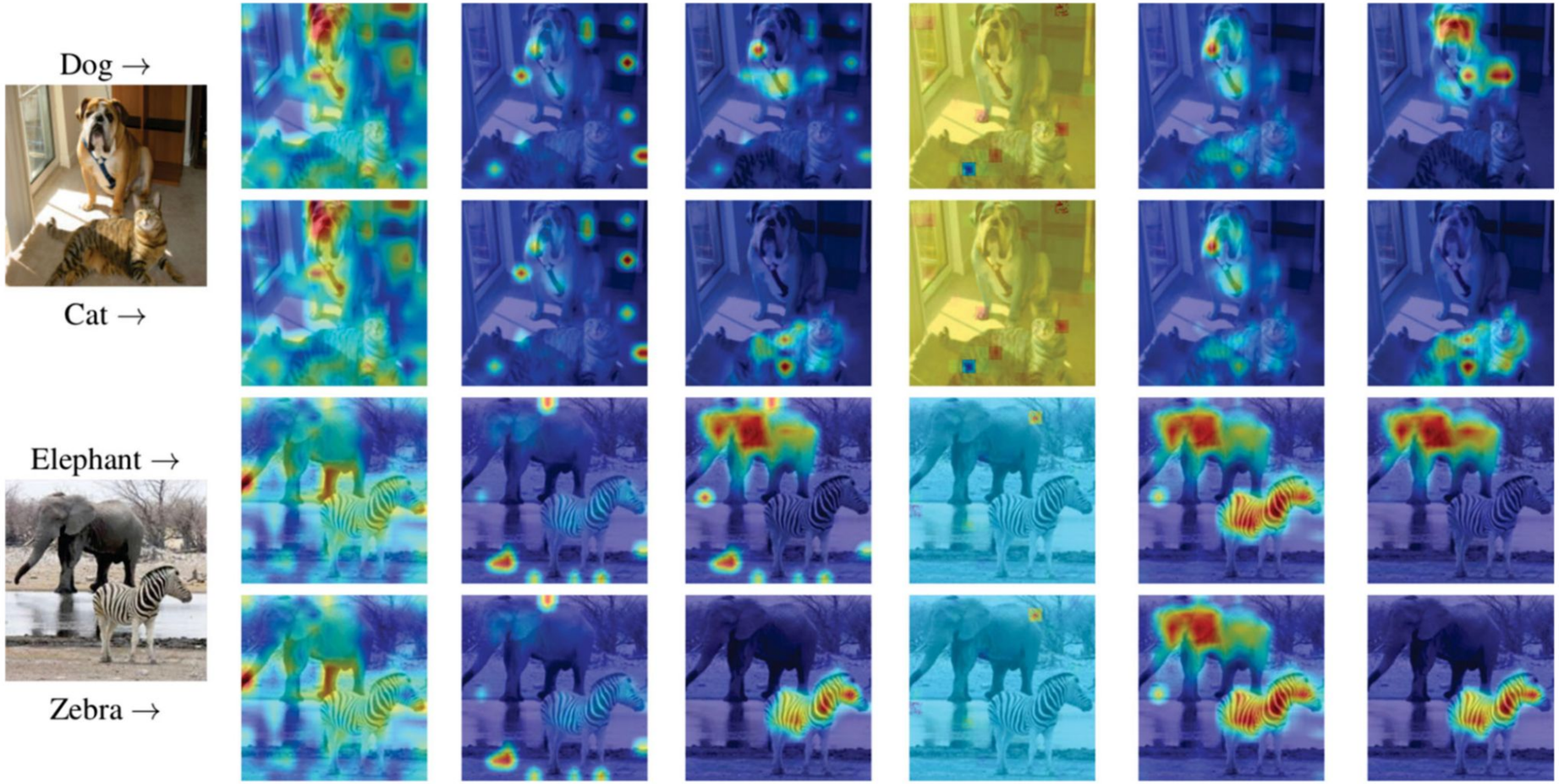




 https://www.researchgate.net/publication/355693348_Transformers_in_computational_visual_media_A_survey

Image Attention

Transformers With Images



 https://www.researchgate.net/publication/355693348_Transformers_in_computational_visual_media_A_survey

Encoders vs Decoders

Both Options Have Pros and Cons

Encoder

Core Value Prop

Strongly enforces per-token logic

Relies on bidirectional reasoning

Powerful full-context encodings

Embedding

Decoder

Core Value Prop

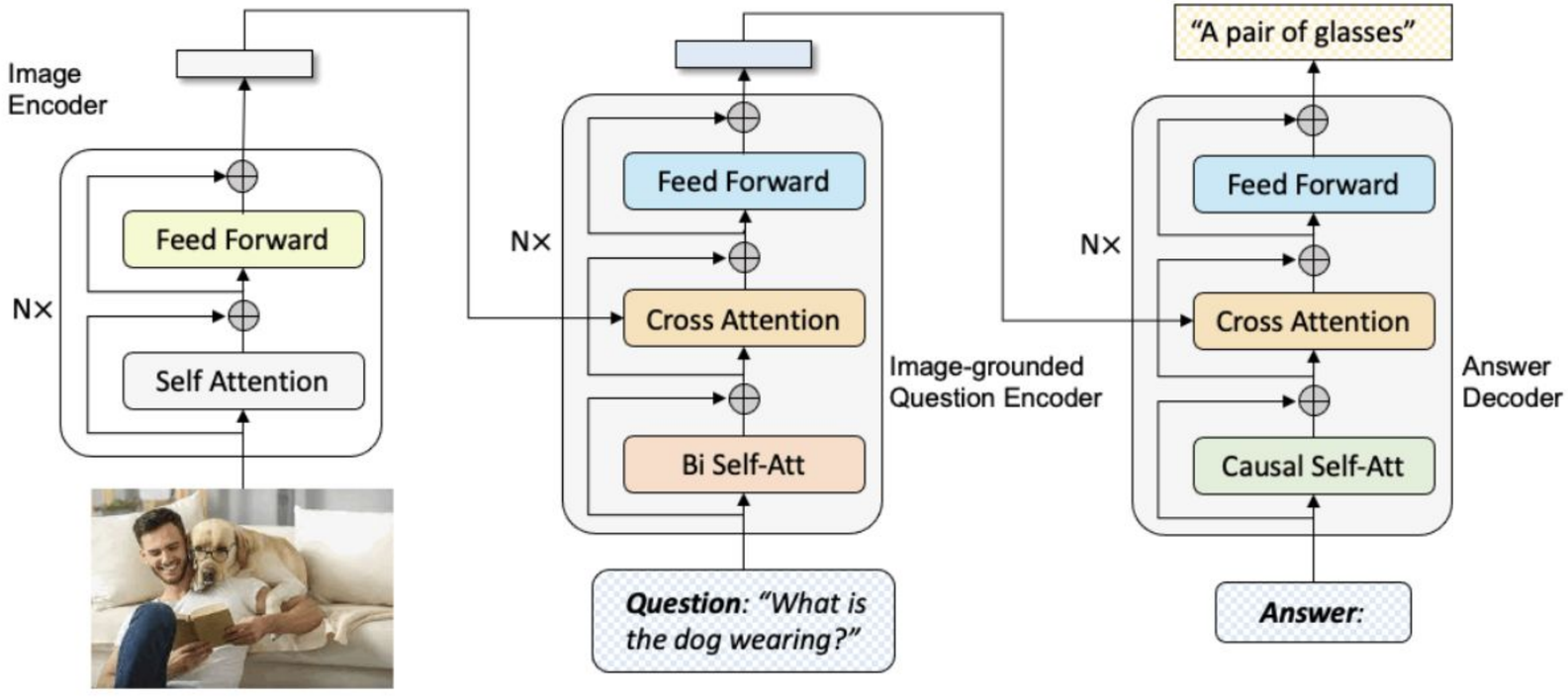
Used to generate novel entries.

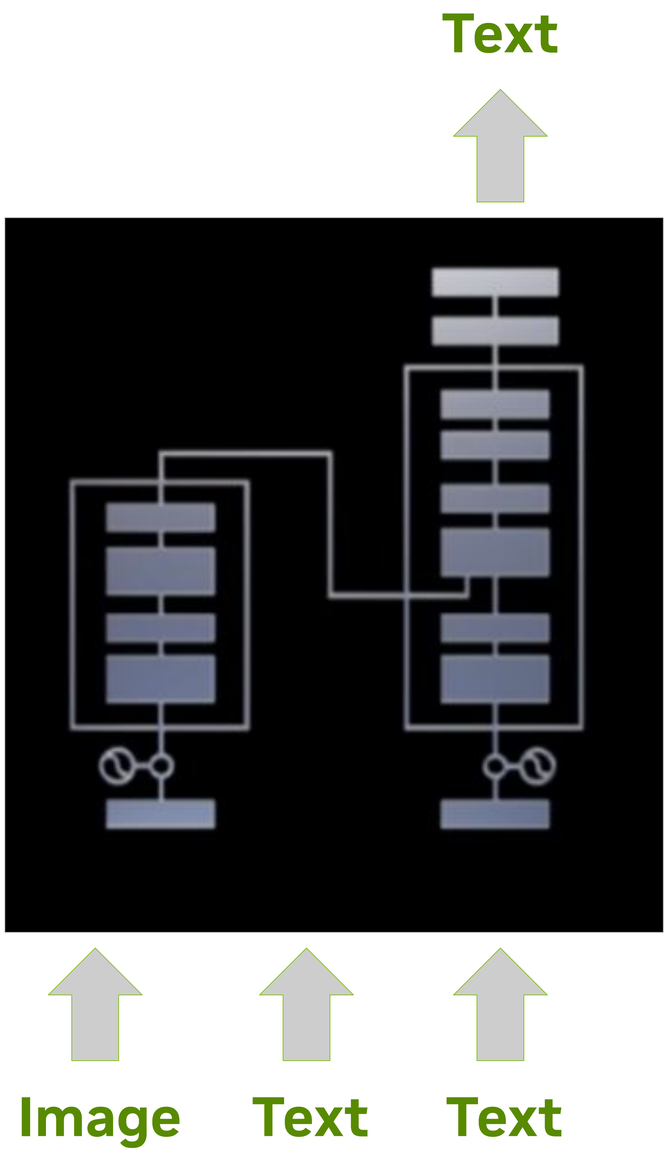
Needs very strong one-directional reasoning for good results.

Forecasting

More Image-Guided Text Generation

Image/Text-In, Text-Out





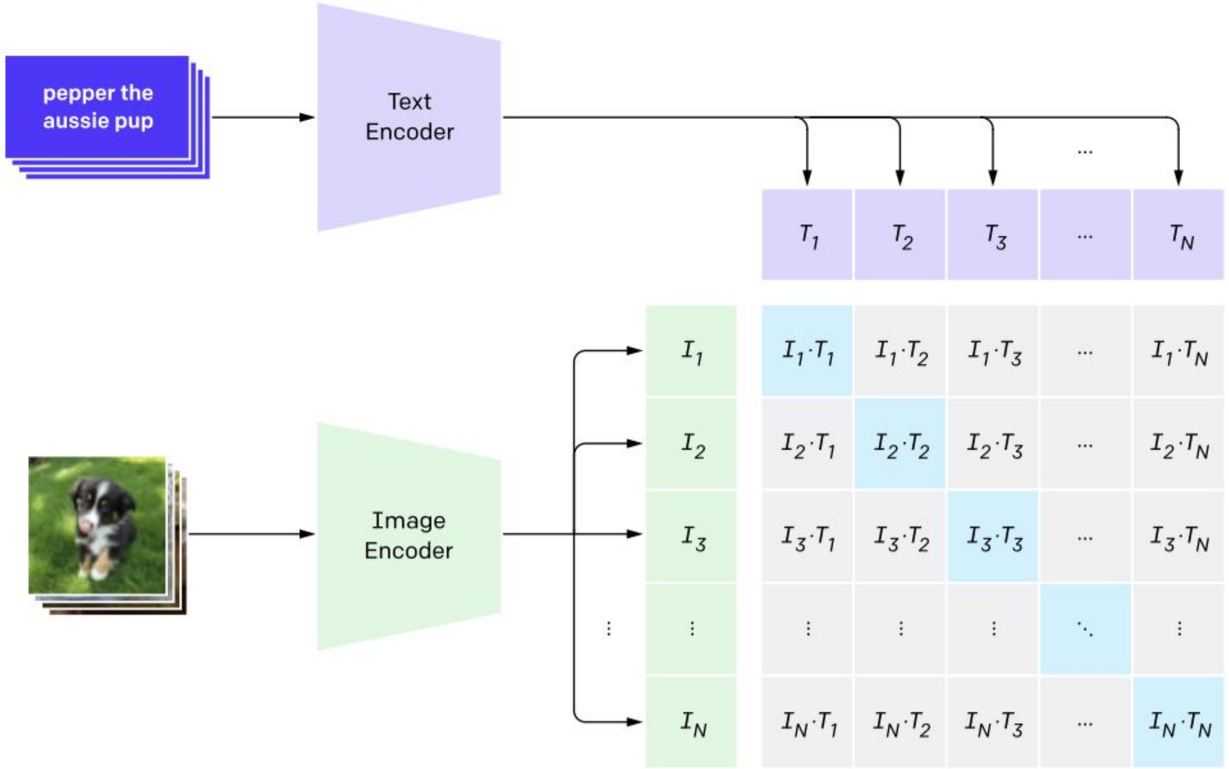
 https://github.com/salesforce/BLIP

Synergizing Encoders

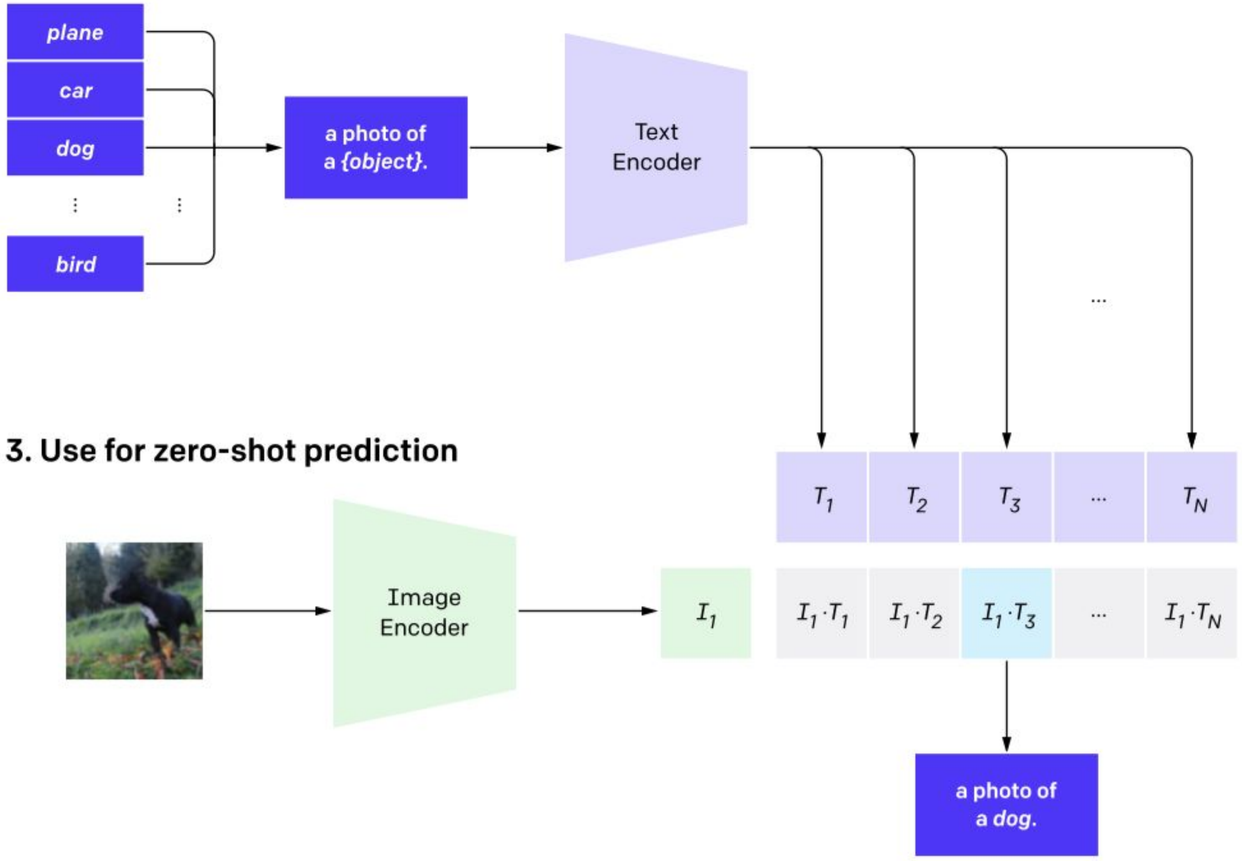
Resolving Modality Gaps via Similarity

CLIP pre-trains an image encoder and a text encoder to predict which images were paired with which texts in our dataset. We then use this behavior to turn CLIP into a zero-shot classifier. We convert all of a dataset's classes into captions such as "a photo of a dog" and predict the class of the caption CLIP estimates best pairs with a given image.

1. Contrastive pre-training



2. Create dataset classifier from label text

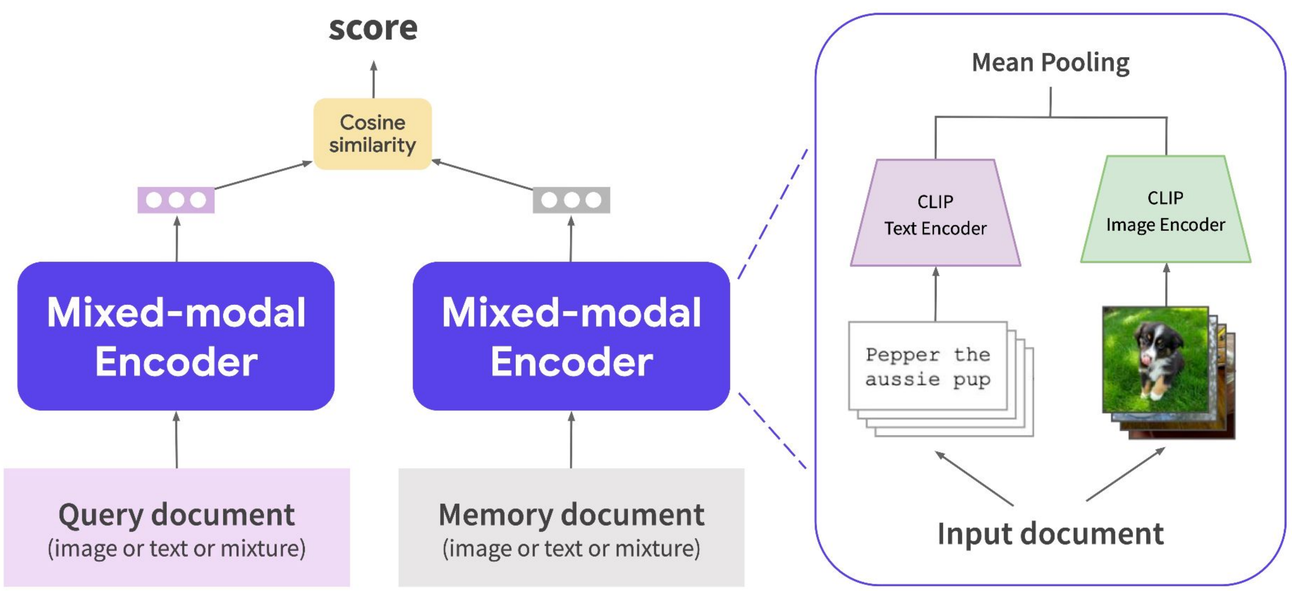


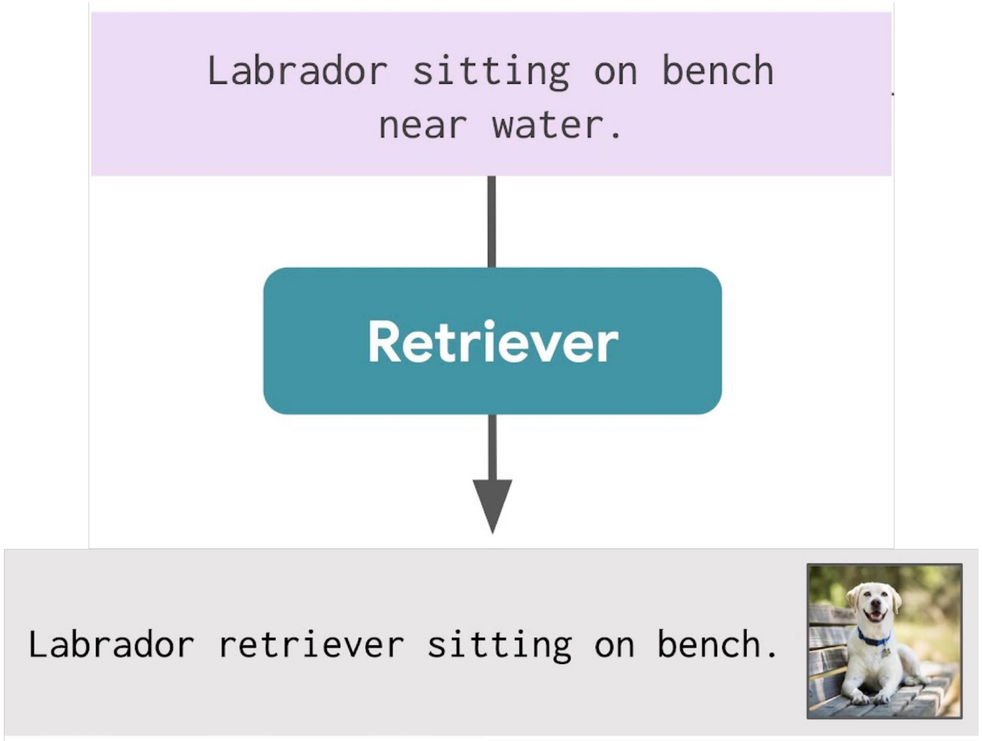
3. Use for zero-shot prediction

Multimodal Retrievers

Resolving Modality Gaps via Similarity

`f(query, memory) ⟶ score`





Multimodal document: image, text, or mixture of them

https://cs.stanford.edu/~myasu/blog/racm3/

Image Generation?

Can Transformers Generate Images?

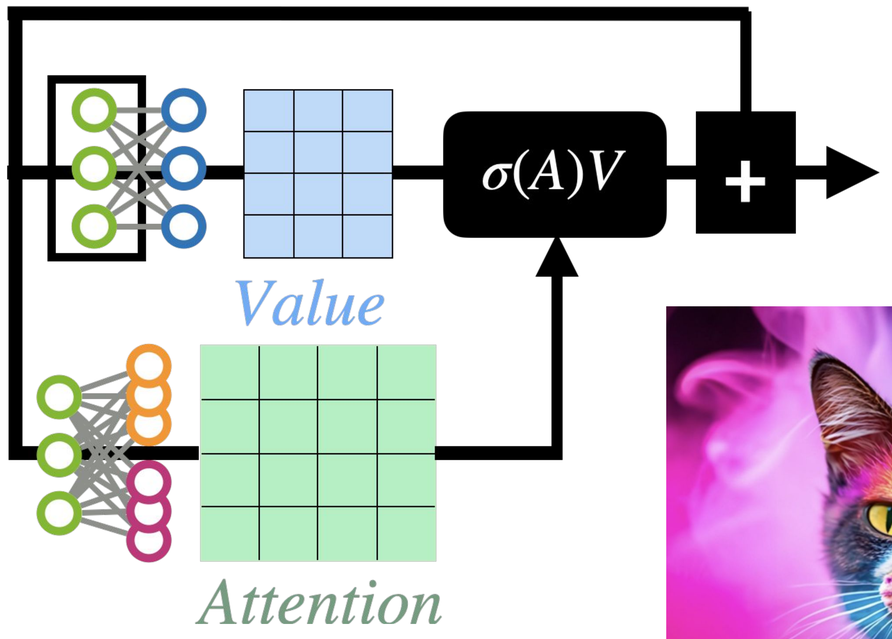
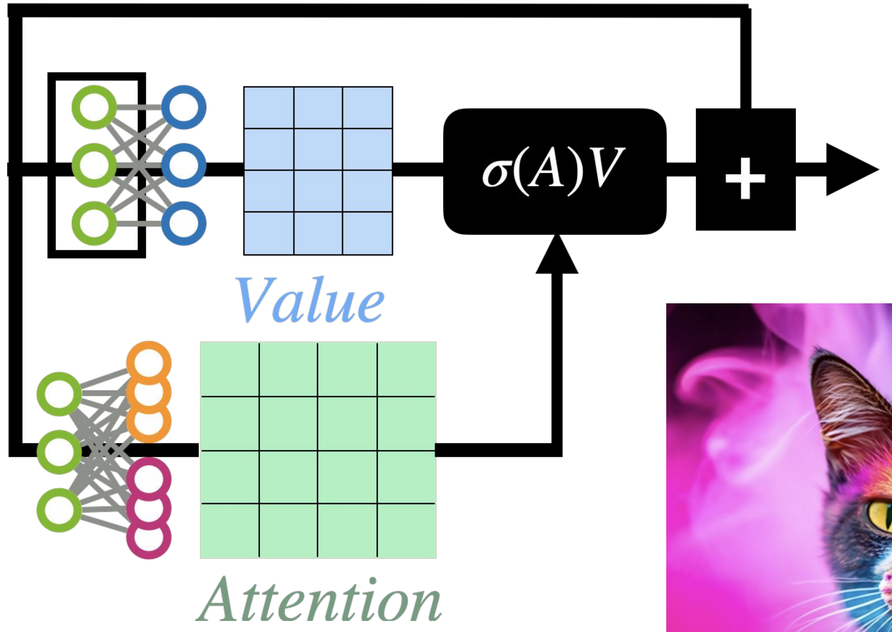


Image Generation?

Can Transformers Generate Images?



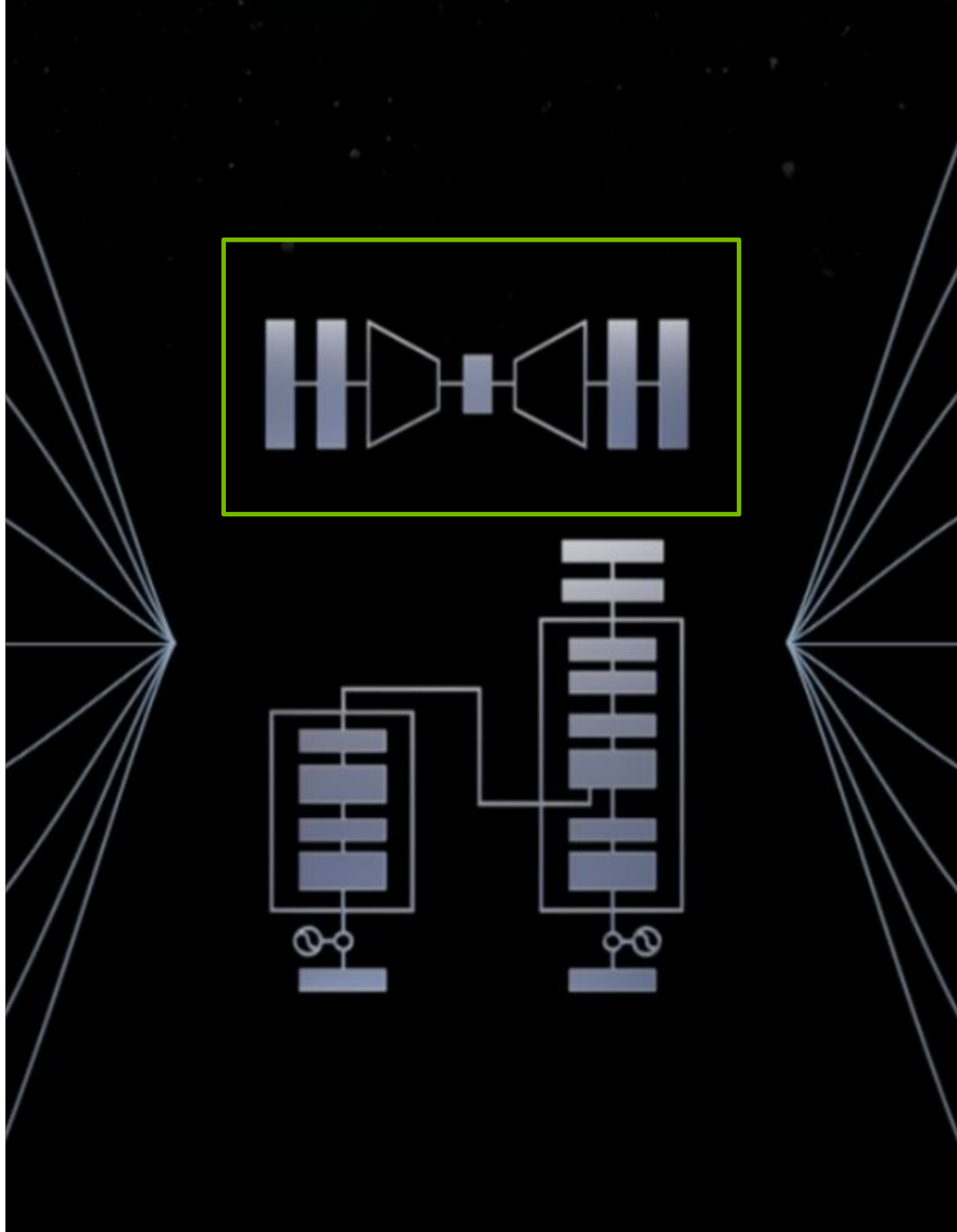
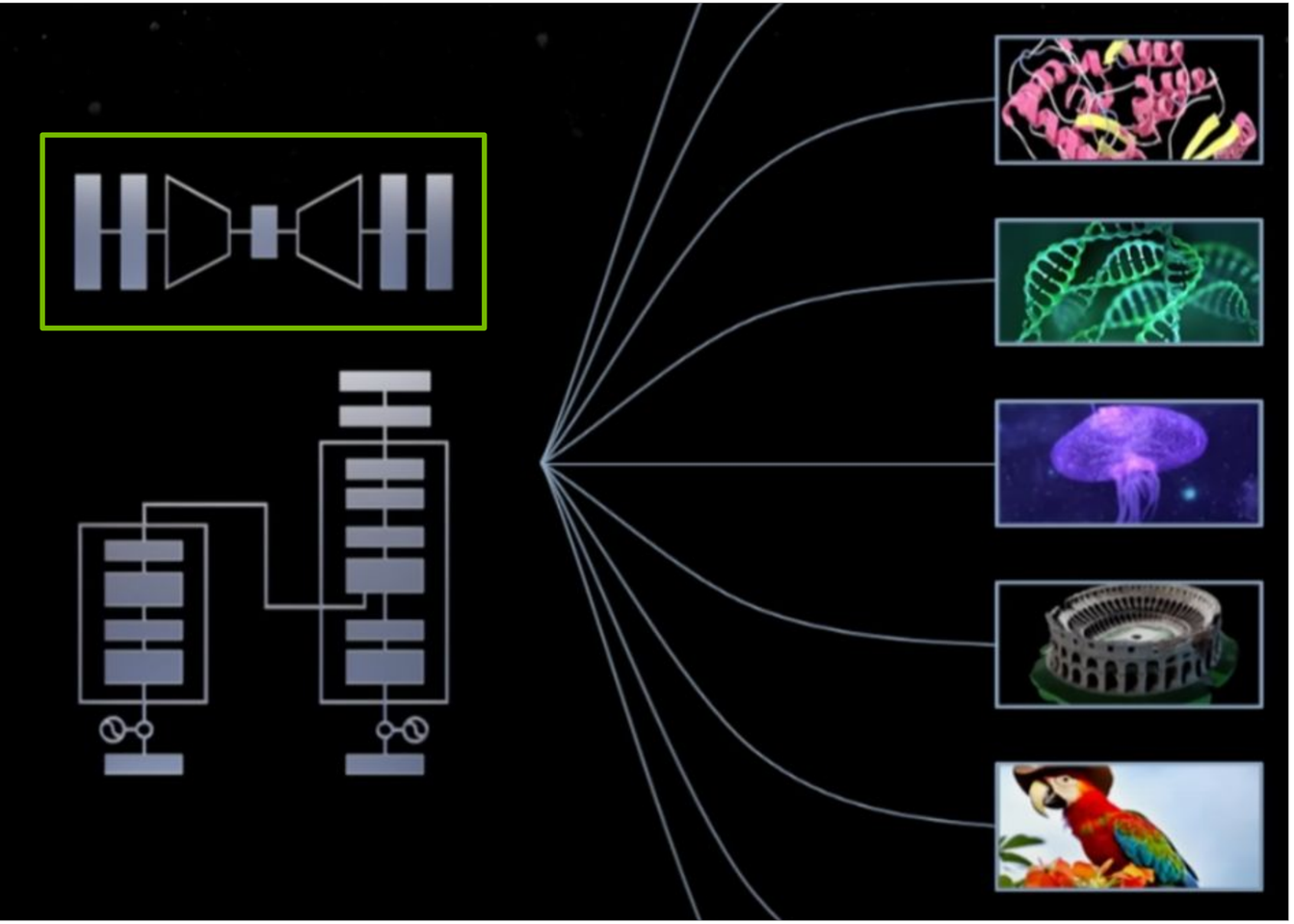
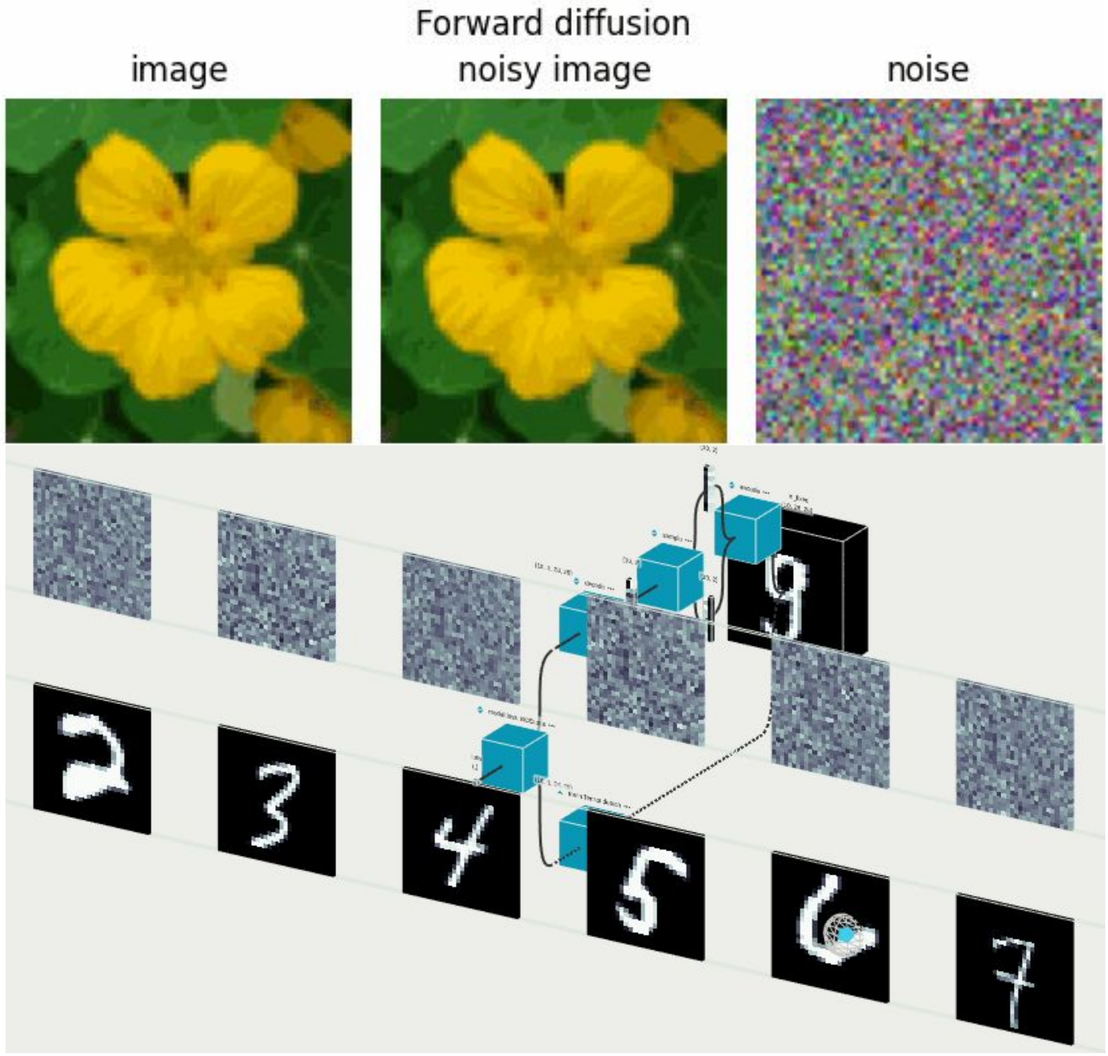


Image Generation?

Can Transformers Generate Images?

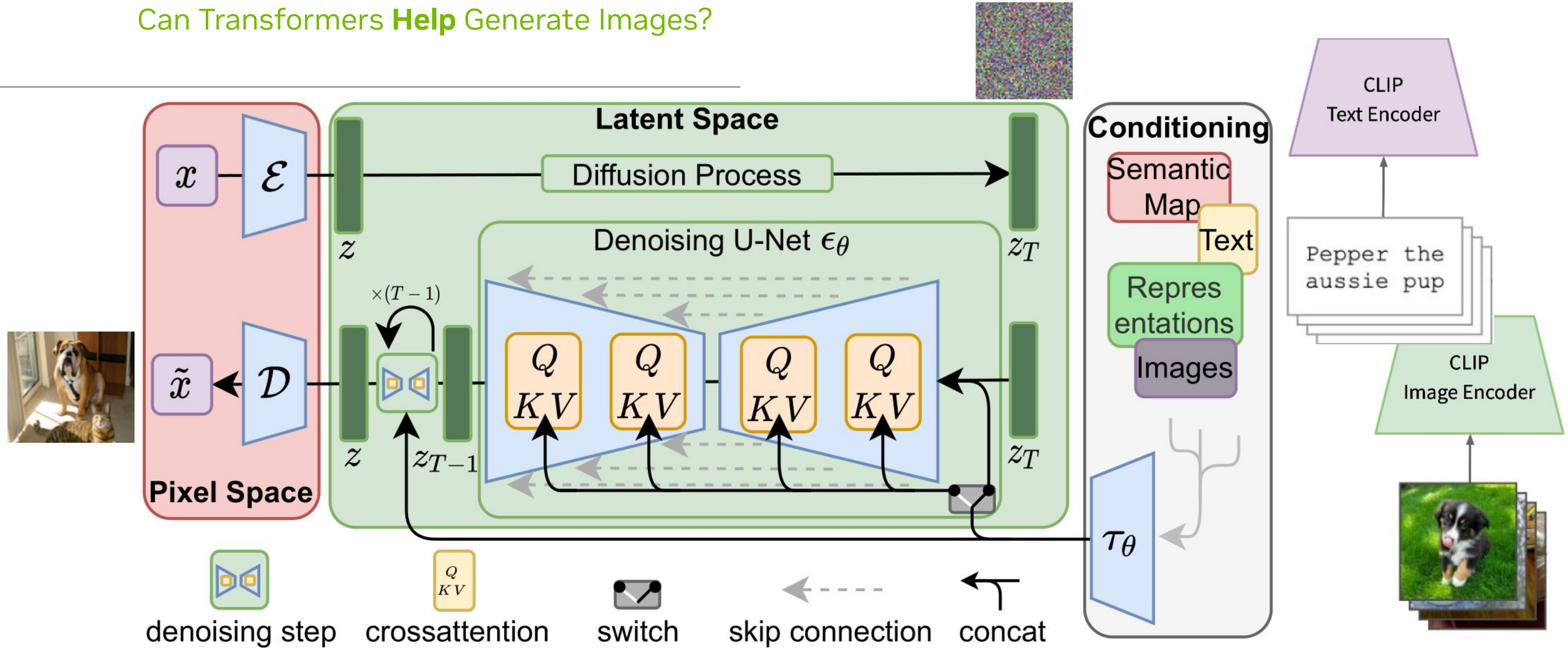




https://ichko.github.io/visualizing-vae-with-efemarai
https://keras.io/examples/generative/ddim/

Image Generation?

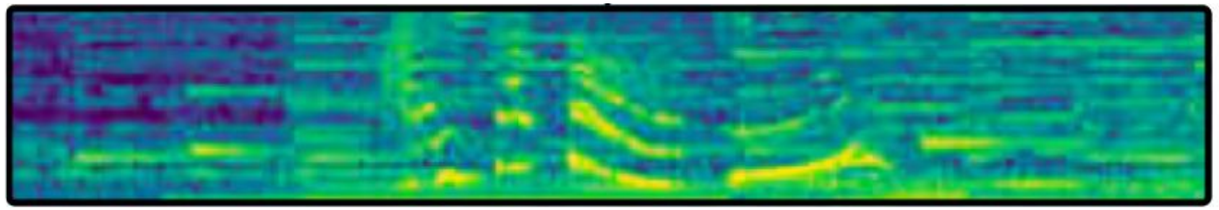
Can Transformers **Help** Generate Images?



https://arxiv.org/abs/2112.10752

Core Transformer Benefits

When Does It Shine



Delivers Semantically-Dense Embeddings

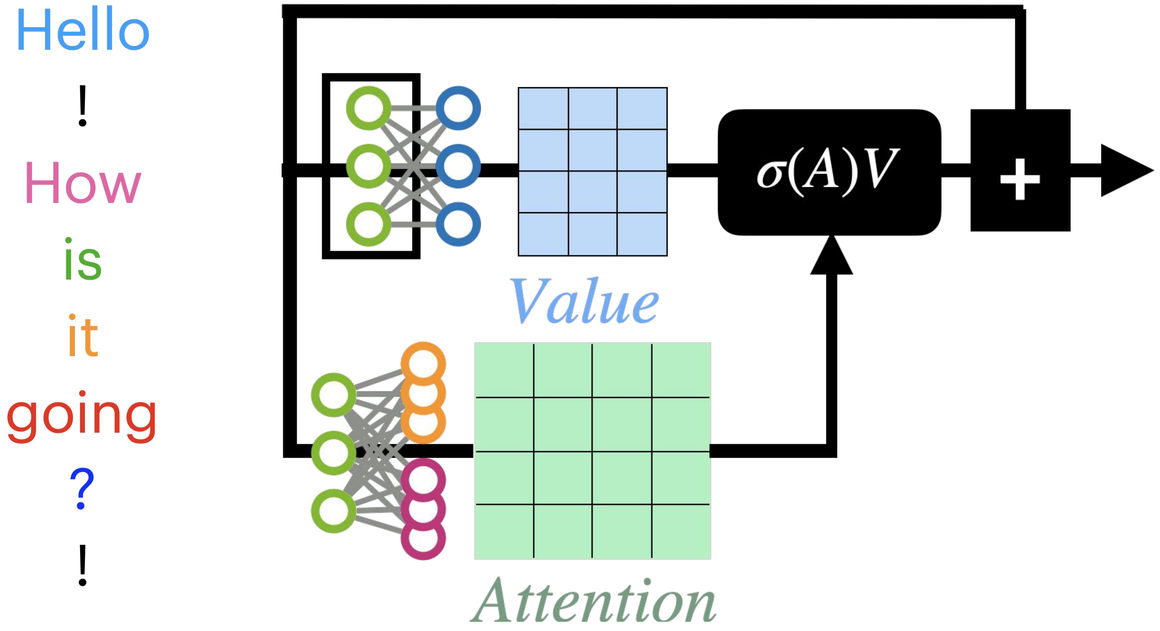
- Easy to reuse. Easy to transfer
- Sequence logic and token logic

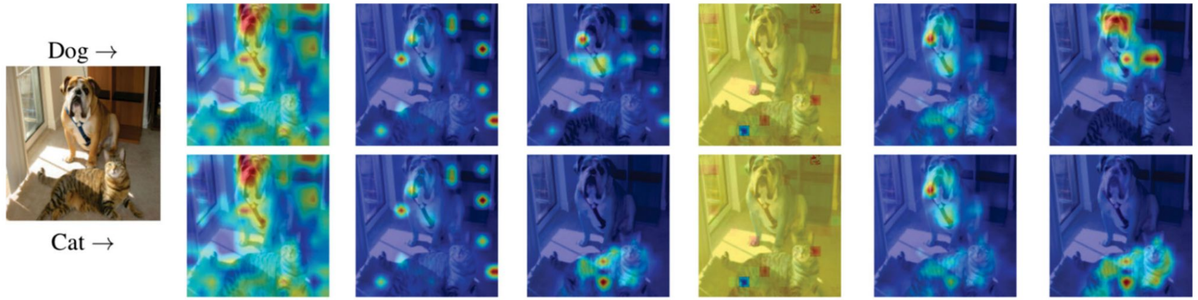
Enforce Attention-Oriented Logic

- Lightweight supervisory interface
- Doubles as interpretable “I care about X”

Synergize Semantically-Dense Embeddings

- Link two modalities for powerful results
- Good for zero-shot and training enforcement





Part 2: Decoding Models

- Encoder-Decoders for Seq2Seq
- Cross-Attention for Multimodal Models
- Large Decoder Models for Text Gen

HF Pipeline Options

Automatic End-to-End Pipelines

NLP Classification

feature-extraction
text-classification
token-classification
fill-mask

NLP Response

question-answering
summarization
translation
text2text-generation
zero-shot-classification

table-qa
document-qa

NLP Generative

text-generation
conversational

Audio

audio-classification
automatic-speech-recognition
text-to-audio
zero-shot-audio-classification

Vision

visual-question-answering
image-classification
image-segmentation
image-to-text
object-detection
depth-estimation
video-classification
mask-generation
zero-shot-image-classification
zero-shot-object-detection

Conversational Tasks

Forecasting Sentences

Text Generation

Generating text is the task of producing new text.
These models can, for example, fill in incomplete text or paraphrase.

Input
Once upon a time,

Text Gen Model

Output
Once upon a time, we knew that our ancestors were on the verge of extinction. The great explorers and poets of the Old World, from Alexander the Great to Chaucer, are dead and gone. A good many of our ancient explorers and poets have

Conversational

Conversational response modelling is the task of generating conversational text that is relevant, coherent and knowledgable given a prompt.
These models have applications in chatbots, and as a part of voice assistants

Input
Hey my name is Julien! How are you?

Conversation Model

Answer
Hi Julien! My name is Julia! I am well.

Encoder-Decoder Benefits

RECALL: When T5-Style Seq2Seq Shine

Key Features: Powerful and Efficient

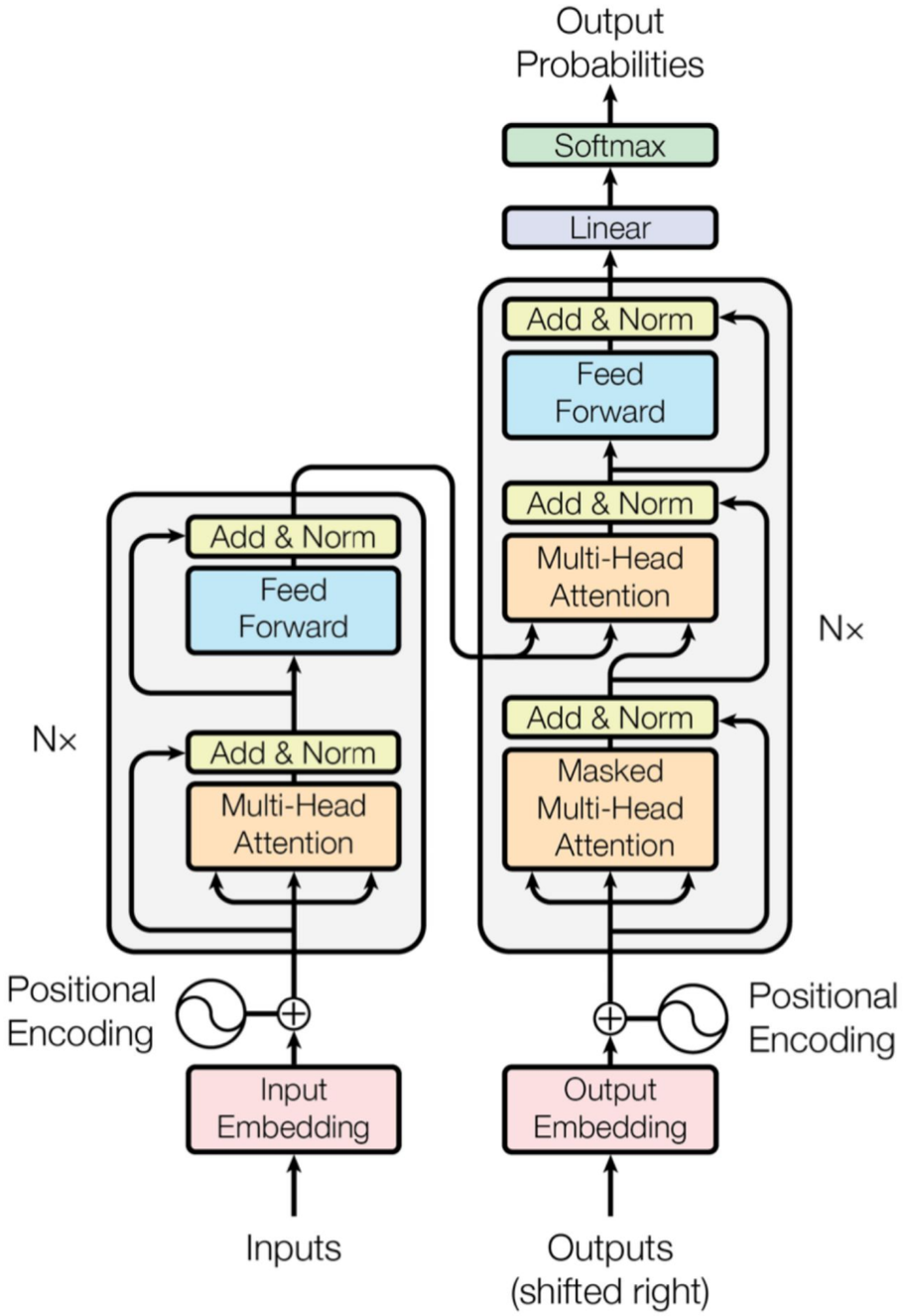
- Grounds harder-to-train decoder with context
- Leverages decoder's forecasting formulation

BEST WHEN

- Context and response are strongly separated

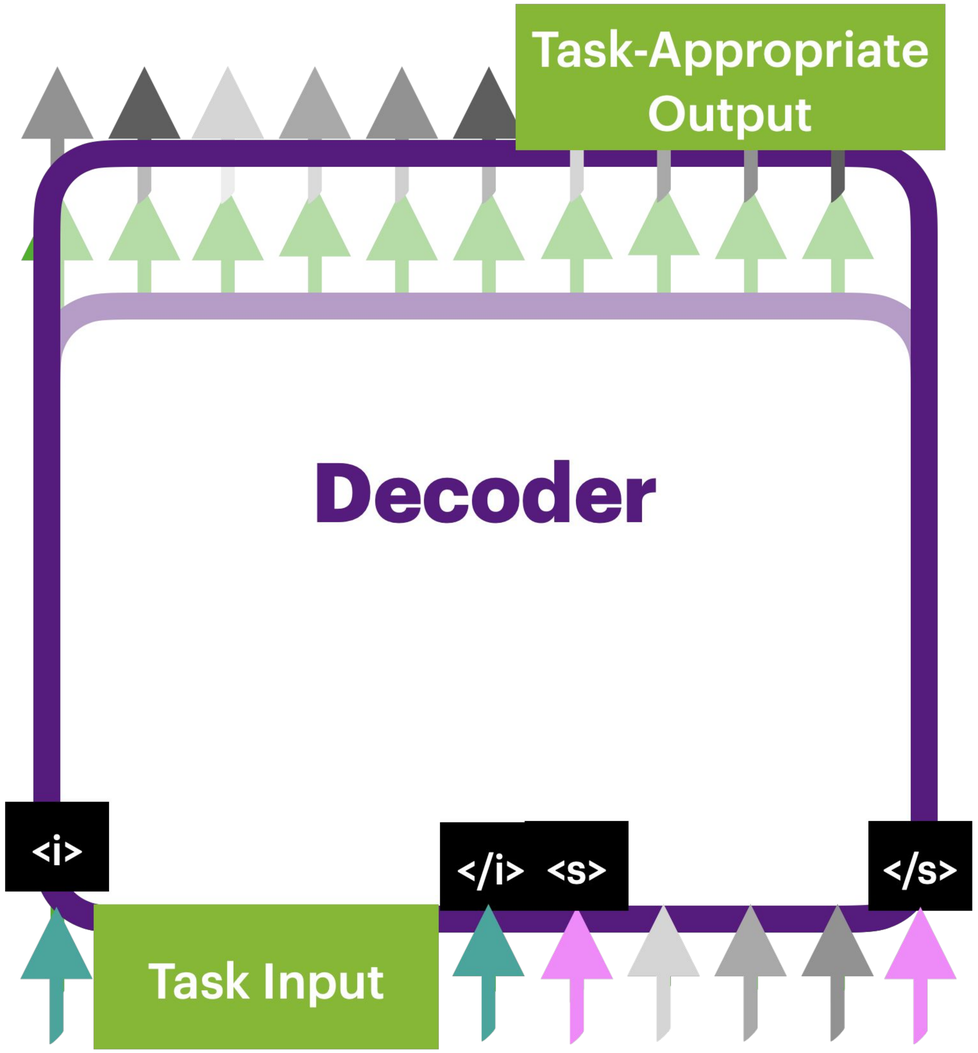
BUT NOT IDEAL WHEN

- There is little-to-no context/response division



Decoder-Only Options

Continue The Thought



HARD PROS

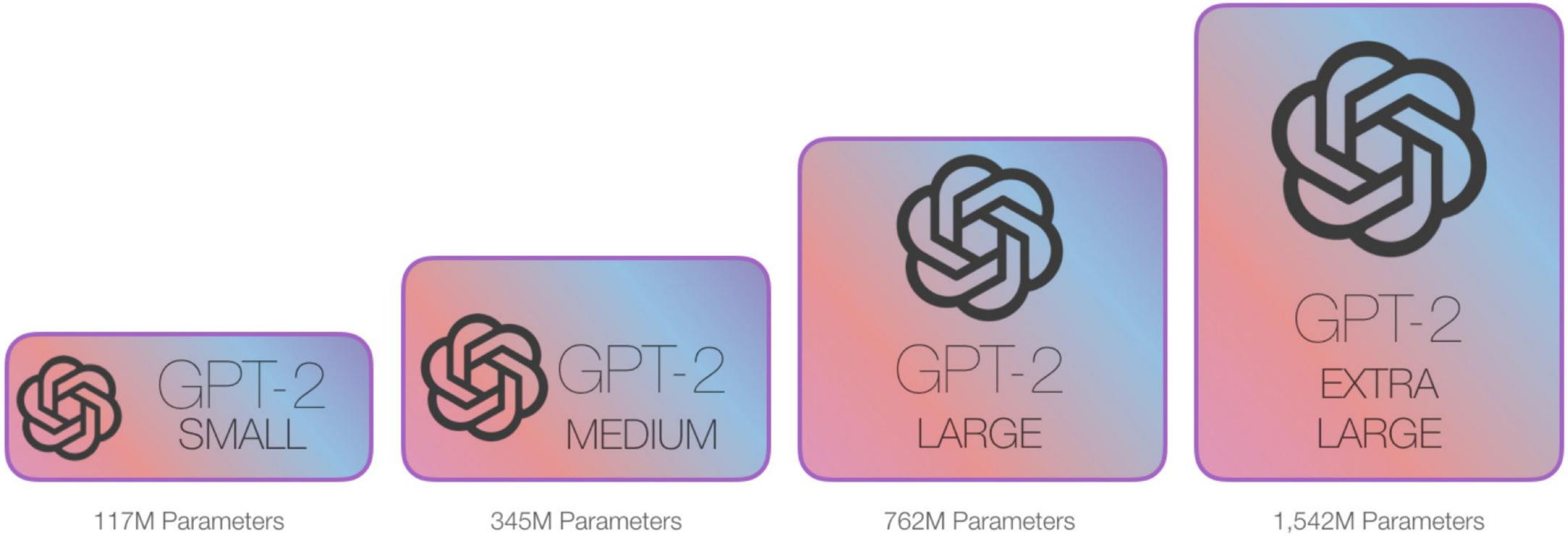
- Everything goes in one path
- Decoder gets much more training

HARD CONS

- Needs way more training
- Needs to handle way more inputs

GPT Series

Decoder-only Flagship



http://jalammar.github.io/illustrated-gpt2/

GPT Series

Decoder-only Flagship

Sufficient For Simplistic Task Backbones

GREATLY Insufficient For Real Chat

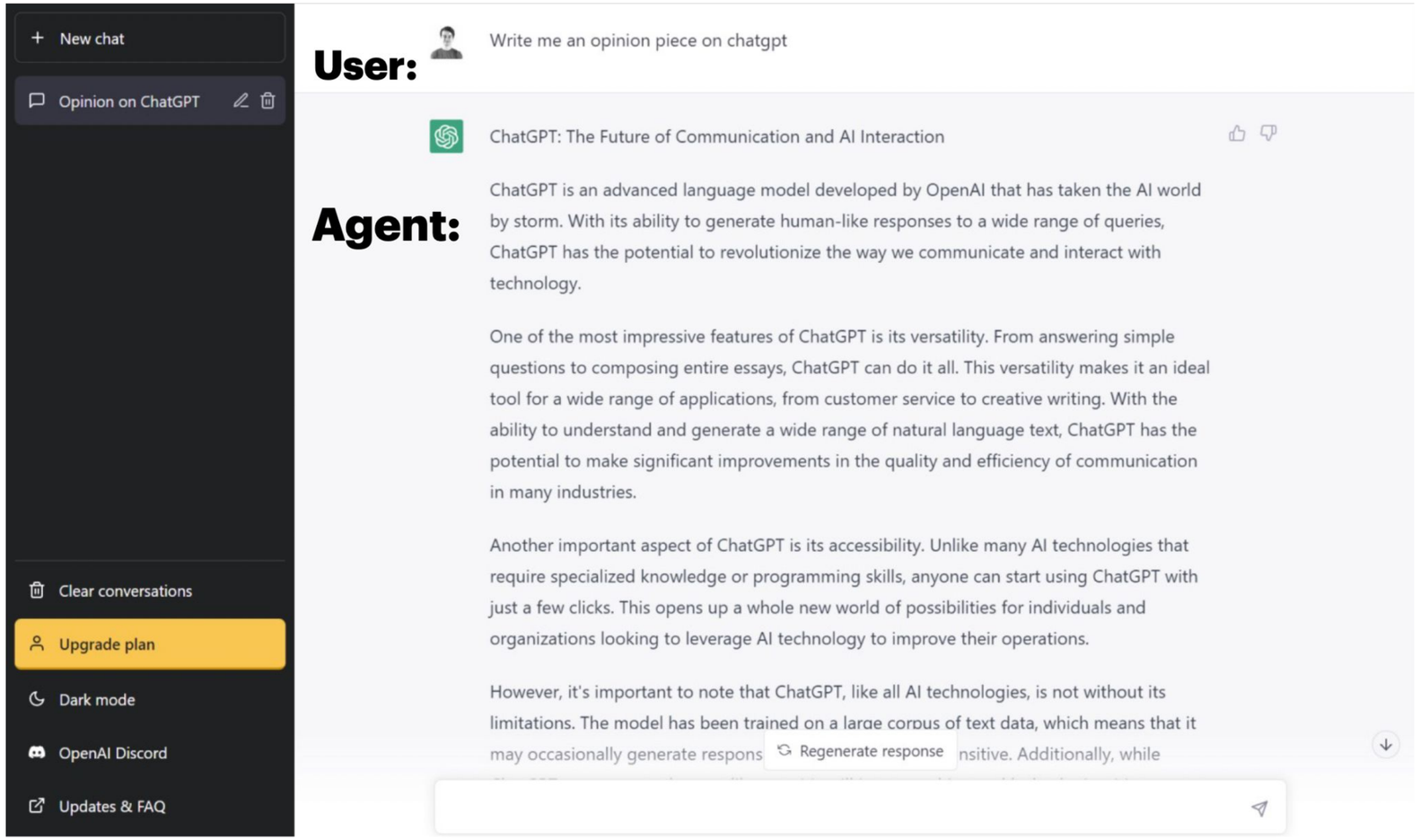
http://jalammar.github.io/illustrated-gpt2/

ChatGPT

GPT-Based Conversational Model

System Message: Please be helpful and nice

Example Conversation: ...



https://www.trustedreviews.com/opinion/ctrlaltdelete-a-chatgpt-powered-bing-could-finally-dethrone-google-4301438

GPT In Popular Culture

What We Want LLMs To Do

Features Should Include:

- Support Arbitrary Conversation Options
- Follow Along And Reason Well
- Be Nice, Inoffensive, and Helpful
- Don't Fall Apart
- Be As Correct As Possible
- Be As Powerful As Humans
- Be More Powerful Than Humans
- Solve Every Problem Ever

ChatGPT

Decoder-Based LLM

Chat-Specific Fine-Tuning

RLHF/RAG Training

Instruction Fine-Tuning

Many Bells and Whistles

GPT In Popular Culture

What We Want LLMs To Do

Features Should Include:

- Support Arbitrary Conversation Options
- Follow Along And Reason Well
- Be Nice, Inoffensive, and Helpful
- Don’t Fall Apart
- Be As Correct As Possible
- Be As Powerful As Humans
- Be More Powerful Than Humans
- Solve Every Problem Ever

ChatGPT

Decoder-Based LLM

Chat-Specific Fine-Tuning

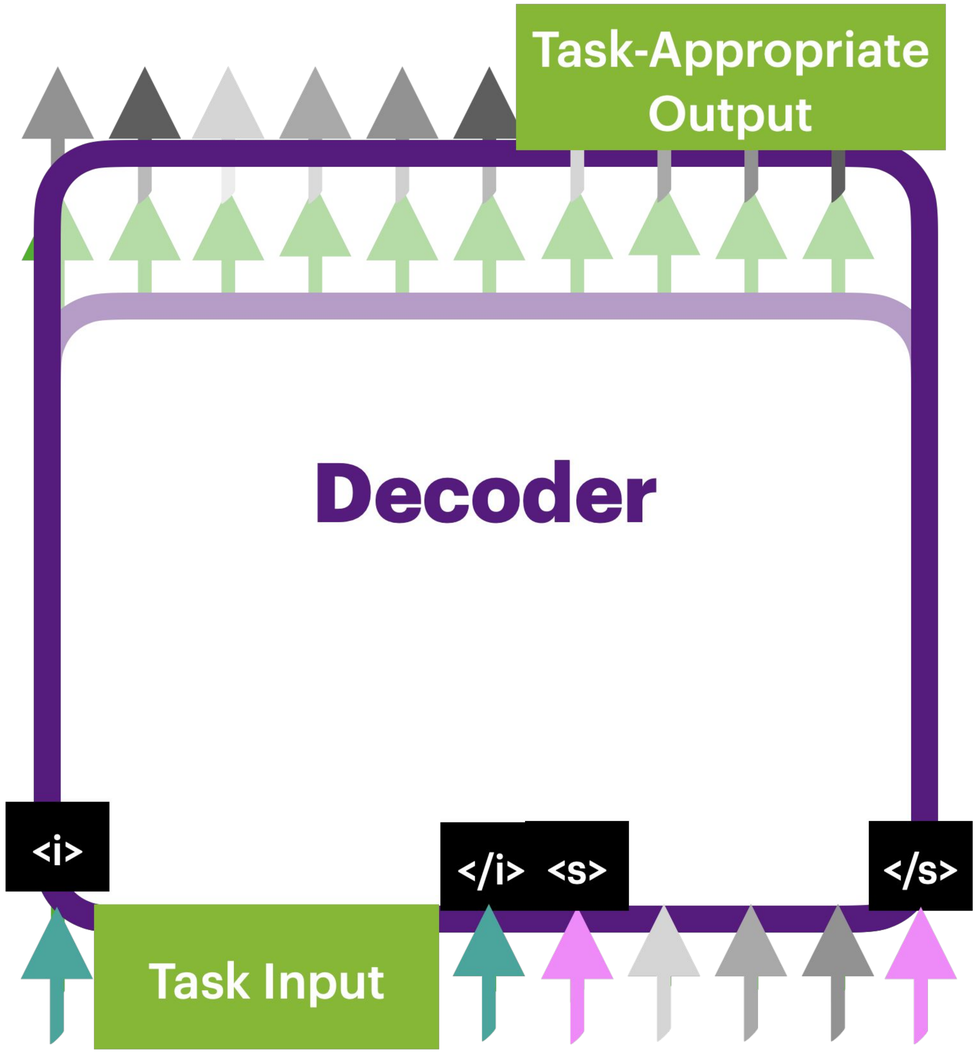
RLHF/RAG Training

Instruction Fine-Tuning

Many Bells and Whistles

Decoder Default Behavior

What “Should” A Good Conversation Model Do?



Have Good Grasp of General Language
Maintain Good Default Behavior
Be Able To Follow Instructions

Pre-Training

Have Specific Behavior Patterns
Have Highly-Specialized Vocabulary

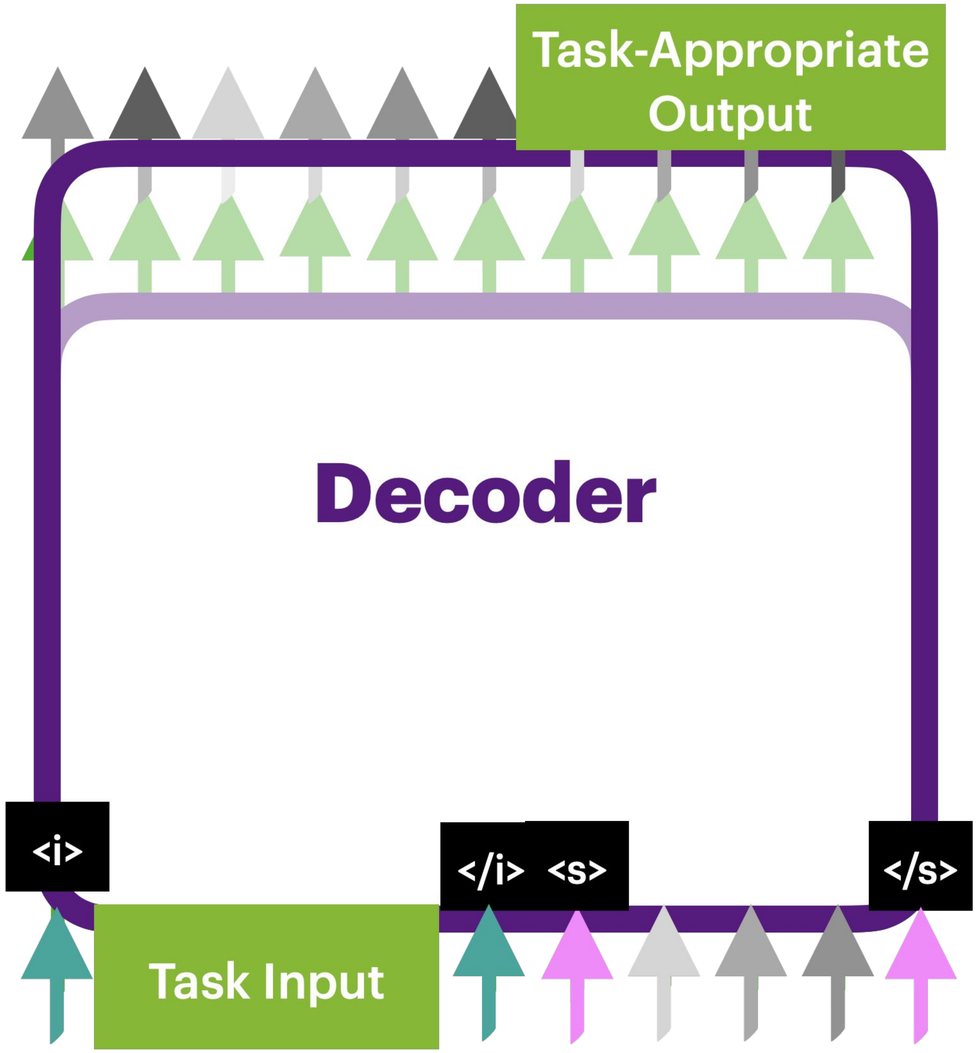
Prompt-Eng
Fine-Tuning

Tell Things Truthfully
Think Things Through
Be Able To Take Actions

Extra Controls
External Data
Tooling Access

Decoder Default Behavior

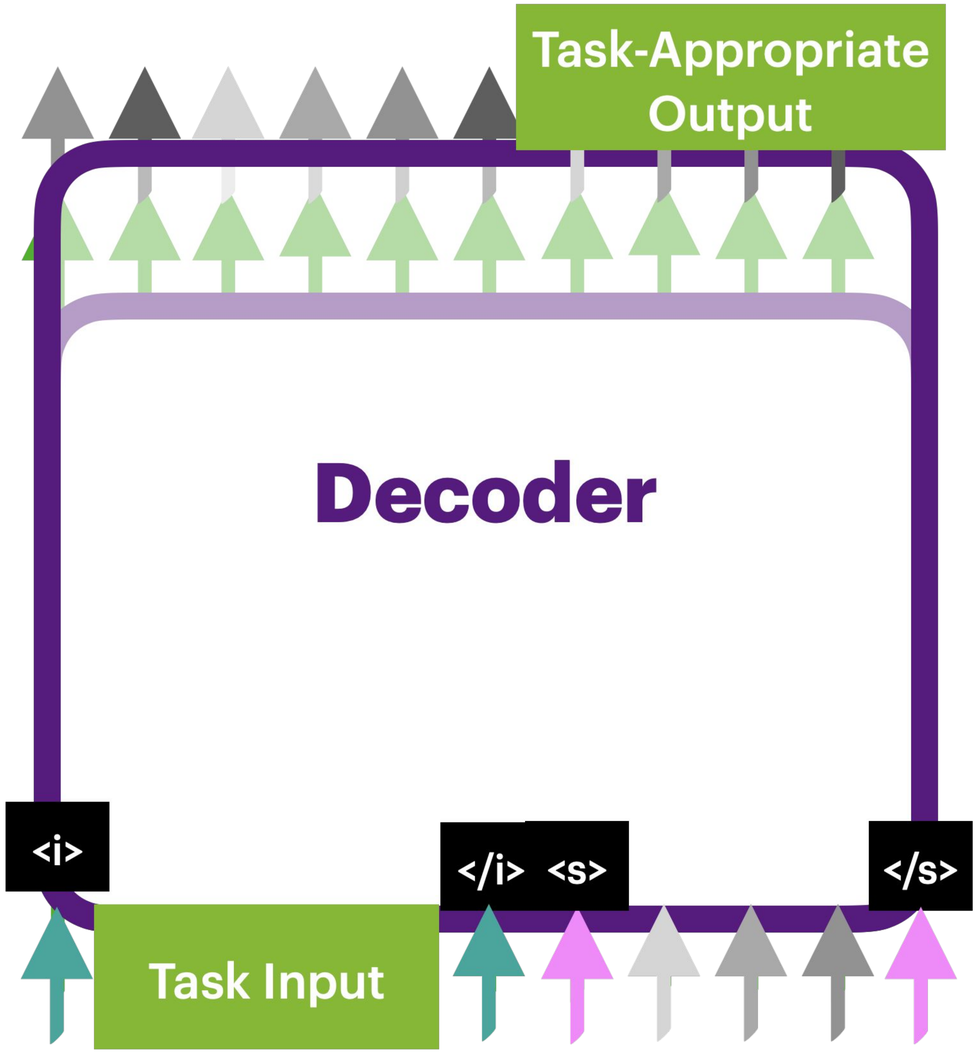
What “Should” A Good Conversation Model Do?



Have Good Grasp of General Language Maintain Good Default Behavior Be Able To Follow Instructions	**Pre-Training**
Have Specific Behavior Patterns Have Highly-Specialized Vocabulary	Prompt-Eng Fine-Tuning
Tell Things Truthfully Think Things Through Be Able To Take Actions	Extra Controls External Data Tooling Access

Decoder Default Behavior

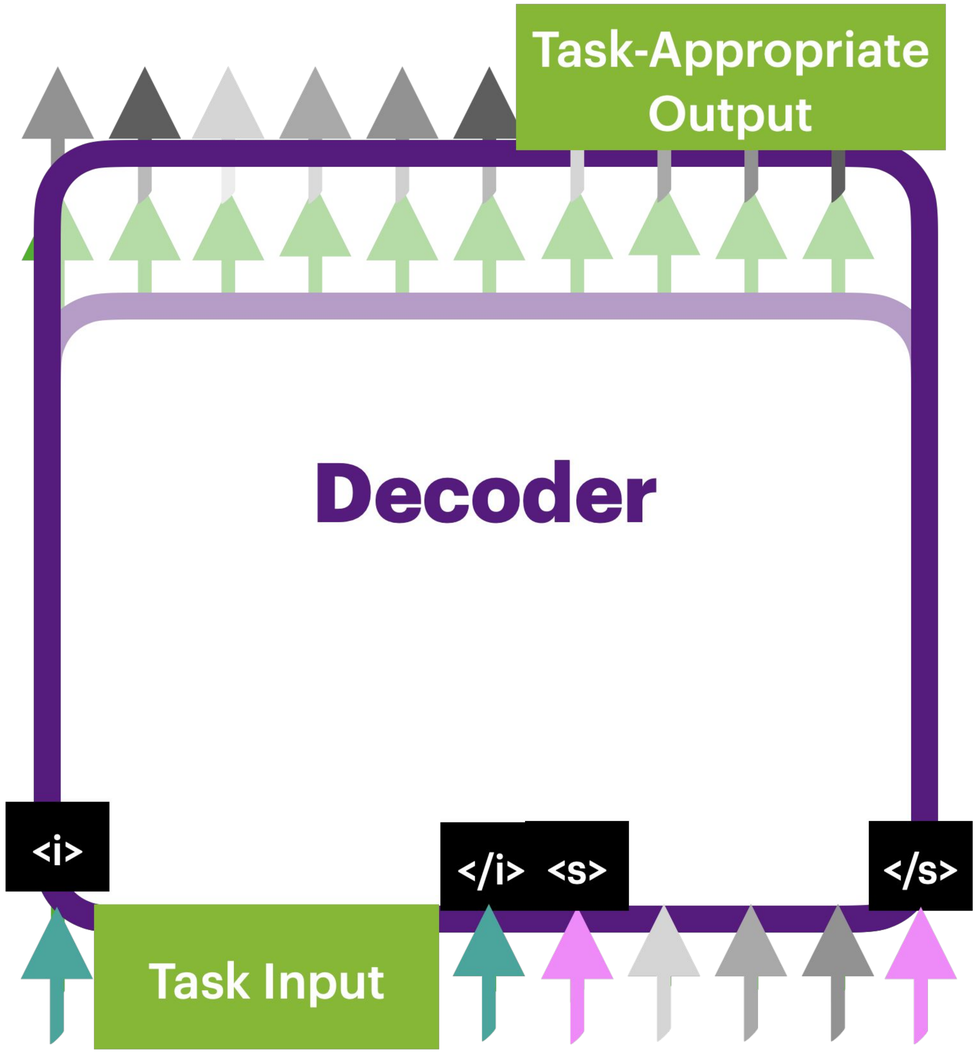
What “Should” A Good Conversation Model Do?



Have Good Grasp of General Language Maintain Good Default Behavior Be Able To Follow Instructions	Pre-Training
Have Specific Behavior Patterns Have Highly-Specialized Vocabulary	Prompt-Eng Fine-Tuning
Tell Things Truthfully Think Things Through Be Able To Take Actions	Extra Controls External Data Tooling Access

Decoder Default Behavior

What “Should” A Good Conversation Model Do?



Have Good Grasp of General Language Maintain Good Default Behavior Be Able To Follow Instructions	Pre-Training
Have Specific Behavior Patterns Have Highly-Specialized Vocabulary	Prompt-Eng Fine-Tuning
Tell Things Truthfully Think Things Through Be Able To Take Actions	Extra Controls External Data Tooling Access

Core Areas To Improve

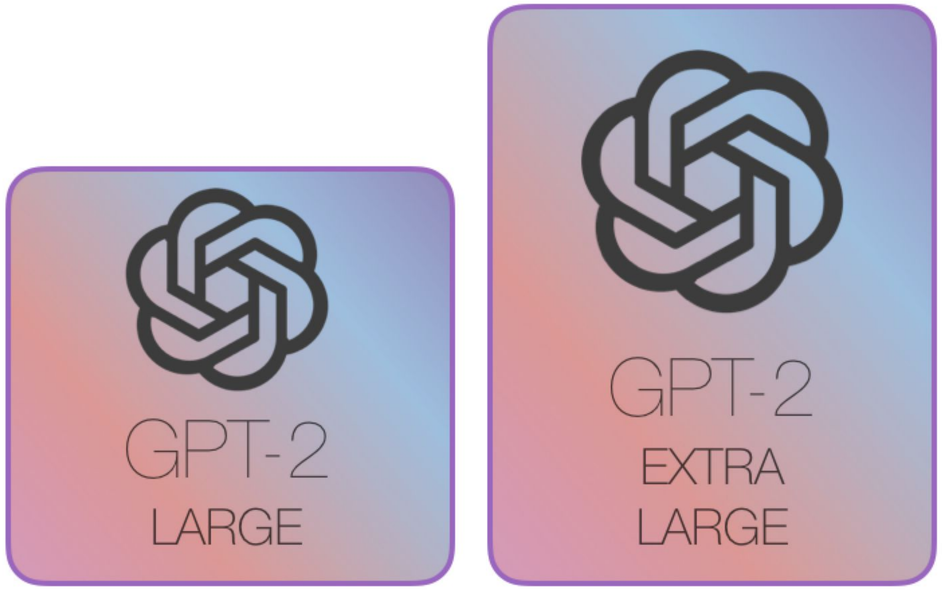
Why Is Conversation So Much Harder?

NEEDS MORE POWER [This Part]

- Needs more training, more data, and larger model.
- Maintain context and consistent autoregression.

STOCHASTIC PARROT BY DEFAULT [Part 3]

- It merely predicts the most probable token.
- Limit to context window and no real memory systems.
- Will hallucinate, follow distribution of training data.

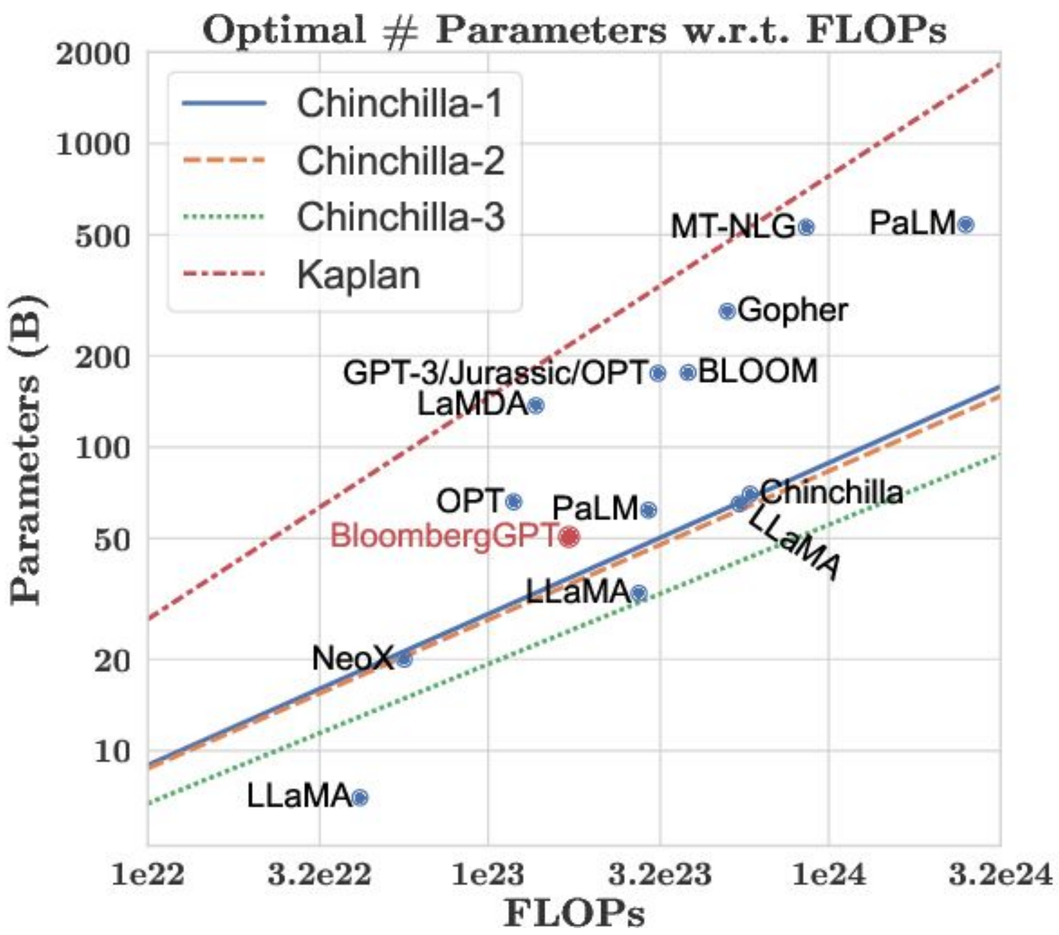


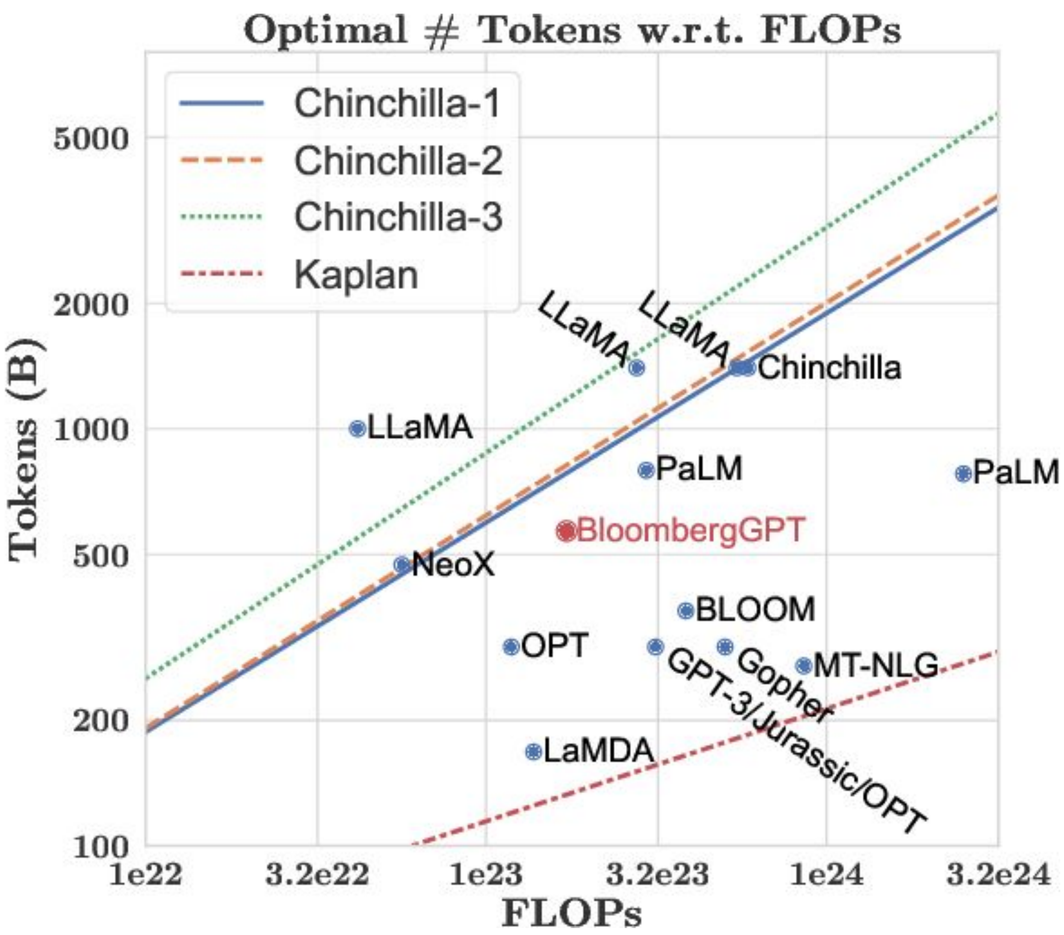
762M Parameters

1,542M Parameters

GPT Scaling Laws

Bigger Is Better… To A Point

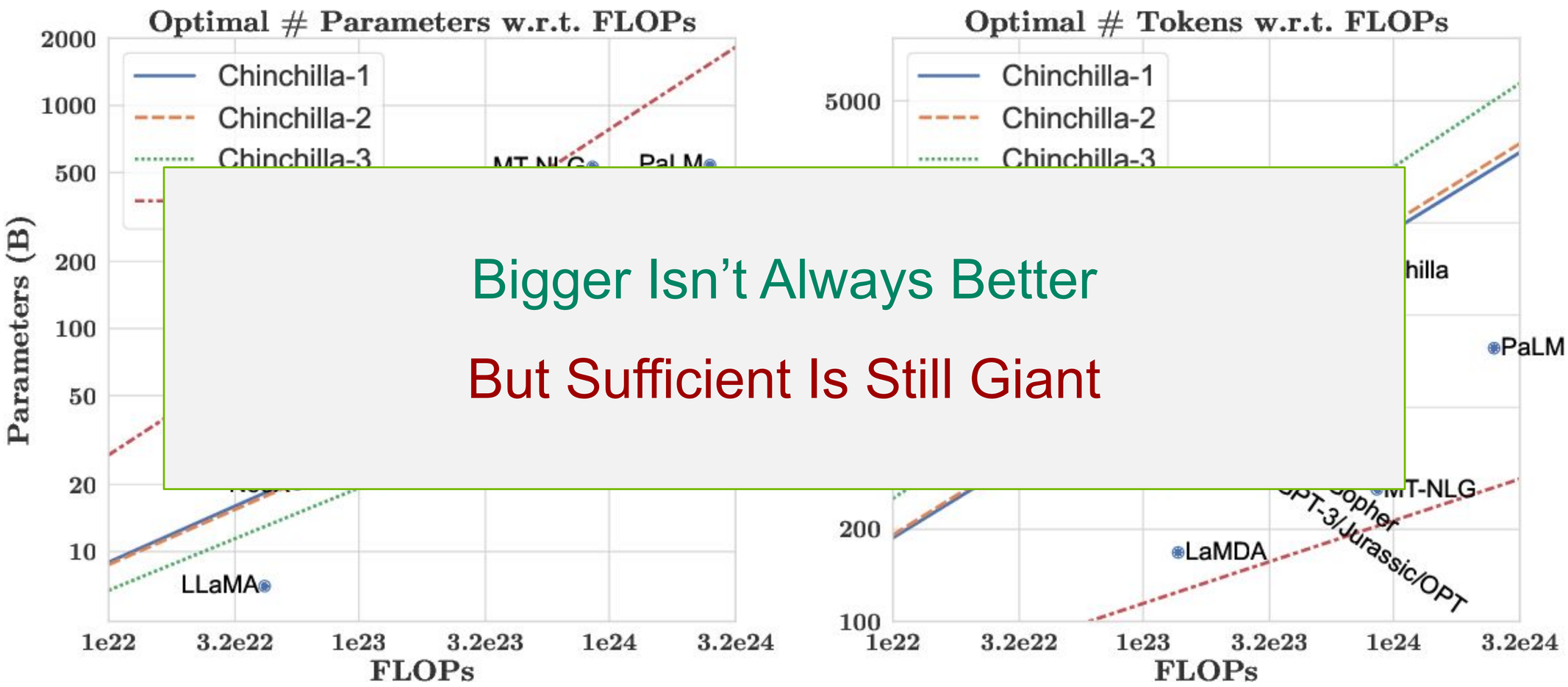




https://www.semanticscholar.org/paper/BloombergGPT%3A-A-Large-Language-Model-for-Finance-Wu-Irsoy/83edcfbb206ddad38a971d605da09390604248ea

GPT Scaling Laws

Bigger Is Better… To A Point



Bigger Isn't Always Better

But Sufficient Is Still Giant

https://www.semanticscholar.org/paper/BloombergGPT%3A-A-Large-Language-Model-for-Finance-Wu-Irsoy/83edcfbb206ddad38a971d605da09390604248ea

Off-The-Shelf Models

Starting Points For Conversation Tasks

GPT4 (OpenAI)

- Great SOTA performance. ChatGPT Backbone.
- Black-boxy, and only cloud-hosted.

GPT-43B (NVIDIA)

- Currently NeMo-Service-walled.
- More useful for deployment.

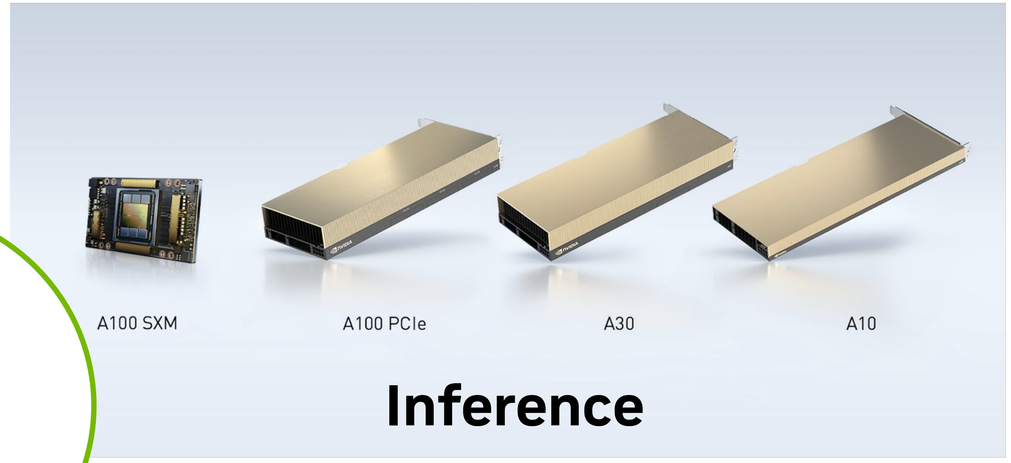
Llama-2 (Meta)

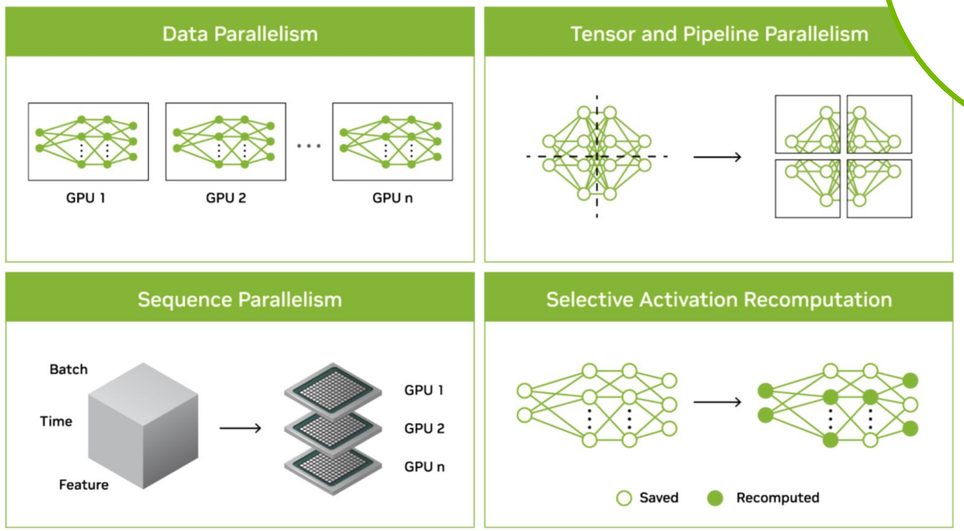
- SOTA Models w/ Llama-2 license.
- Not OSS but commercial-friendly.
- Locally-hosted and HF-compatible.

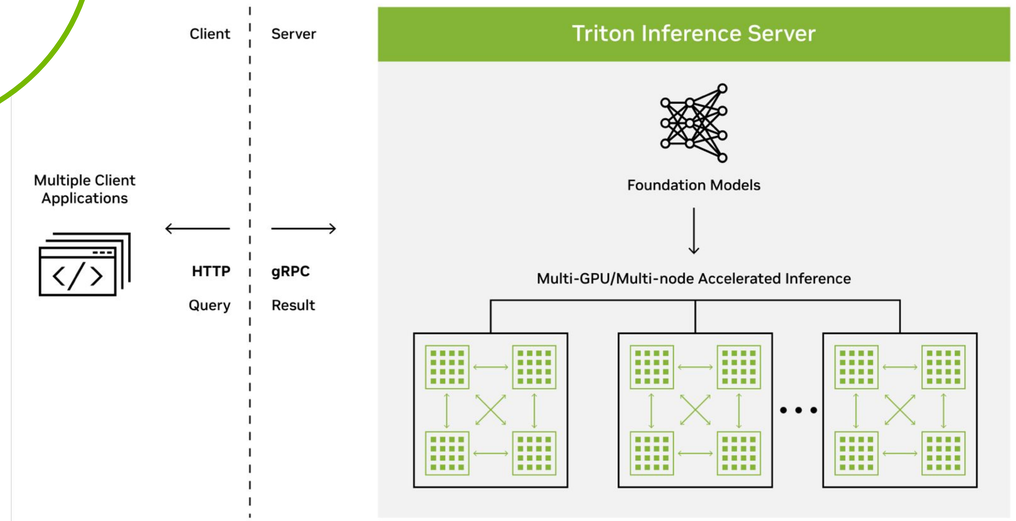
Handling Large LLMs

A Compute / Skillset Budget

Fine Tuning

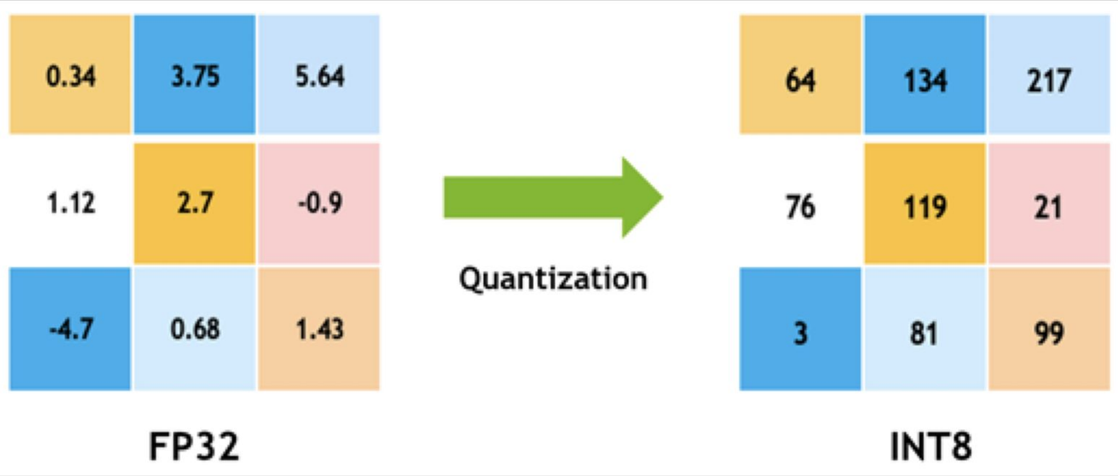




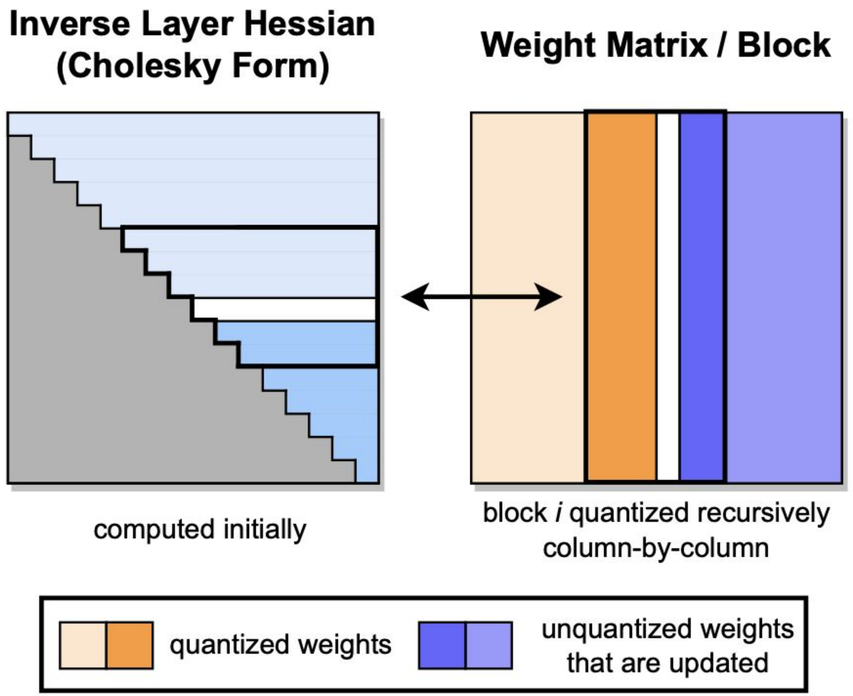


Quantized LLMs

Reducing Model Sizes



General Quantization



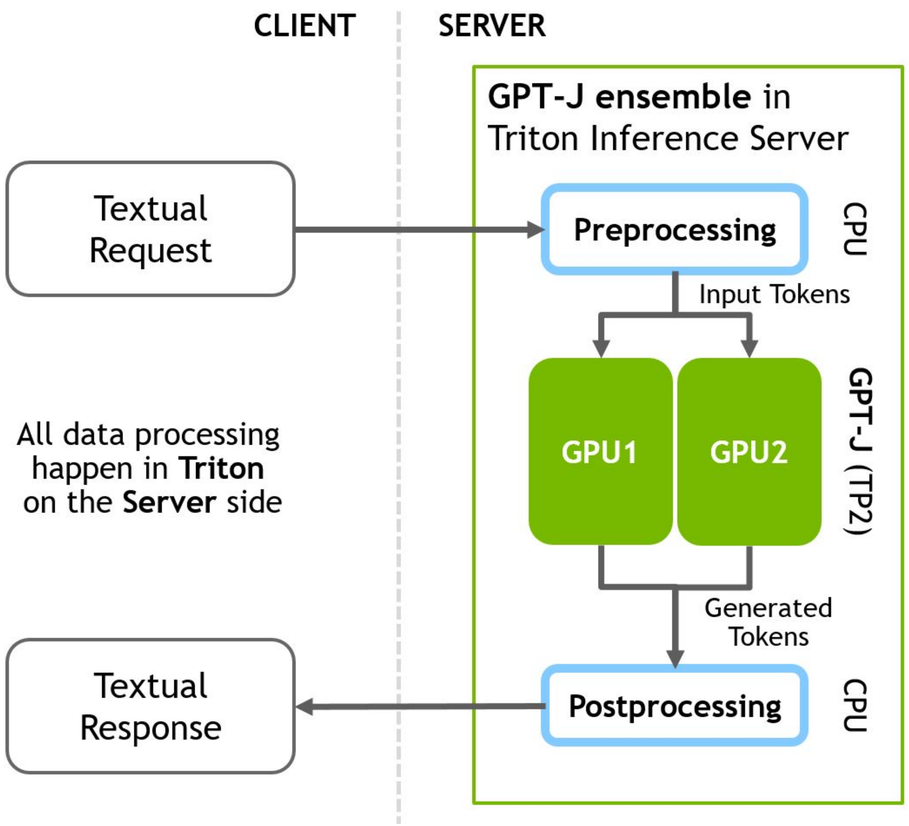
GPTQ Adaptive Quantization

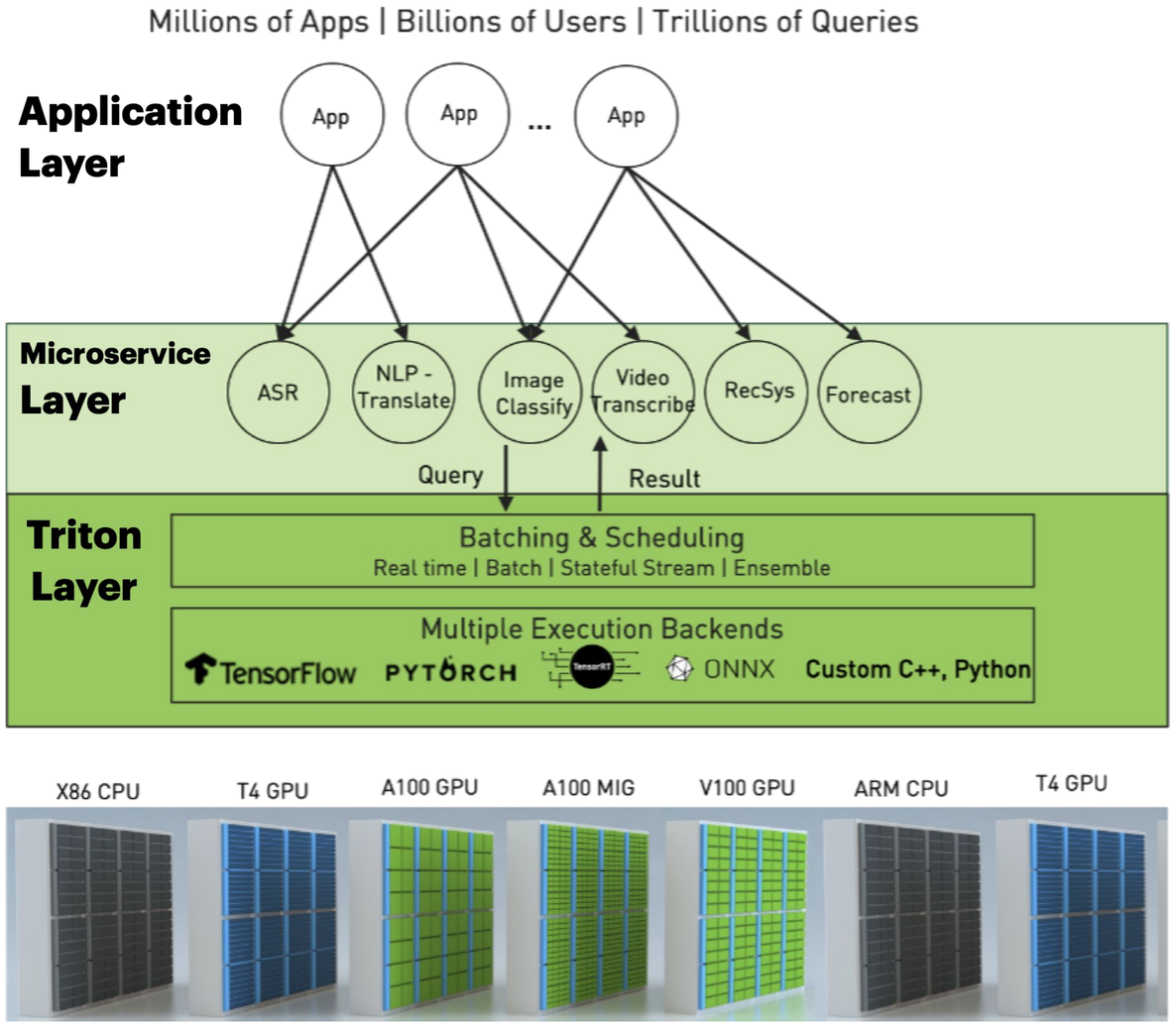
https://developer.nvidia.com/blog/deploying-gpt-j-and-t5-with-fastertransformer-and-triton-inference-server/

https://developer.nvidia.com/blog/simplifying-ai-inference-in-production-with-triton/

Server-Hosted LLMs

Streaming Responses from LLM Pool



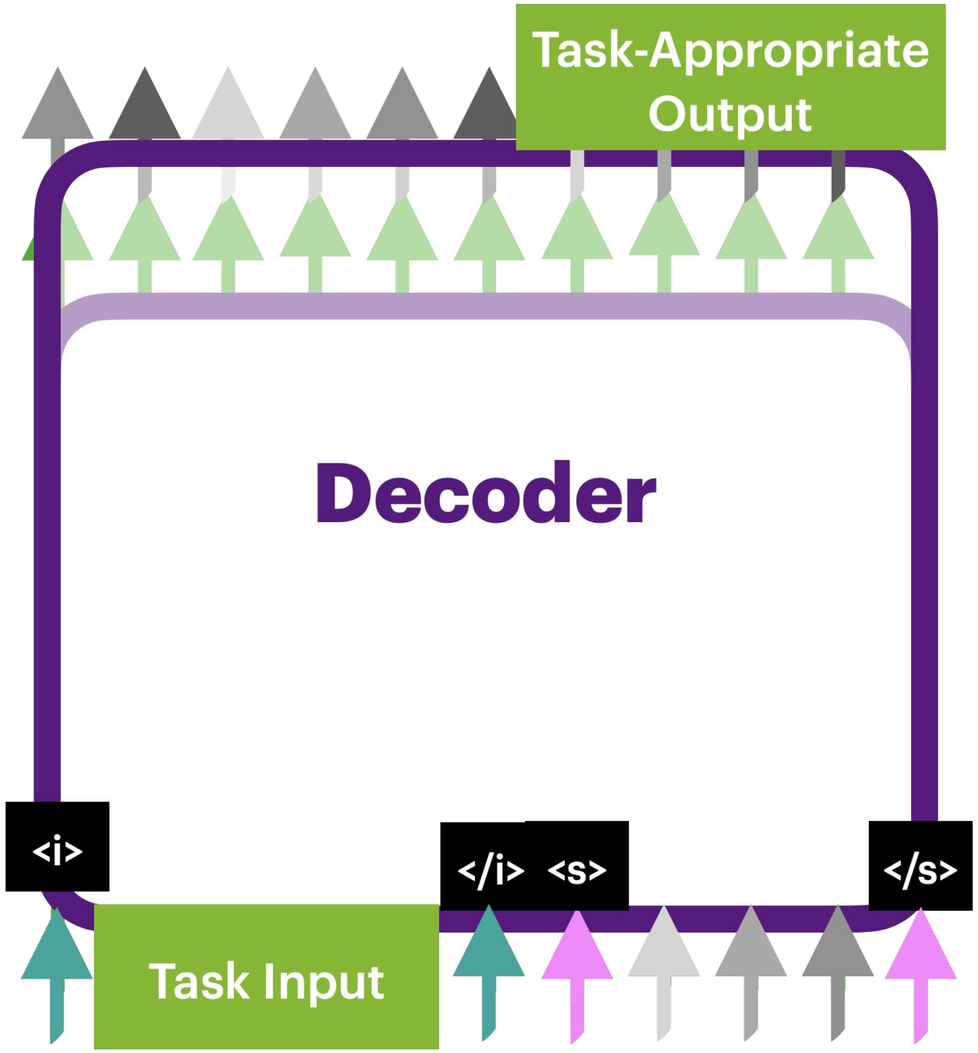


https://developer.nvidia.com/blog/deploying-gpt-j-and-t5-with-fastertransformer-and-triton-inference-server/

https://developer.nvidia.com/blog/simplifying-ai-inference-in-production-with-triton/

Decoder Default Behavior

What “Should” A Good Conversation Model Do?



Have Good Grasp of General Language Maintain Good Default Behavior Be Able To Follow Instructions	Pre-Training
Have Specific Behavior Patterns Have Highly-Specialized Vocabulary	Prompt-Eng Fine-Tuning
Tell Things Truthfully Think Things Through Be Able To Take Actions	Extra Controls External Data Tooling Access

Good Practices

Things To Remember In The Wild

Using Off-The-Shelf	Fine-Tuning Yourself	Using Cloud-Hosted
Use Fine-Tuned SOTA Models	Start From Existing SOTA Models	Know Where Your Data Is Going
Use Smaller/Quantized Models When Possible	Tune Smaller LLMs With Help Of Big LLMs	Obfuscate Your Data
Add Structure and Control	Fine-Tune For Default Behavior!	Re-Verify Performance Regularly

Chatting With Llama-2

A Typical SOTA Instruction-Tuned Model

```
<s>[INST]<<SYS>>
{{system_message}}
<</SYS>>

{{instruction}} [/INST]
{{primer}}
```

```
<s>[INST]<<SYS>>
{{system_message}}
<</SYS>>

{{inst1}} [/INST] {{gen1}}
</s><s>[INST]
{{inst2}} [/INST]
```

```
<s>[INST]<<SYS>>
You are a code generator. Please
provide Python code per the
instruction.
<</SYS>>

Write a Fibonacci method [/INST]
```python
## Implementation of Fibonacci w/
```
```

```
<s>[INST]<<SYS>>
You are a chatbot! Please Help!
<</SYS>>

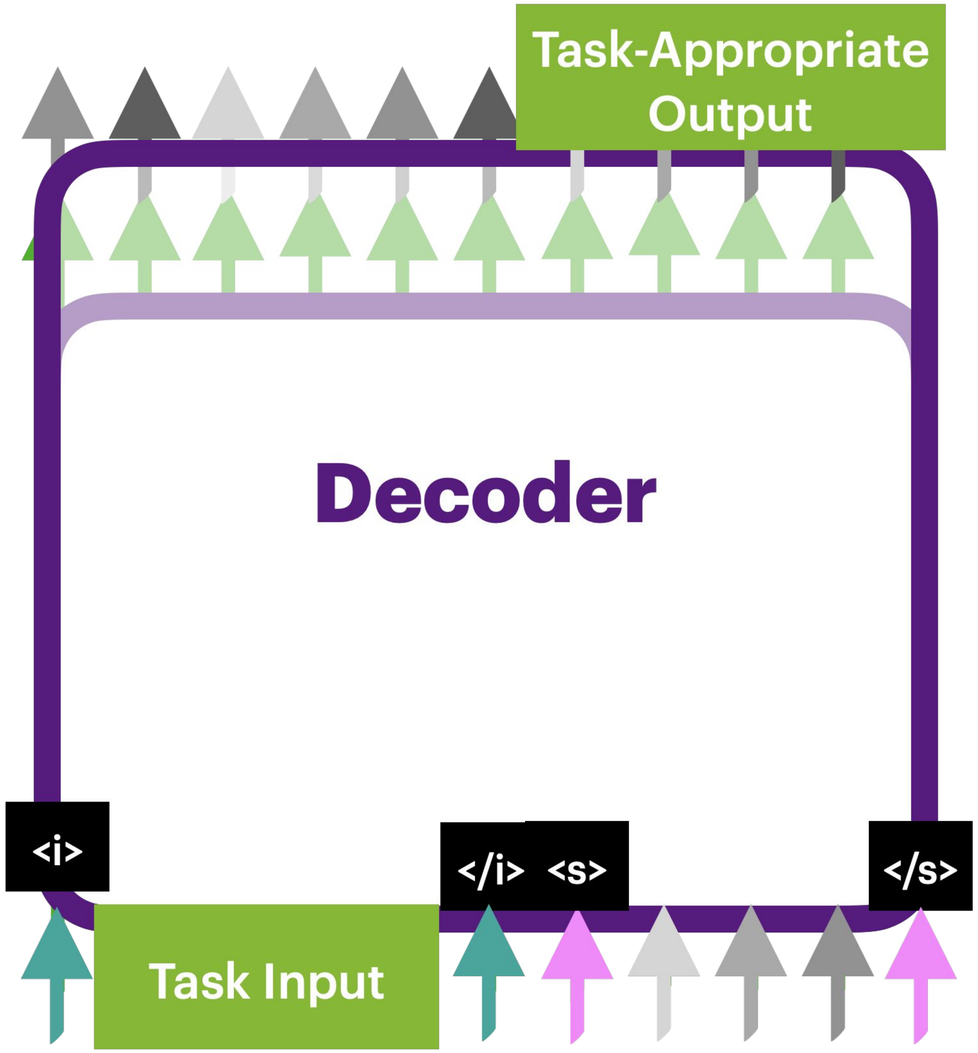
User: Hello Agent
AI: [/INST]
Hello! What can I help you with today?
</s><s>[INST]
User: I'd like to learn more about
Birds!
AI: [/INST]
```

Part 3: Stateful Systems

- Controlling Stateful Agents
- Final Project: Making An Agent

[Recall] Decoder Default Behavior

What "Should" A Good Conversation Model Do?



Have Good Grasp of General Language
Maintain Good Default Behavior
Be Able To Follow Instructions

Pre-Training

Have Specific Behavior Patterns
Have Highly-Specialized Vocabulary

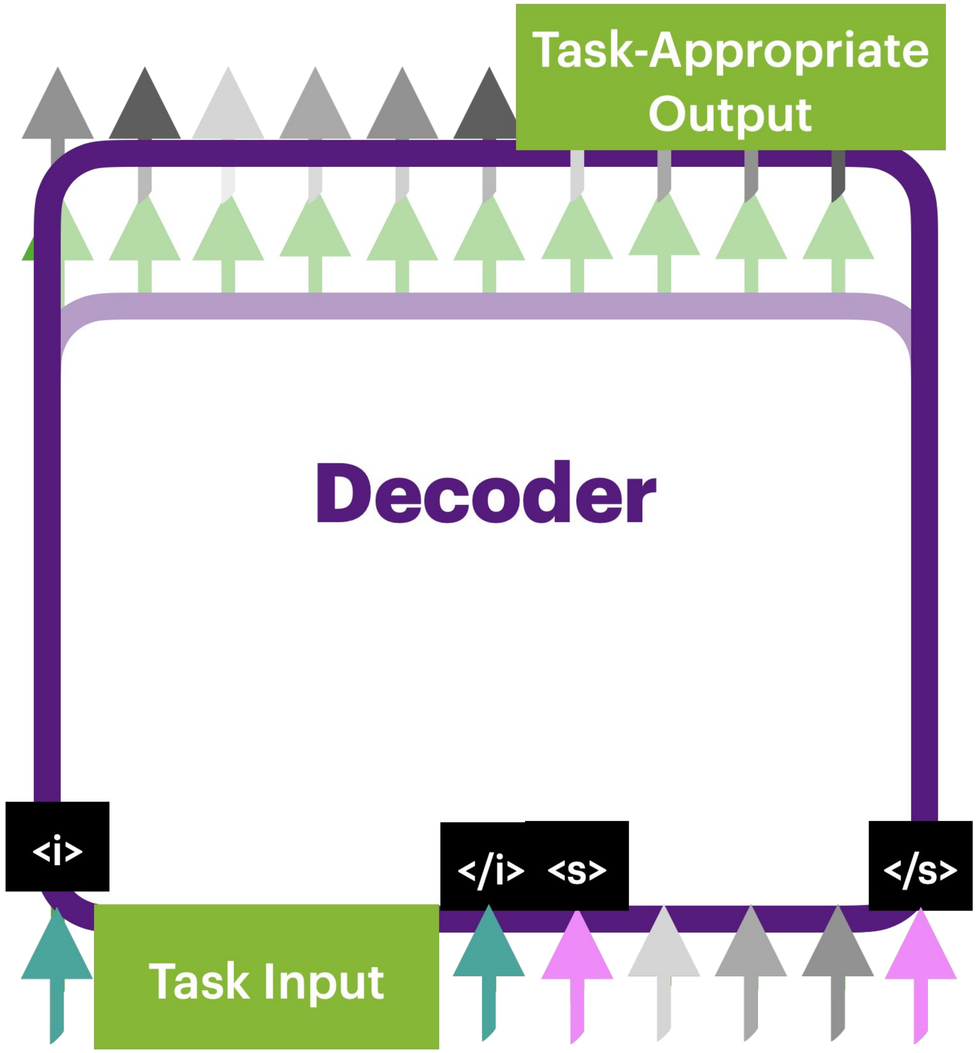
Prompt-Eng
Fine-Tuning

Tell Things Truthfully
Think Things Through
Be Able To Take Actions

Extra Controls
External Data
Tooling Access

[Recall] Decoder Default Behavior

What "Should" A Good Conversation Model Do?



Llama-2-70B is a great base!	**Pre-Training**
Llama-2-70B-HF-Chat is a great base!	**Prompt-Eng** **Fine-Tuning**

Tell Things Truthfully
Think Things Through
Be Able To Take Actions

Extra Controls
External Data
Tooling Access

Core Areas To Improve

Why Is Conversation So Much Harder?

NEEDS MORE POWER [Part 2]

- Needs more training, more data, and larger model.
- Maintain context and consistent autoregression.

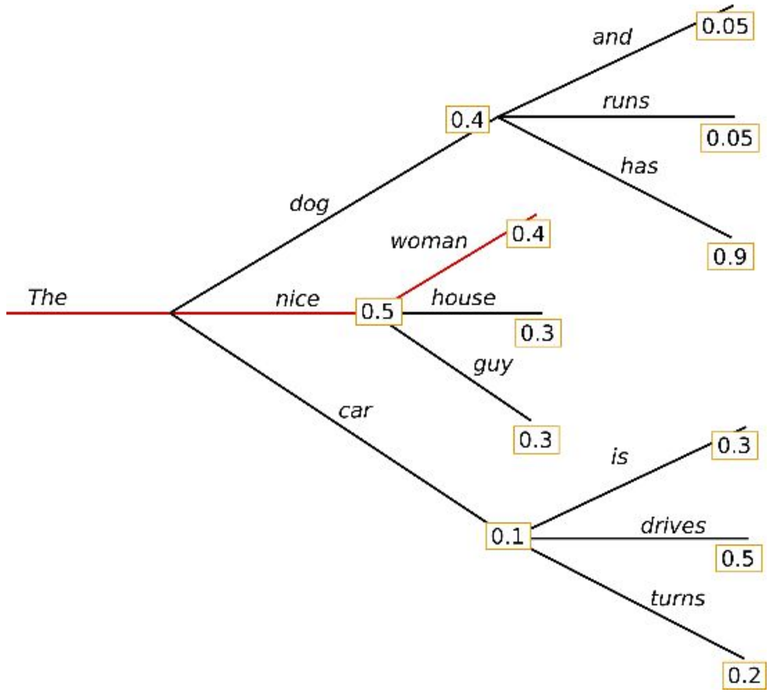
STOCHASTIC PARROT BY DEFAULT

- It merely predicts the most probable token.
- Limit to context window and no real memory systems.
- Will hallucinate, follow distribution of training data.

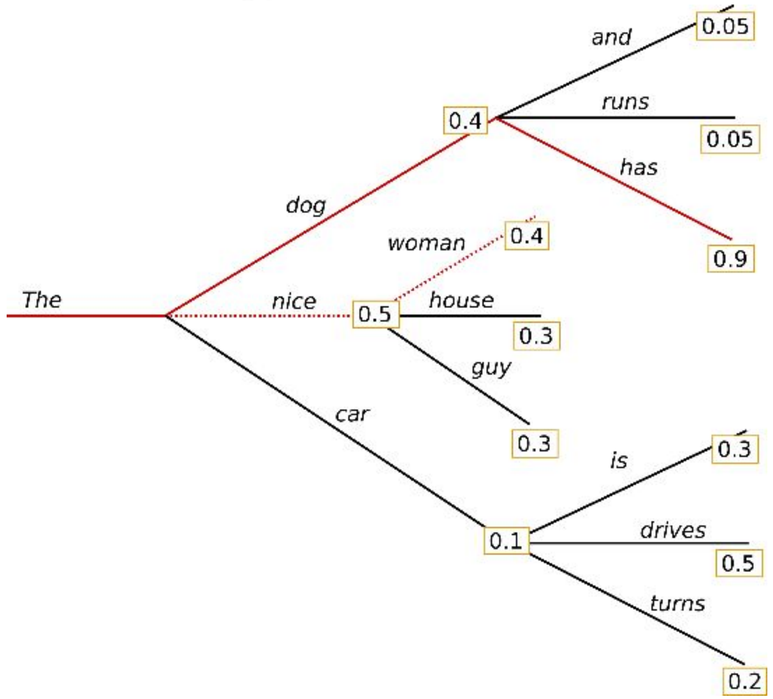
Sampling Strategies

For Non-Greedy Non-Deterministic Parrots

Strategy = Greedy



Strategy = Beam Search



Non-Random:

- Temperature = 0
- Top-k = 1
- Top-p = 0

Random:

- Temperature = 0.5
- Top-k = 5
- Top-p = 0.8

https://huggingface.co/blog/how-to-generate

Core Areas To Improve

Why Is Conversation So Much Harder?

NEEDS MORE POWER [Part 2]

- Needs more training, more data, and larger model.
- Maintain context and consistent autoregression.

STOCHASTIC PARROT BY DEFAULT

- It merely predicts the most probable token.
- Limit to context window and no real memory systems.
- Will hallucinate, follow distribution of training data.

Stateful LLMs

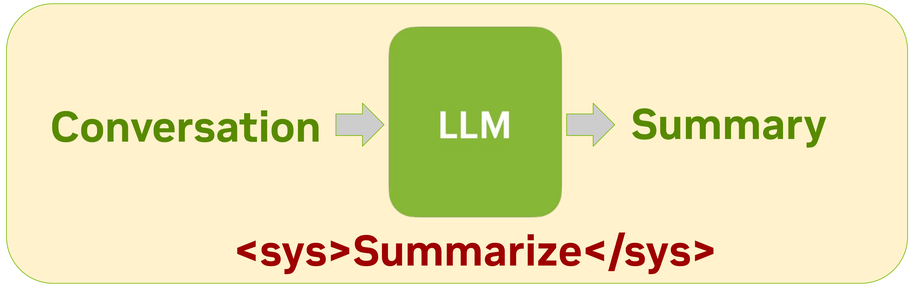
Controlling Generalist LLMs

Stateful LLMs

Controlling Generalist LLMs

Default Chain

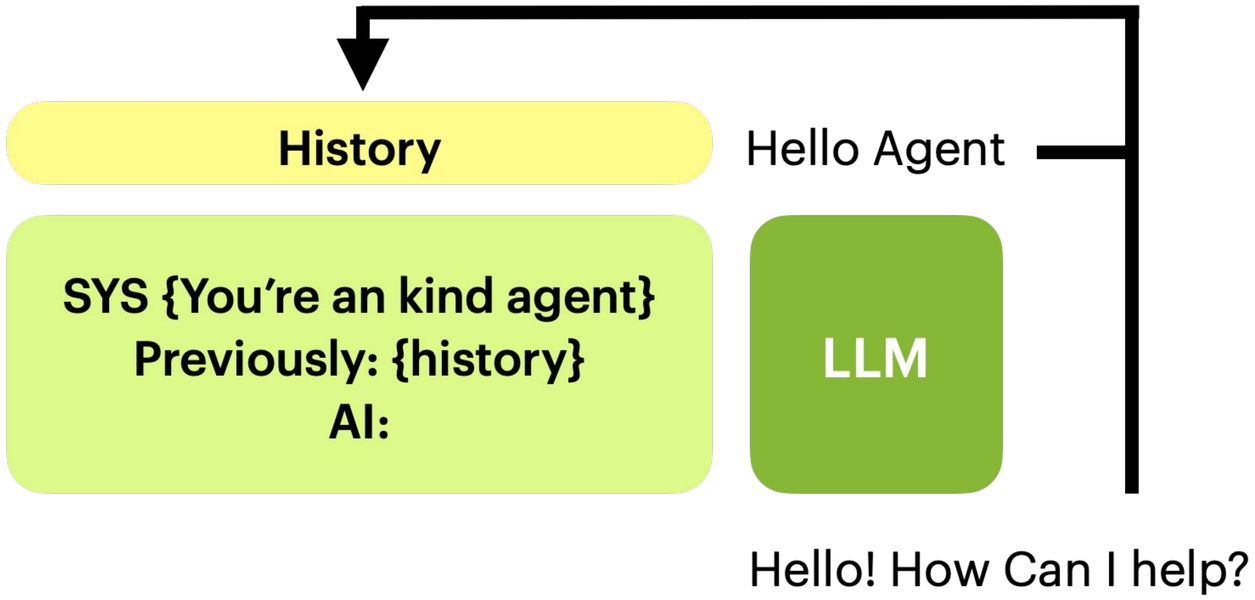
Summarization Chain



Context Chain (Conversation, etc.)

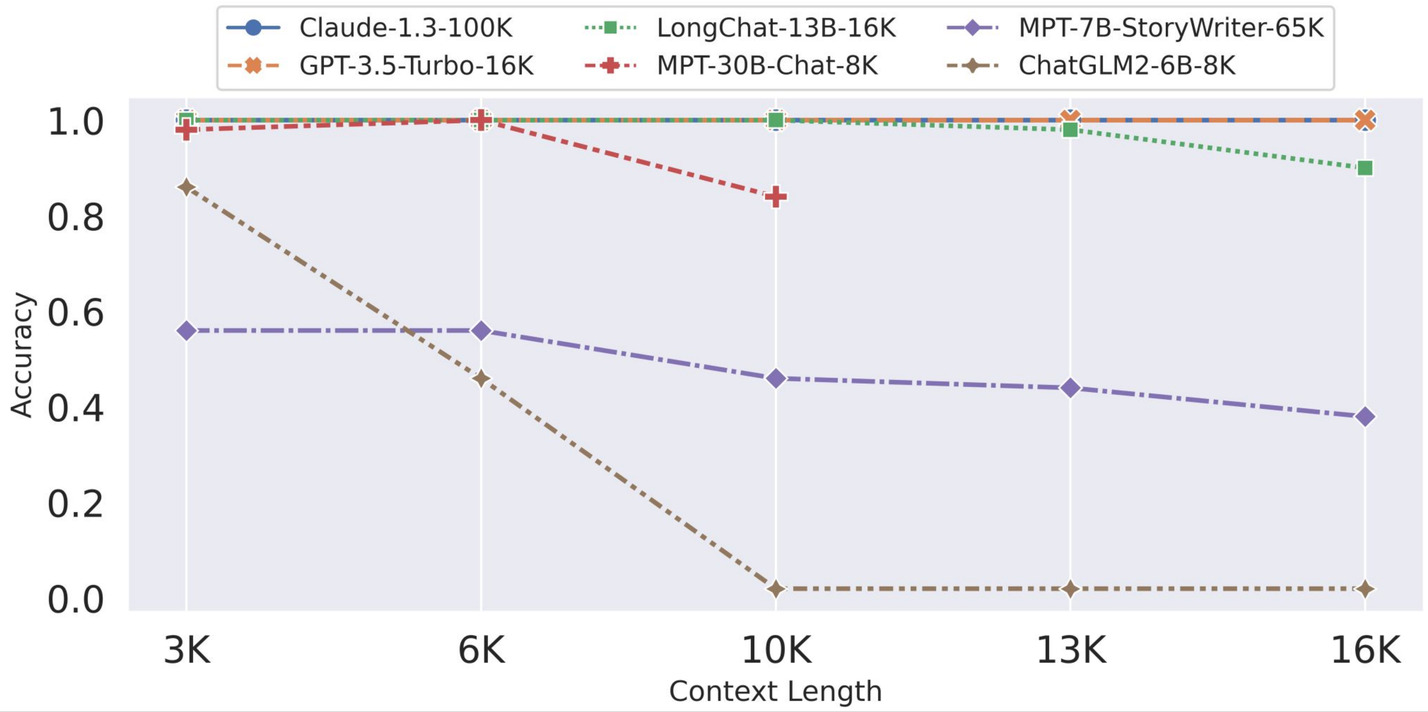
Context Injection

Controlling Generalist LLMs



Context Limits

Enemy of Context Injection



Modern LLMs Have ~32K Limits

- Llama-2 with fine-tuning
- GPT4 full is similar
- **Will continue going up!**

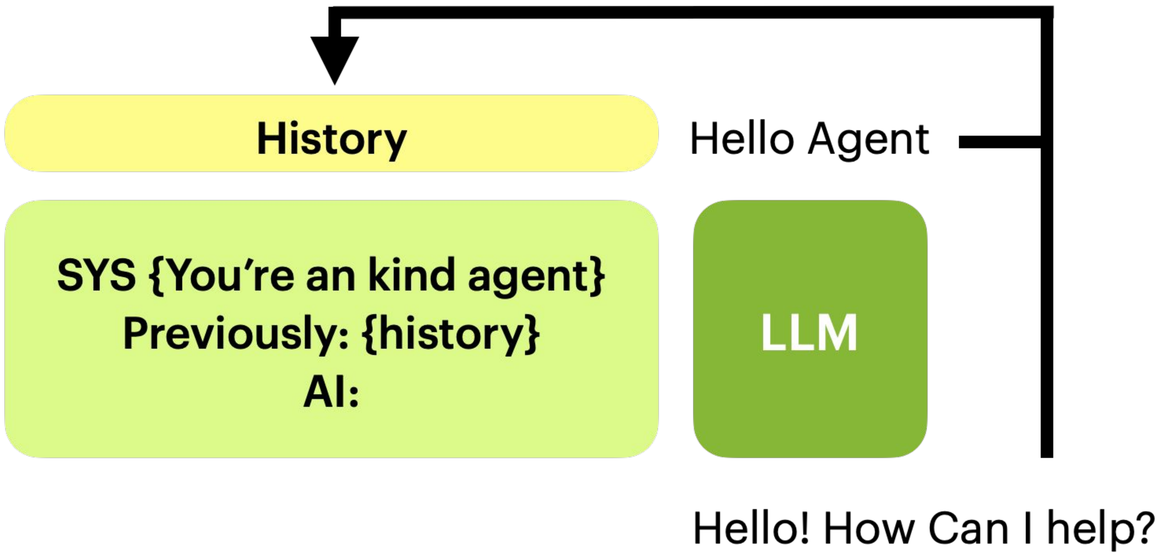
https://lmsys.org/blog/2023-06-29-longchat/

Context Injection

Controlling Generalist LLMs

- **Conversation Buffers**
- Summarization Buffers
- Structured Output Buffers
- Retrieval Buffers

Stateful Model	**Multiple Stateful Models**
Structured Output Model	**Tooling-Assisted Models**

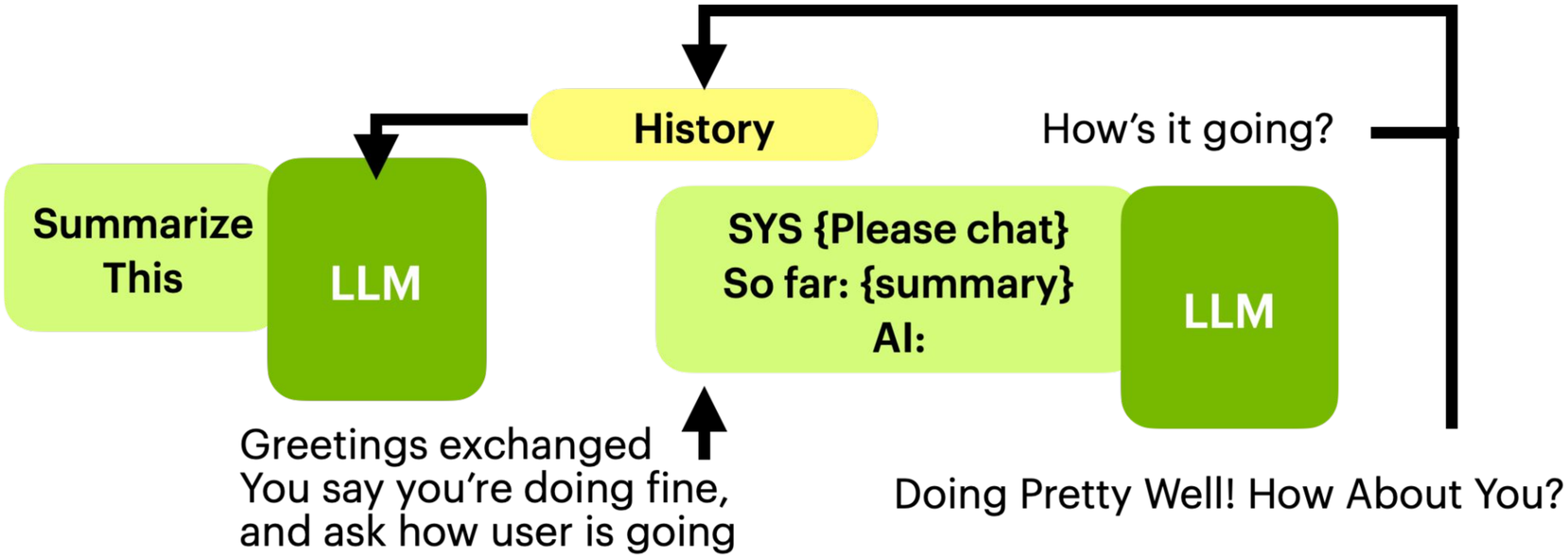


Context Injection

Controlling Generalist LLMs

- Conversation Buffers
- **Summarization Buffers**
- Structured Output Buffers
- Retrieval Buffers

Stateful Model	**Multiple Stateful Models**
Structured Output Model	**Tooling-Assisted Models**

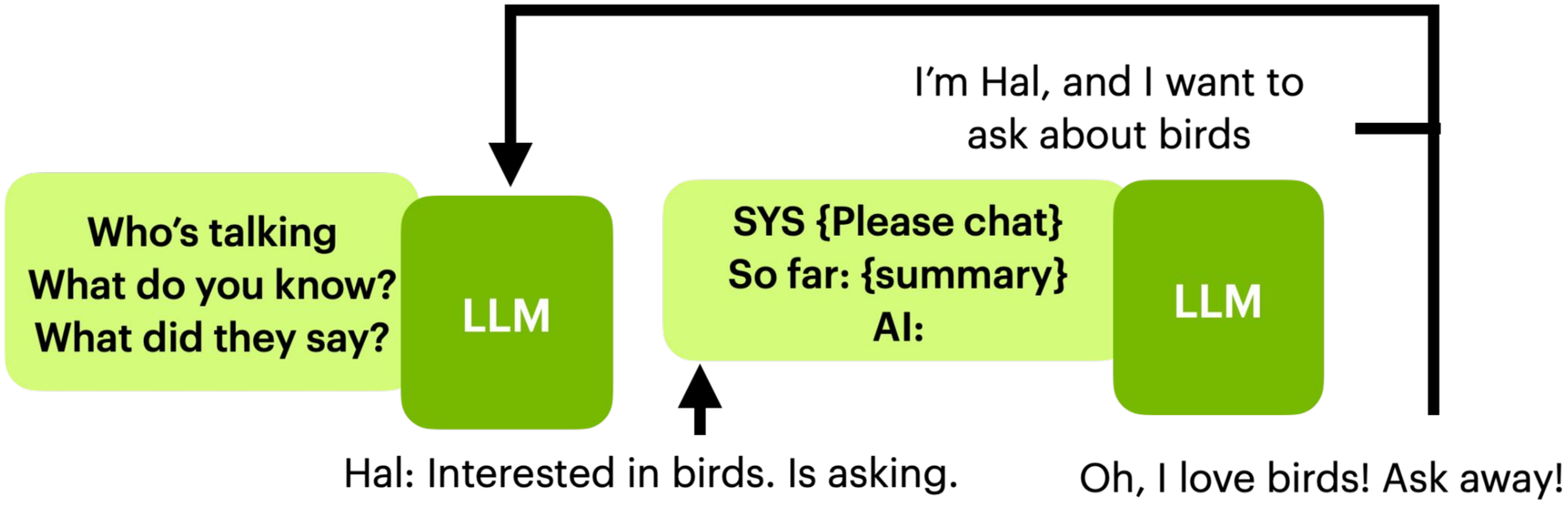


Context Injection

Controlling Generalist LLMs

- Conversation Buffers
- Summarization Buffers
- **Structured Output Buffers**
- Retrieval Buffers

Stateful Model	**Multiple Stateful Models**
Structured Output Model	**Tooling-Assisted Models**

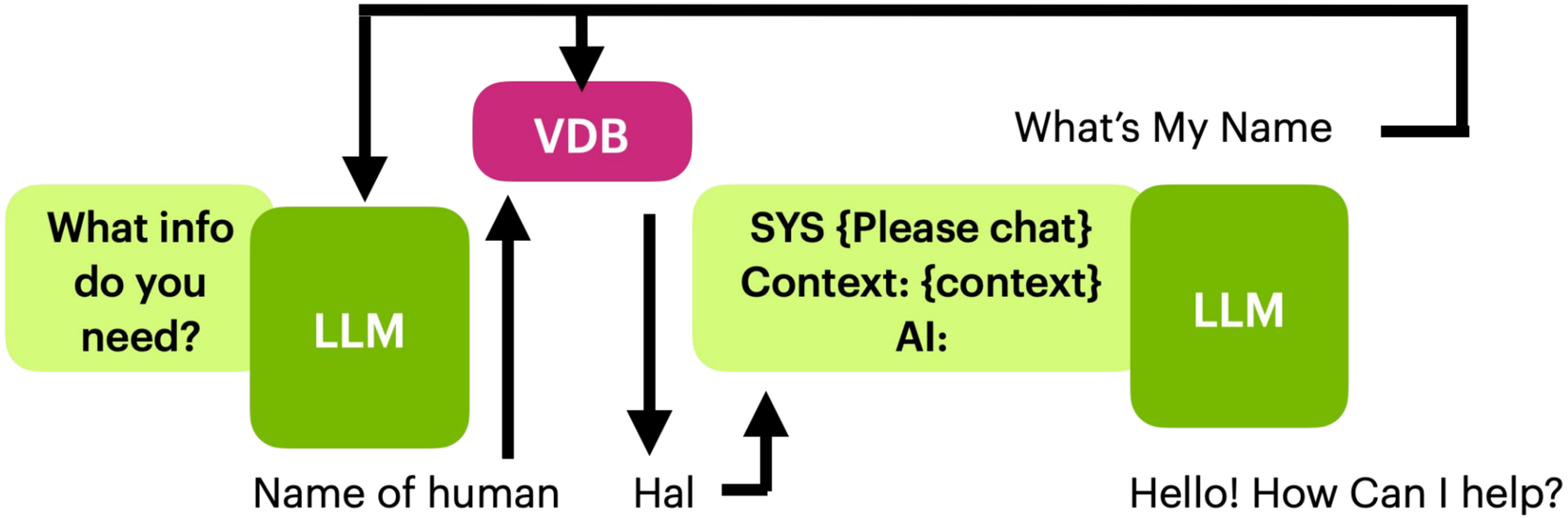


Context Injection

Controlling Generalist LLMs

- Conversation Buffers
- Summarization Buffers
- Structured Output Buffers
- **Retrieval Buffers**

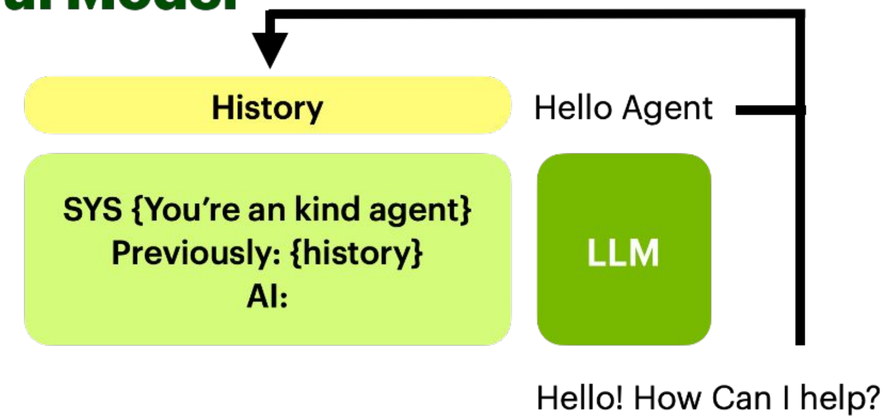
Stateful Model	**Multiple Stateful Models**
Structured Output Model	**Tooling-Assisted Models**



Stateful LLM Highlights

Main Working Paradigms

Stateful Model



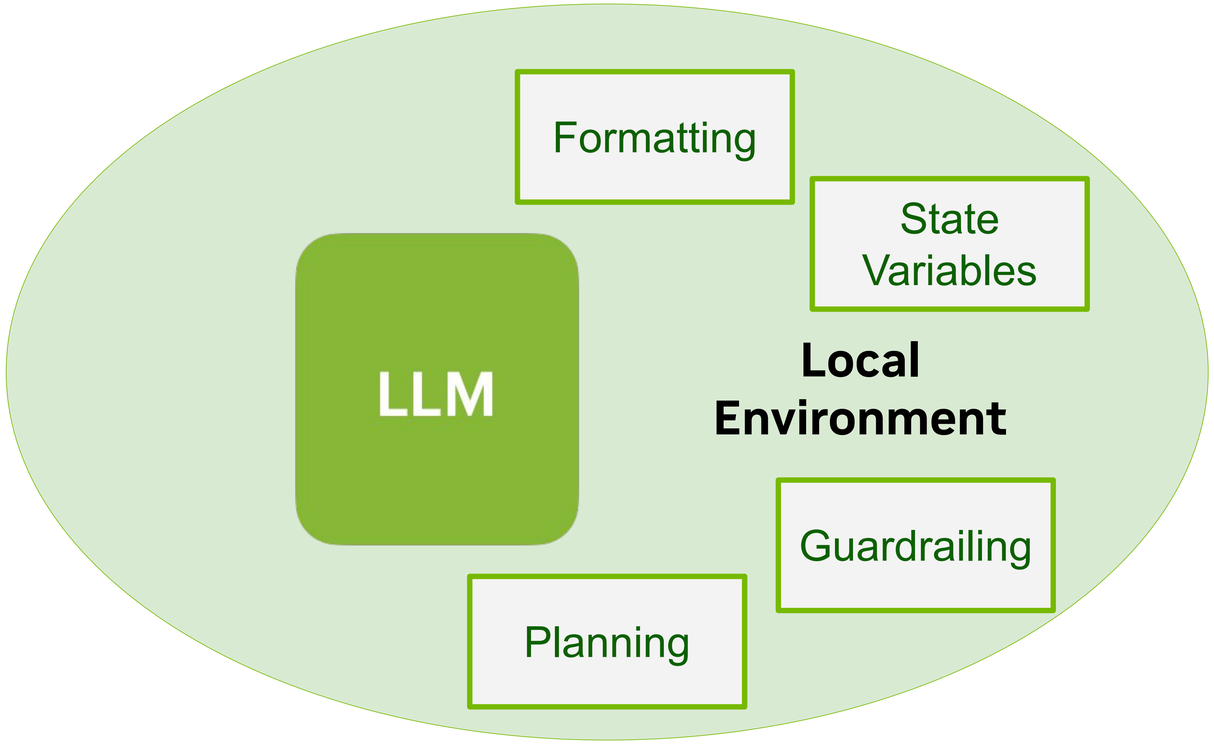
Multiple Stateful Models

Structured Output Model

Tooling-Assisted Models

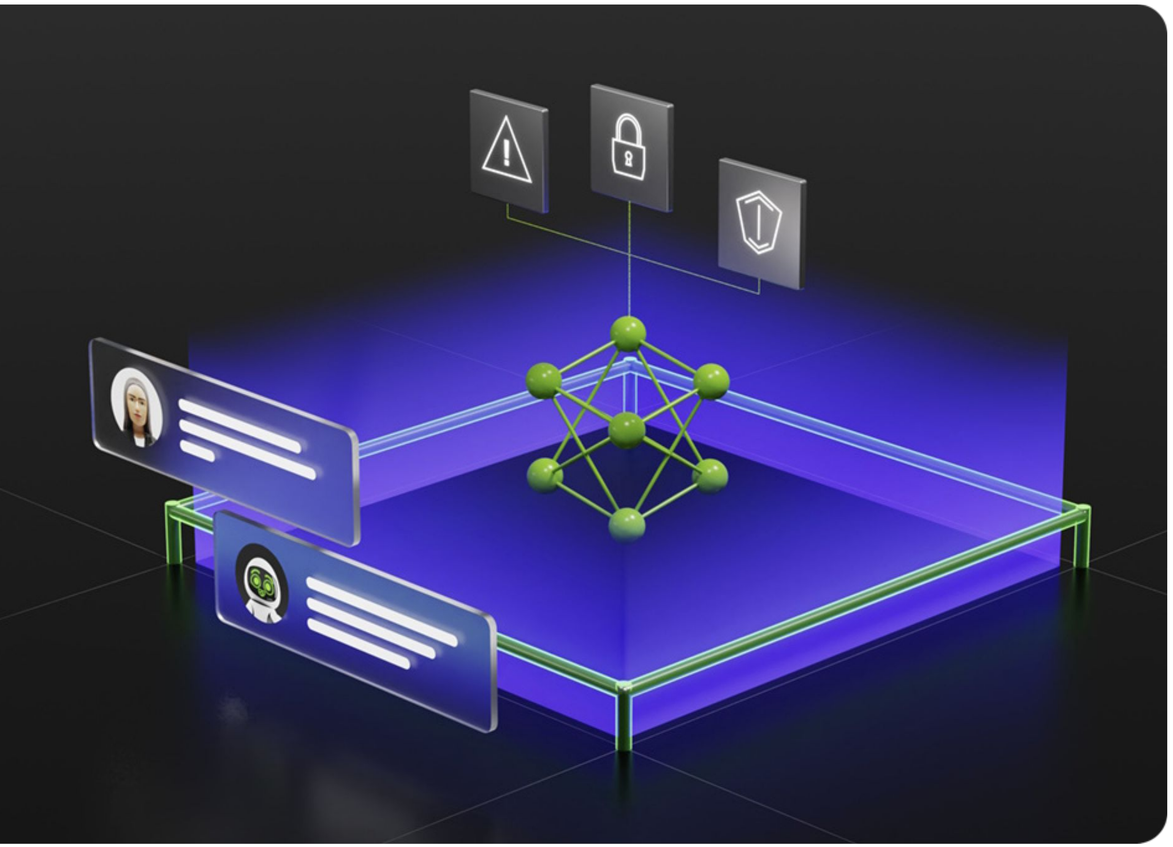
LLM Workspace

Local Environment Working Paradigms



LLM Workspace

Local Environment Working Paradigms



State Variables

- Fill-in-the-blank question-answering to context

Formatting

- Automatic rephrasing/pre-processing

Guardrailing

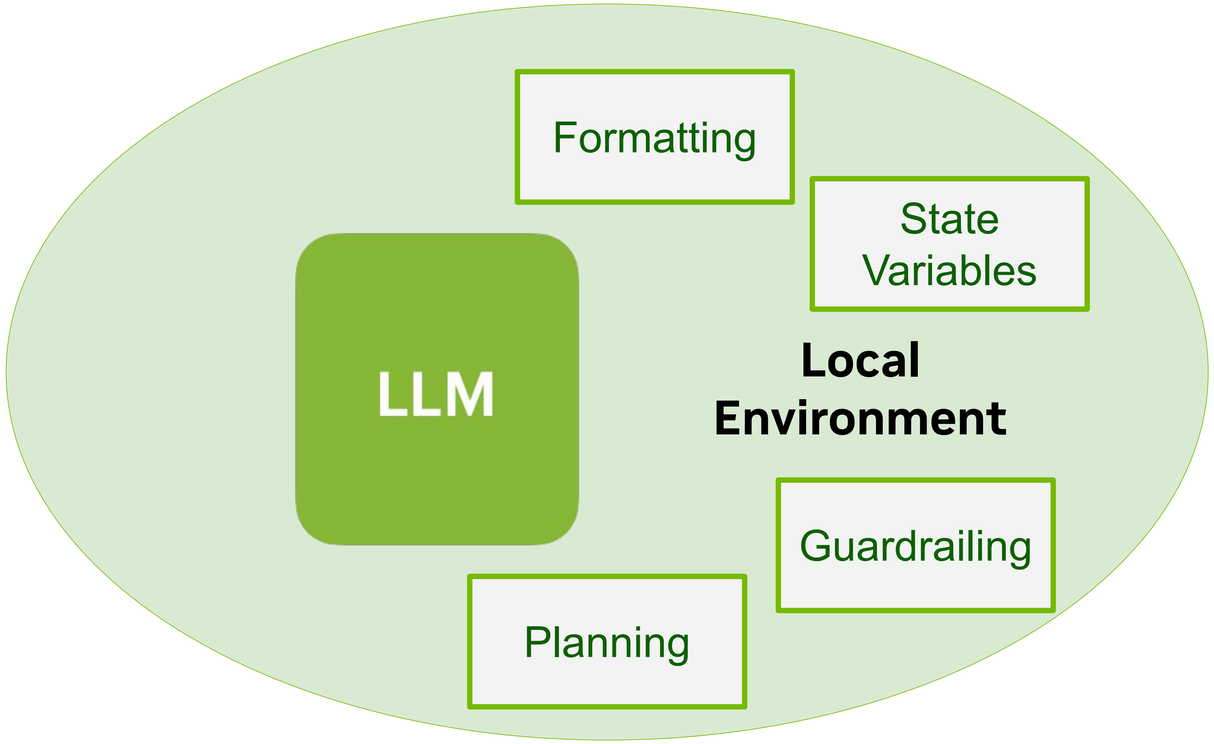
- Rejecting/Re-Querying to stay on track.

Planning

- Make a plan, follow it, and explain the logic

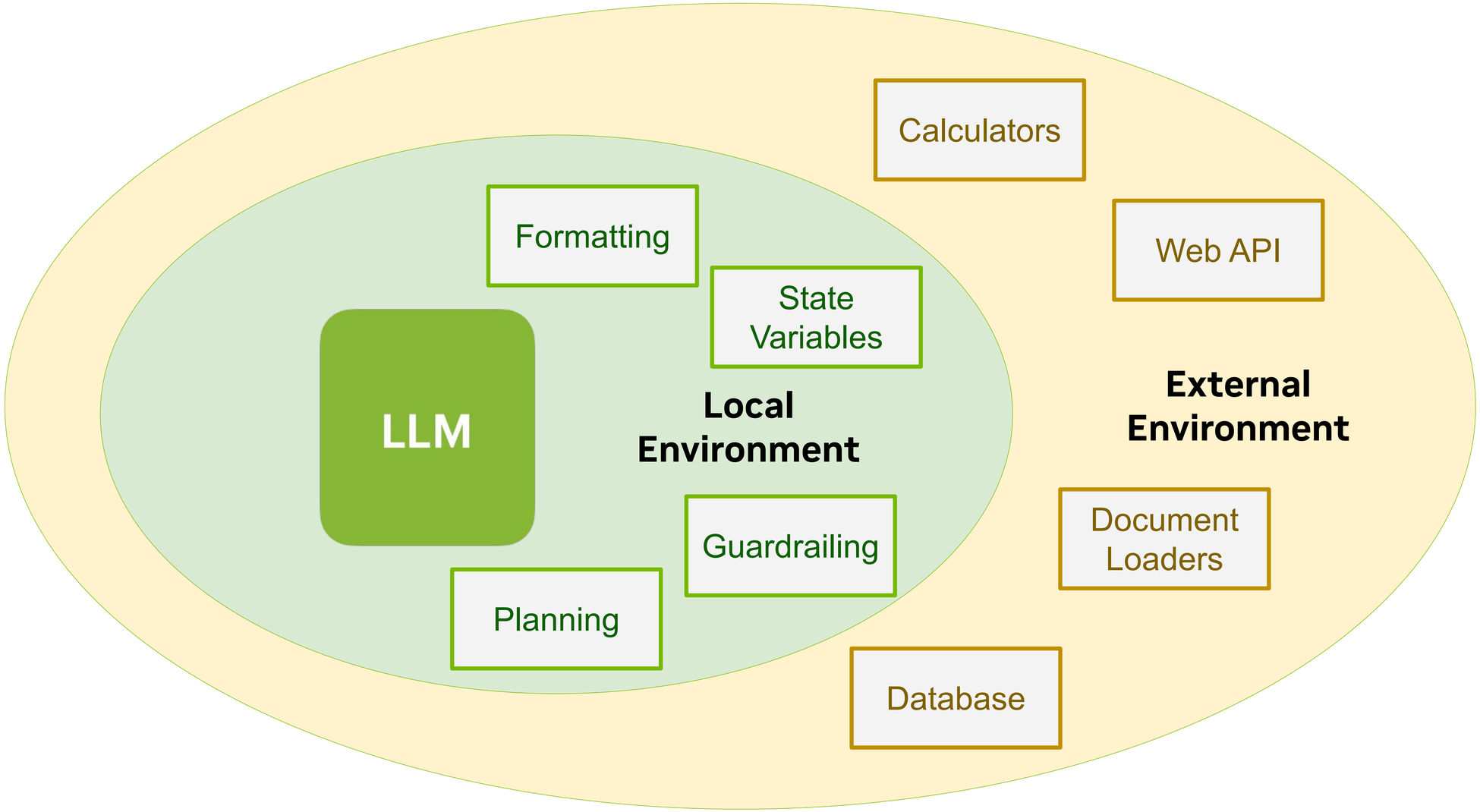
LLM Workspace

Local Environment Working Paradigms



LLM Workspace

Main Working Paradigms



LLM Workspace

Main Working Paradigms

Documents

- Loaders to convert document to context
- Pipe document content to database

Calculator/Web API

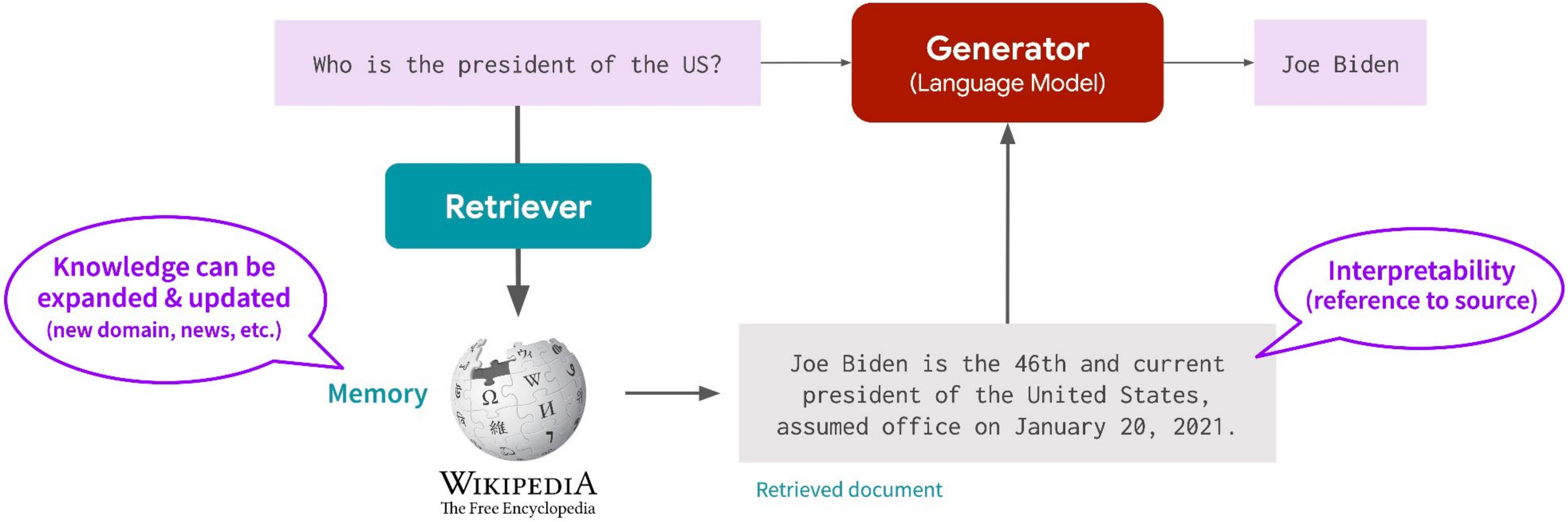
- Manually-specified algorithms
- Predict tool selection/parameters

Databases

- **[Simple]** Predict database query, run command
- **[Vector]** Text Embedding Similarity Search

Retrieval-Augmented Generation

Supporting New Information

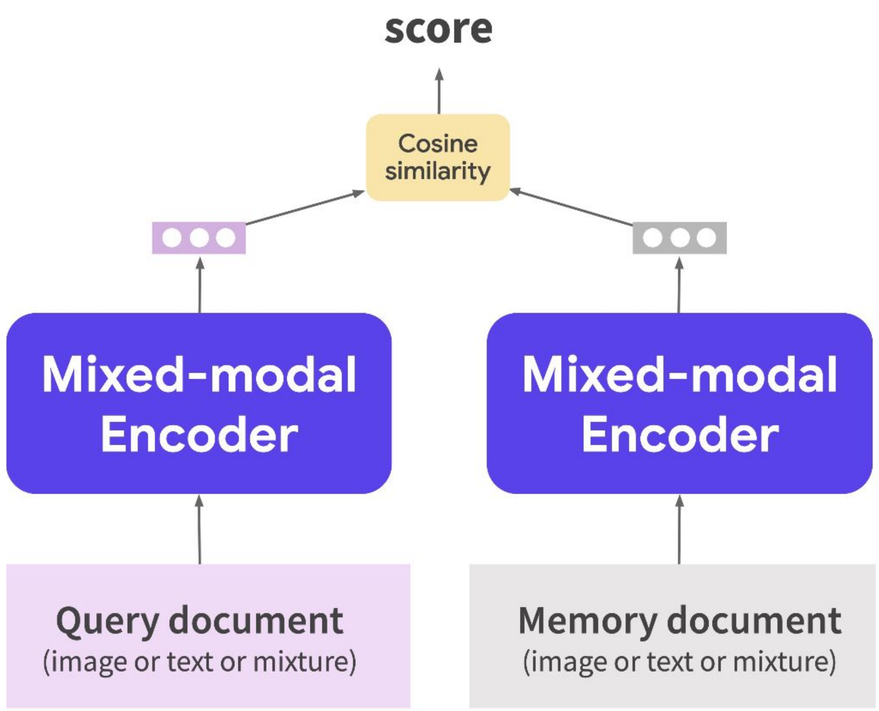


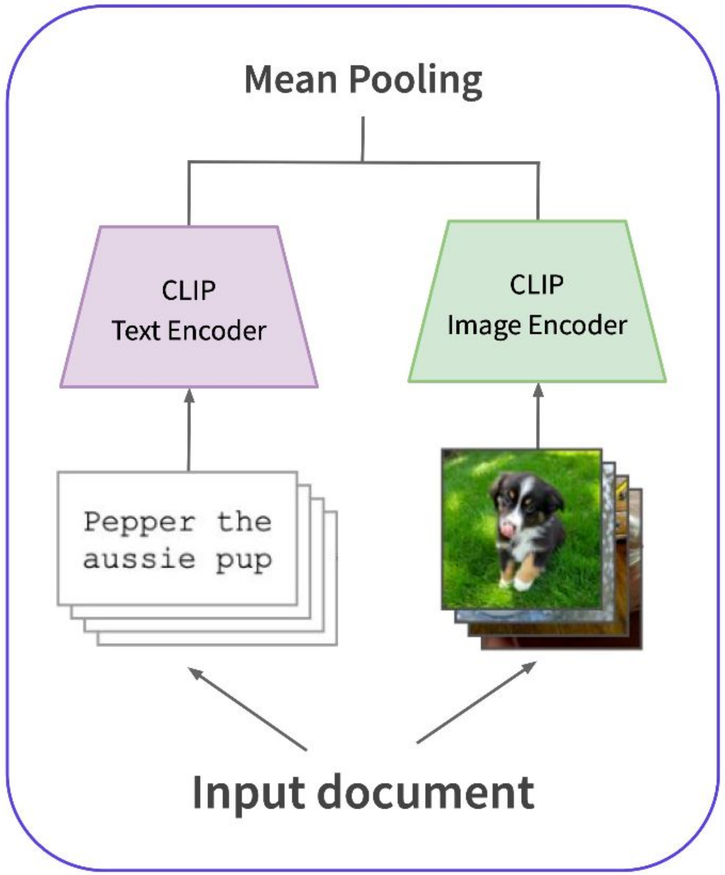
https://cs.stanford.edu/~myasu/blog/racm3/

Vector Databases

Retrieving Documents with Embeddings

```
f(query, memory) ⟶ score
```

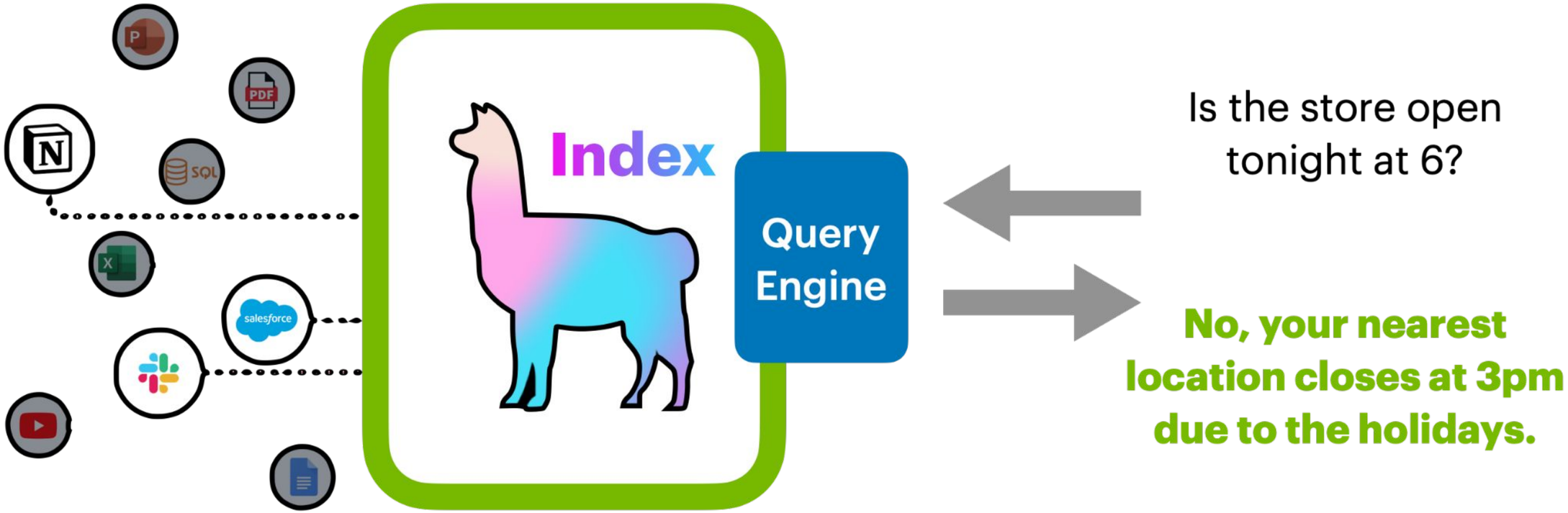




https://cs.stanford.edu/~myasu/blog/racm3/

LlamaIndex

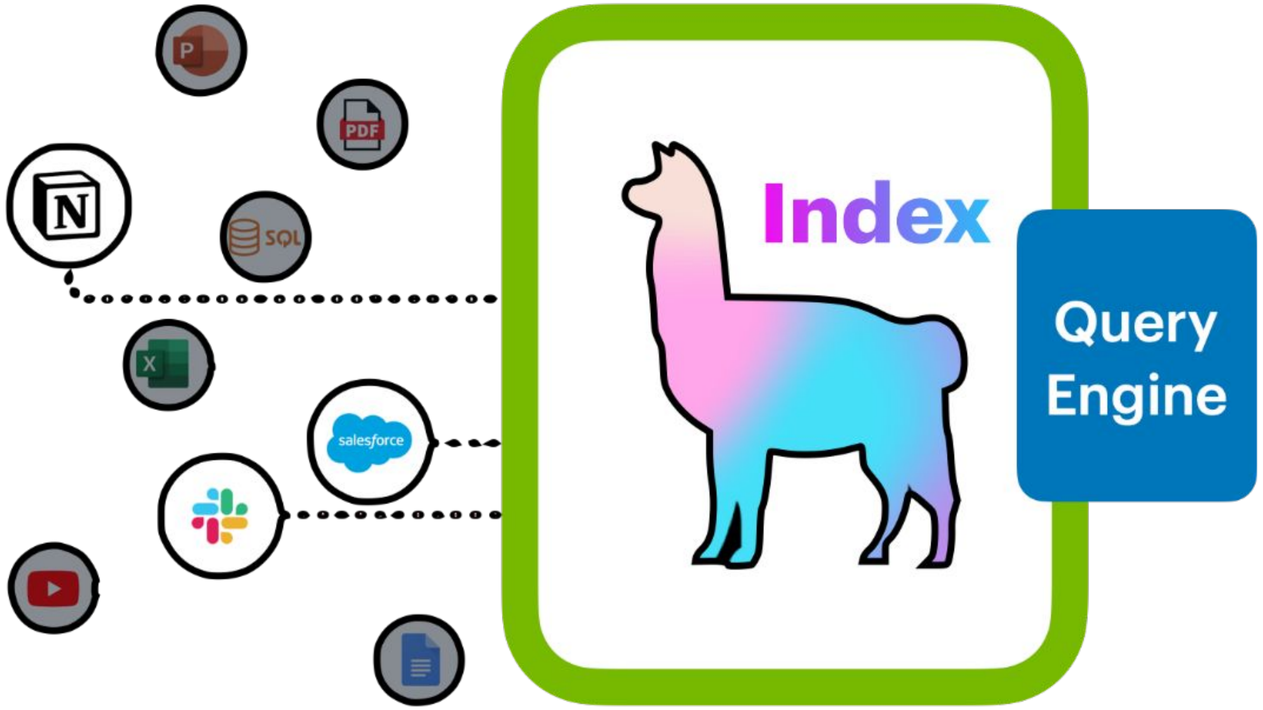
Document Loading and Indexing



https://www.llamaindex.ai

LlamaIndex

Document Loading and Indexing



How much GPU compute do I need for Llama 2?

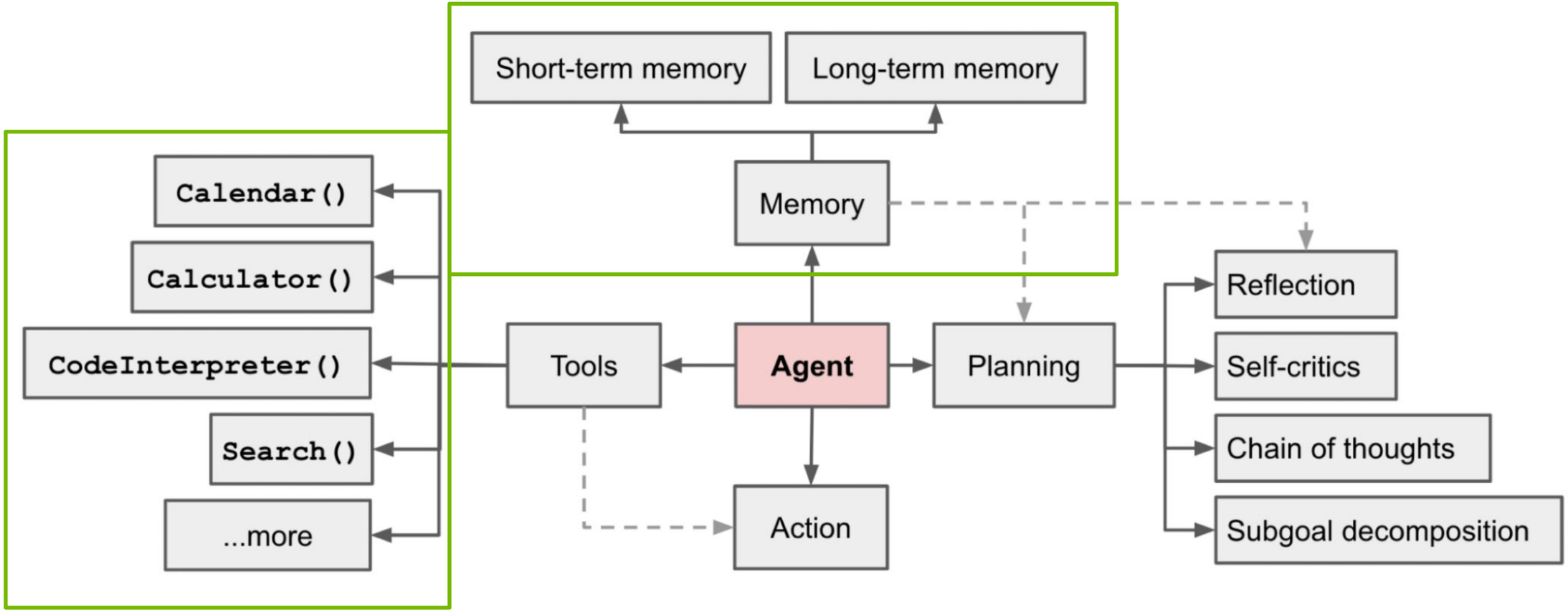
"But how much VRAM is required to ..."

$Q \rightarrow Q'$	Symmetric
$Q \rightarrow R$	Asymmetric
$[Q, Q_1, Q_2, \cdots] \rightarrow [R, R_1, R_2, \cdots]$	Fusion

https://www.llamaindex.ai

LLM Workspace

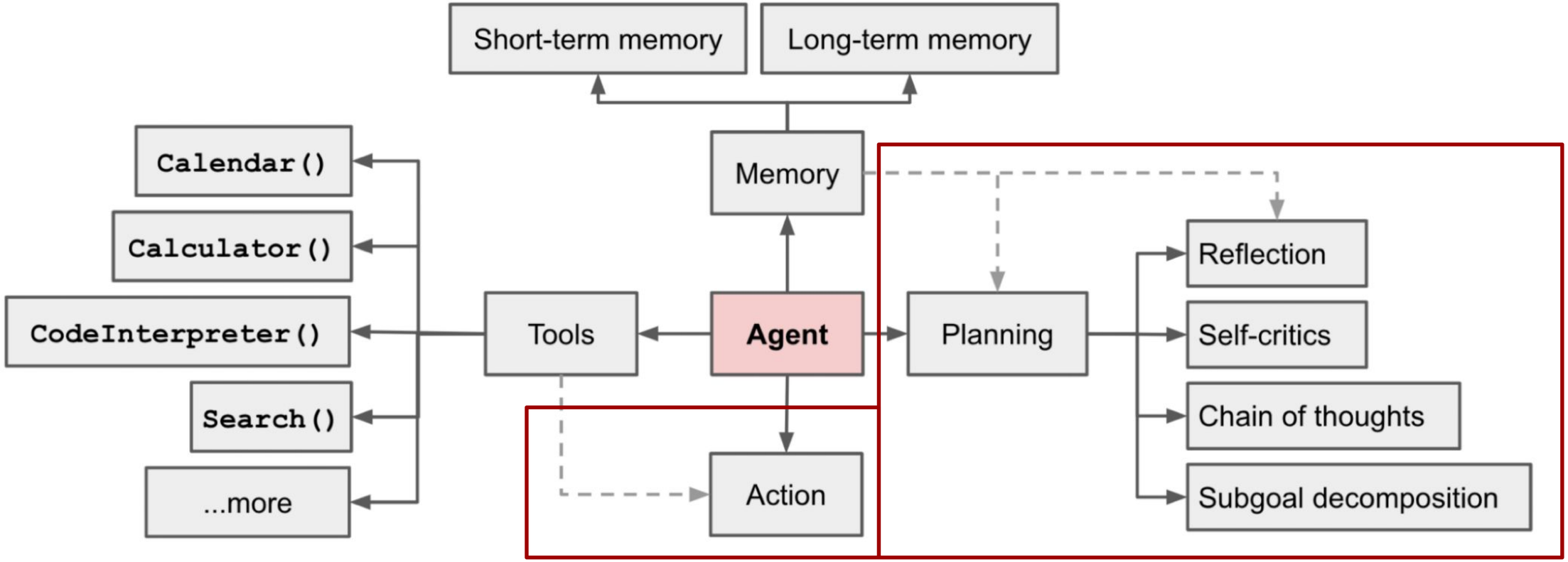
The Big Picture



https://lilianweng.github.io/posts/2023-06-23-agent/

LLM Workspace

The Big Picture



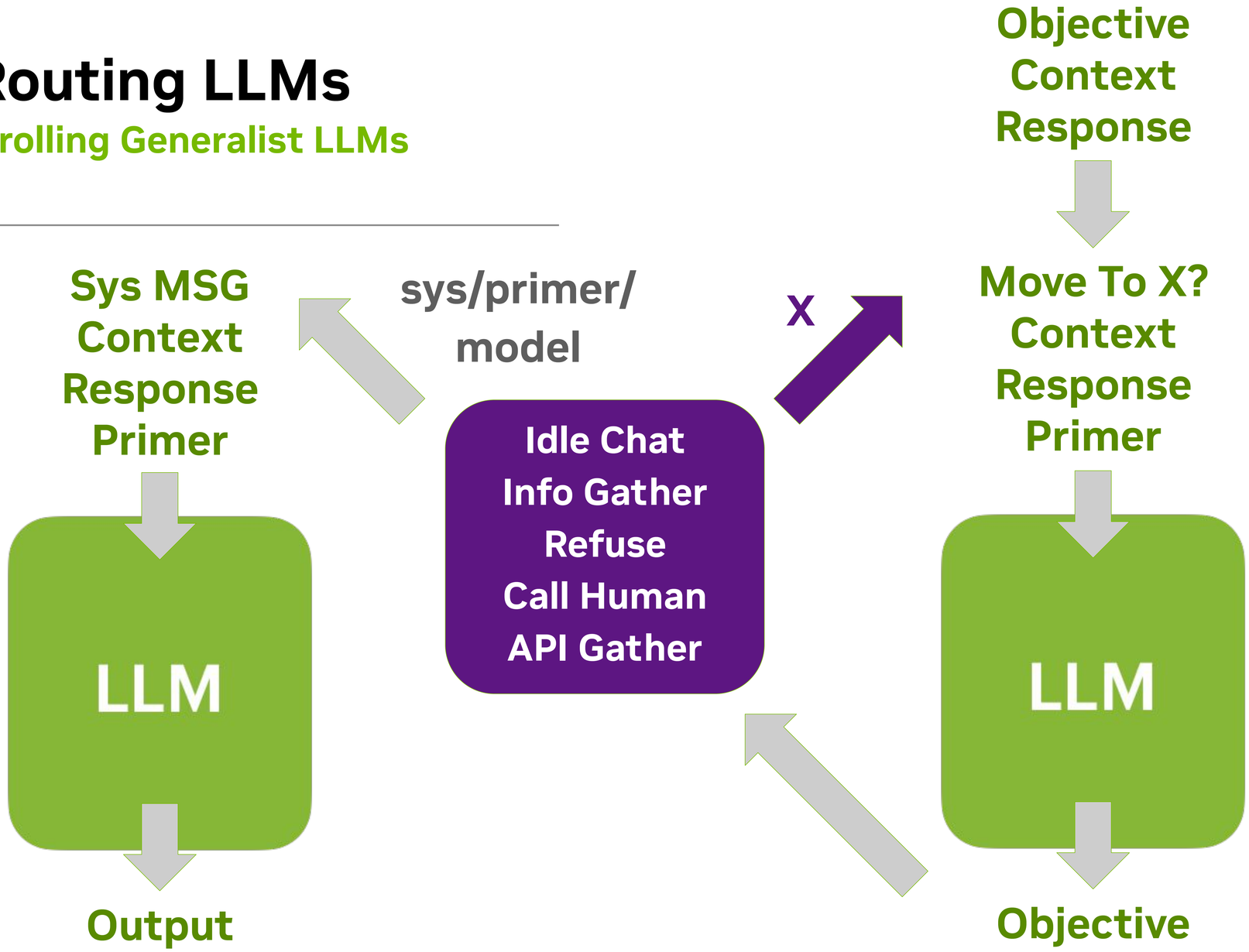
https://lilianweng.github.io/posts/2023-06-23-agent/

Routing LLMs

Controlling Generalist LLMs

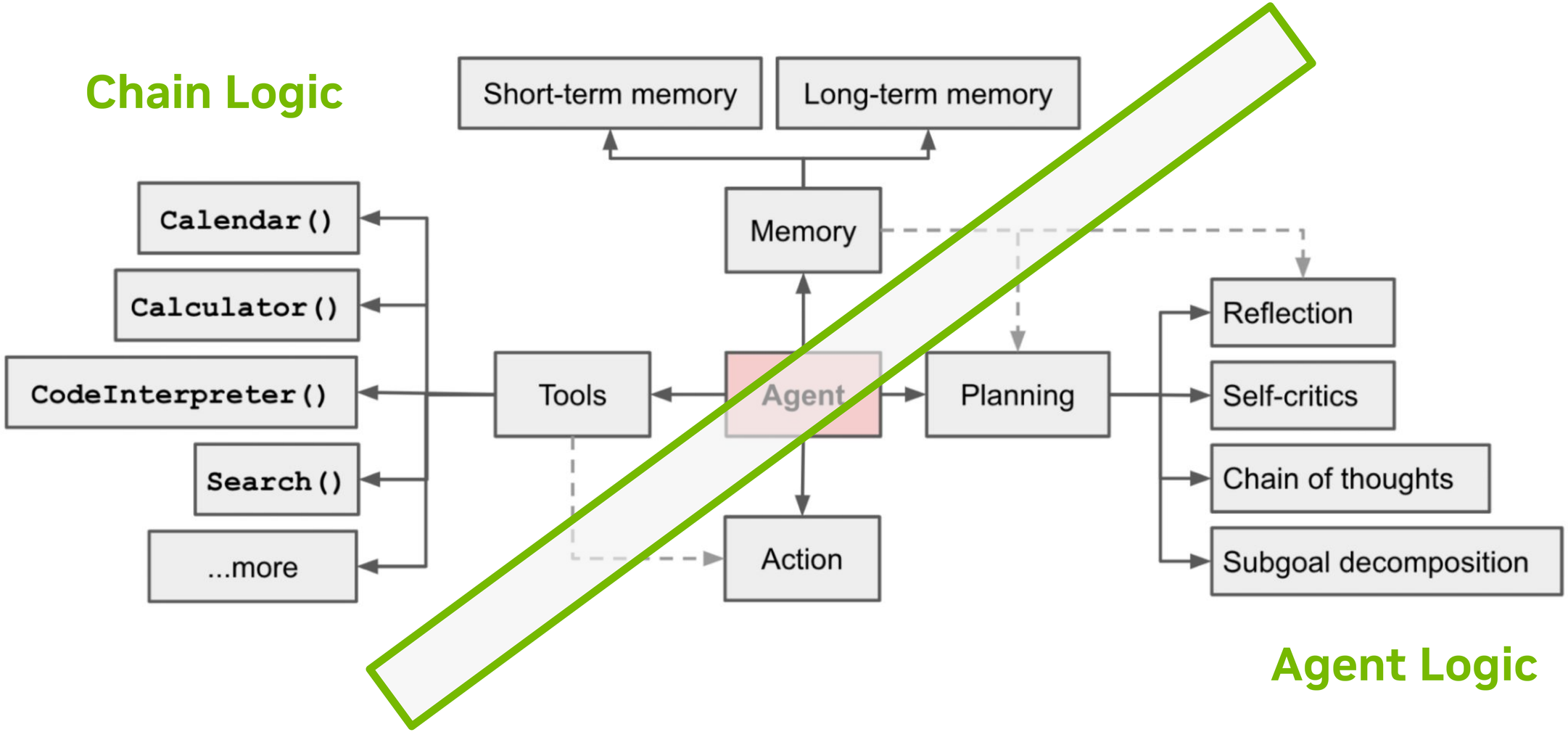
Routing LLMs

Controlling Generalist LLMs



LLM Workspace

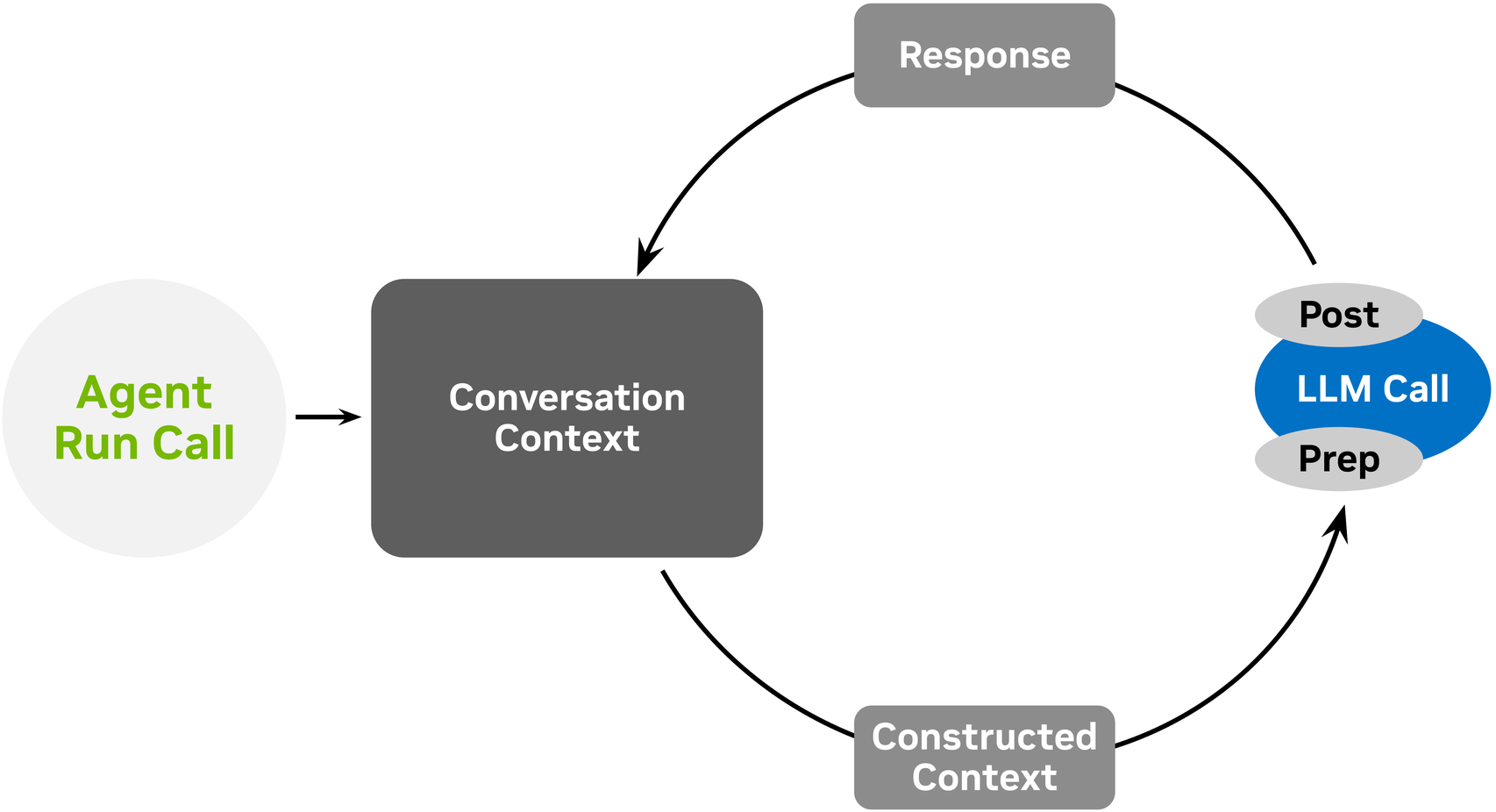
The Big Picture



https://lilianweng.github.io/posts/2023-06-23-agent/

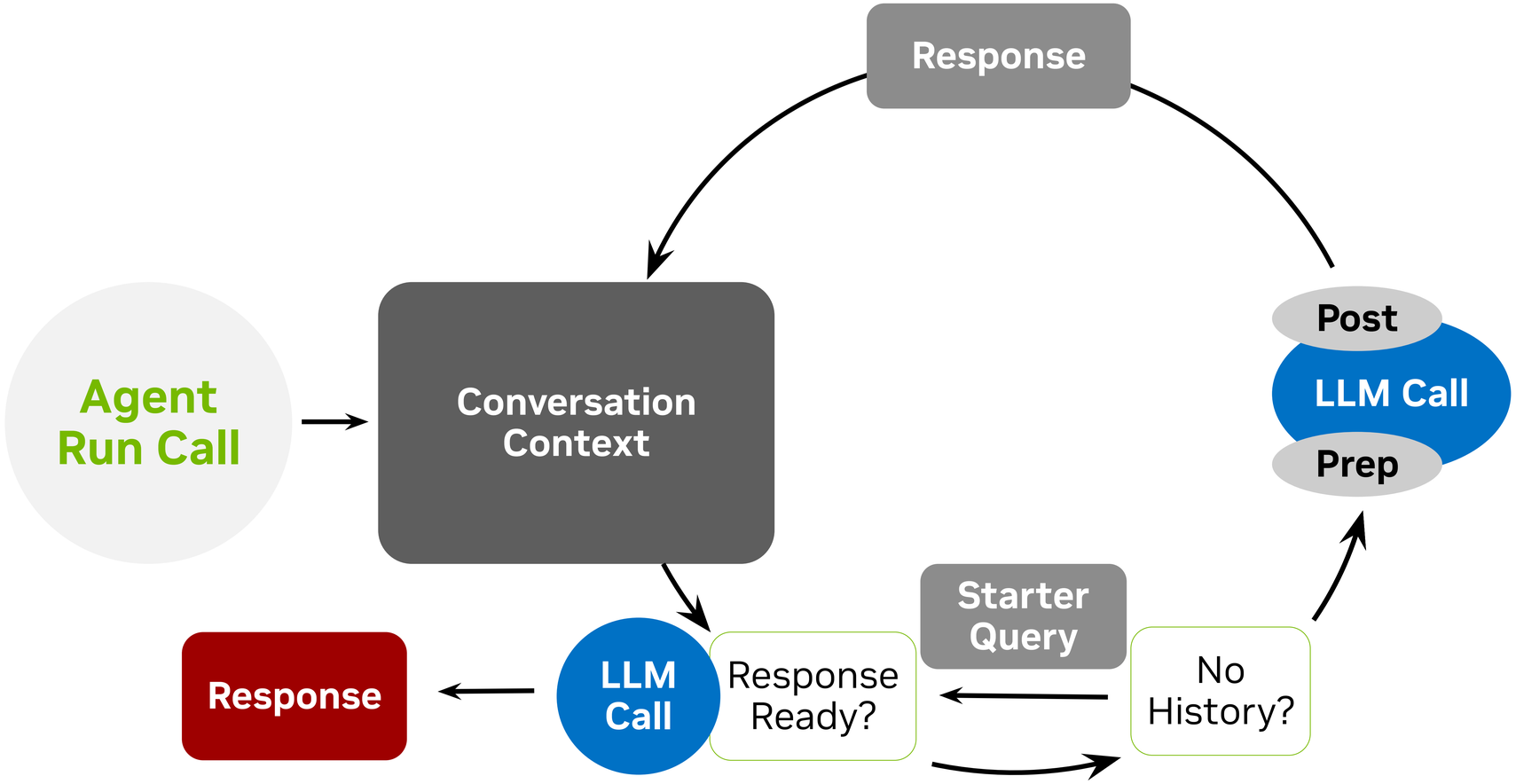
General Agent Event Loop

Making The Agent Think



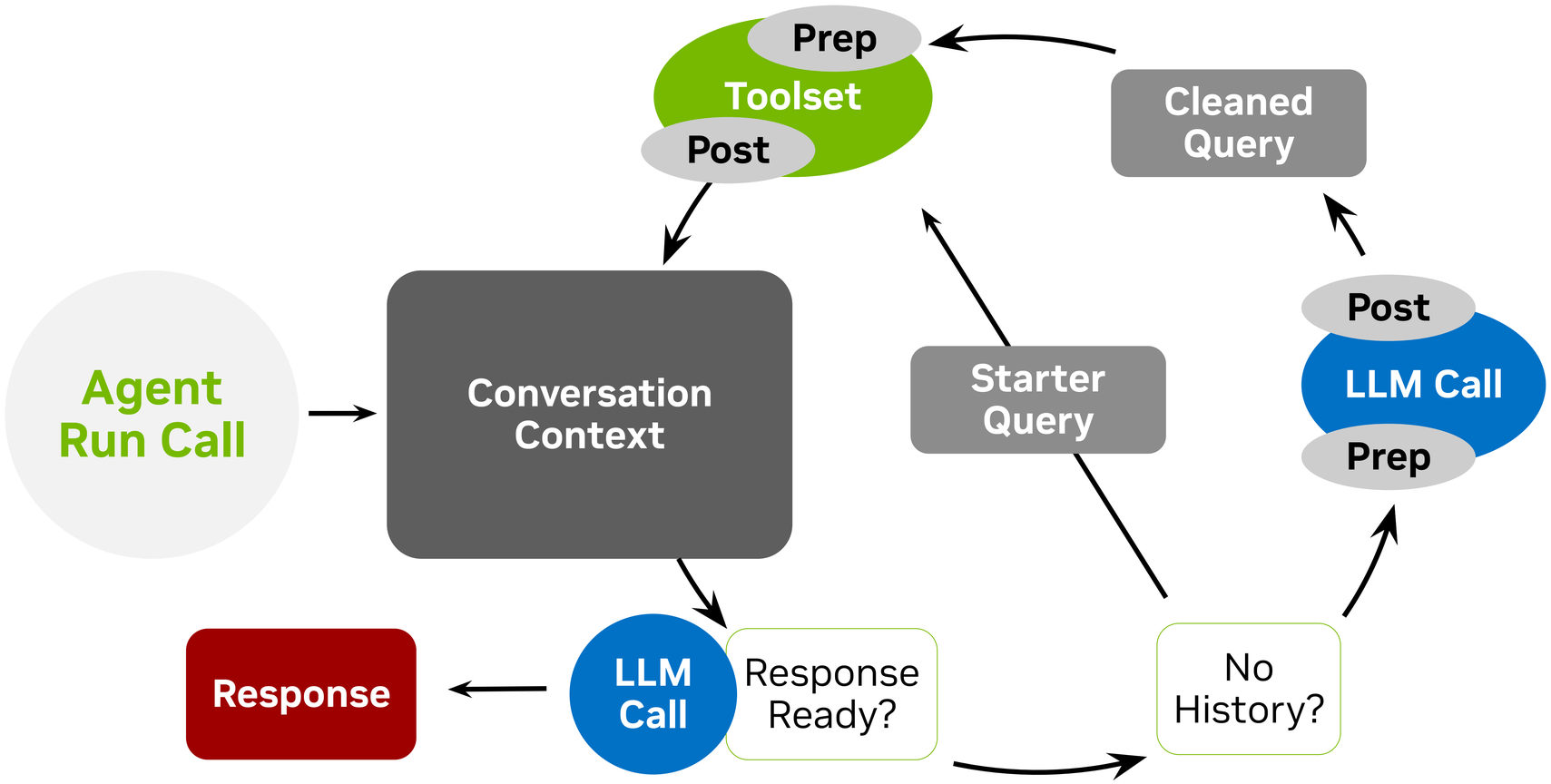
General Agent Event Loop

Letting The Agent Speak



General Agent Event Loop

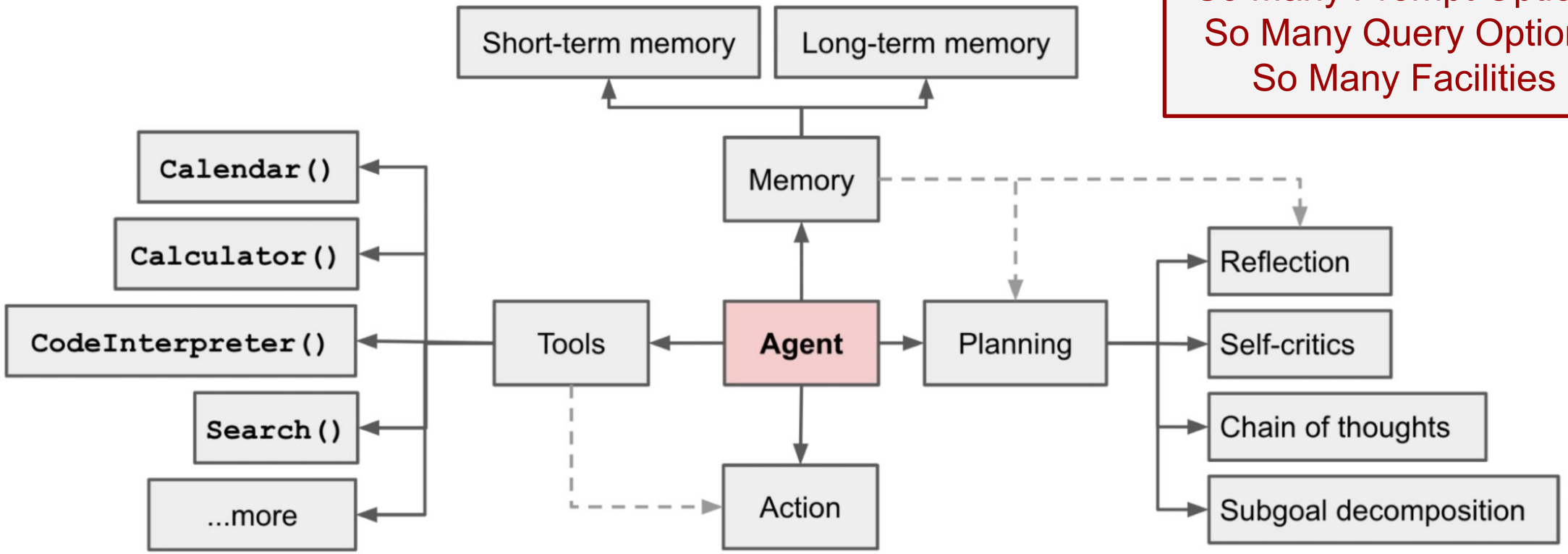
Giving The Agent Tools



LLM Workspace

The Big Picture

So Many Prompt Options
So Many Query Options
So Many Facilities



https://lilianweng.github.io/posts/2023-06-23-agent/

LangChain

Chaining Stochastic Parrots

https://www.langchain.com/

LangChain

Chaining Stochastic Parrots

In Theory

Many LLM Wrappers

So Many Capabilities

In Practice

Requires Customization

Assumes Very Powerful Model

https://www.langchain.com/

Good Practices

Things To Remember In The Wild

Using LangChain

How Does This Actually Work?

Does My Model Actually Work With This?

How Much Will It Cost Me?

Roll Your Own Scheme

How Do Other Composers Do It?

Where Can I Reuse Other's Components?

Where Can I Impose Structure?

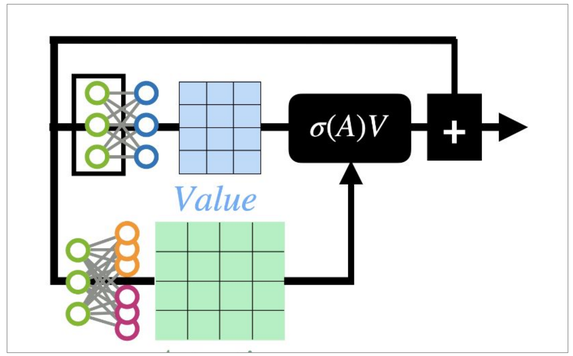
Part 3: Stateful Systems

- Controlling Stateful Agents
- Final Project: Making An Agent

Course Schedule

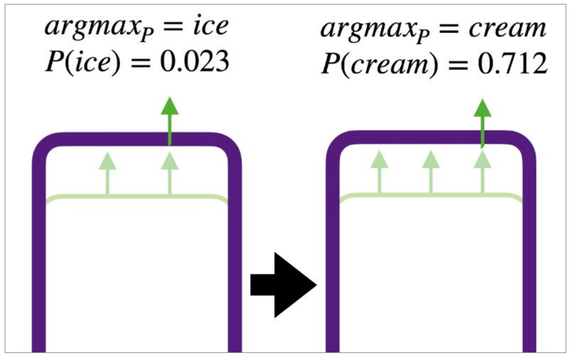
What's The Plan?

Getting Started

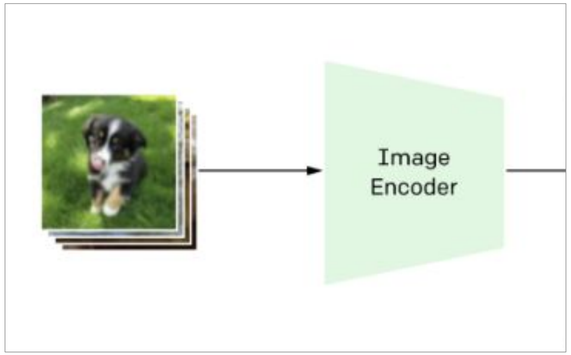


Transformer Intuitions

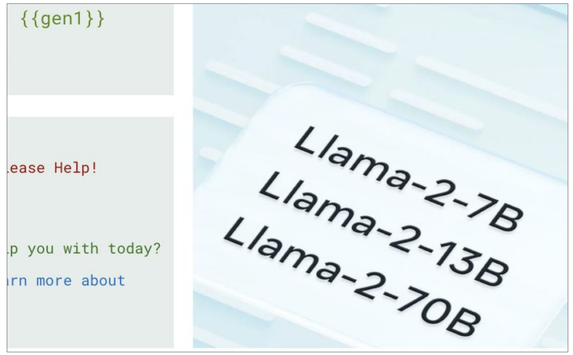
Encoder Tasks



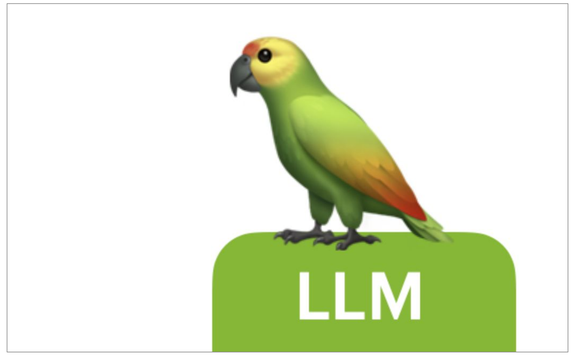
Seq2Seq



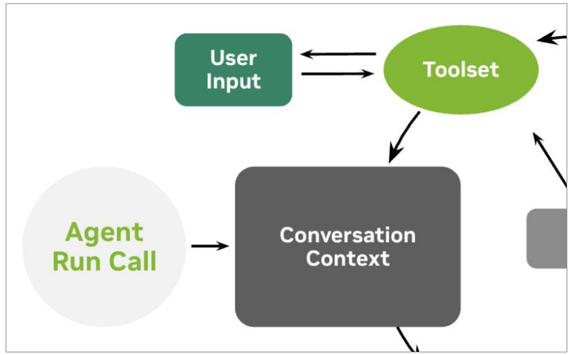
Multimodality



Text Generation



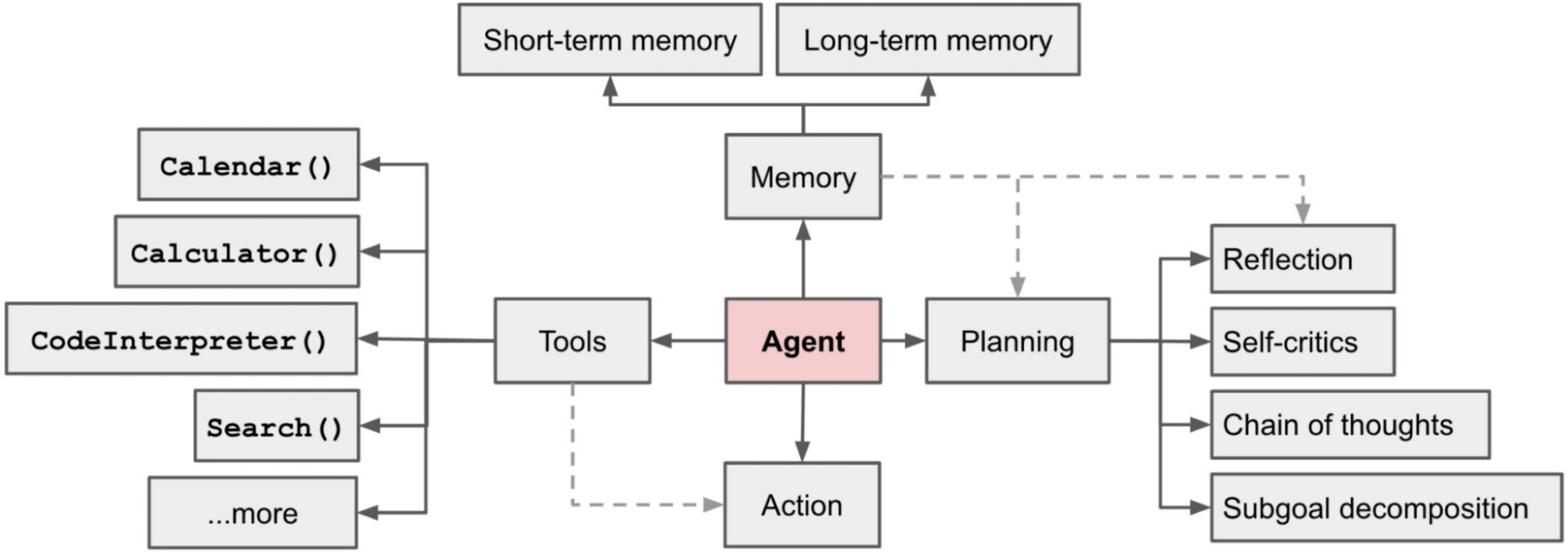
Stateful LLMs



Agent Assessment

LLM Workspace

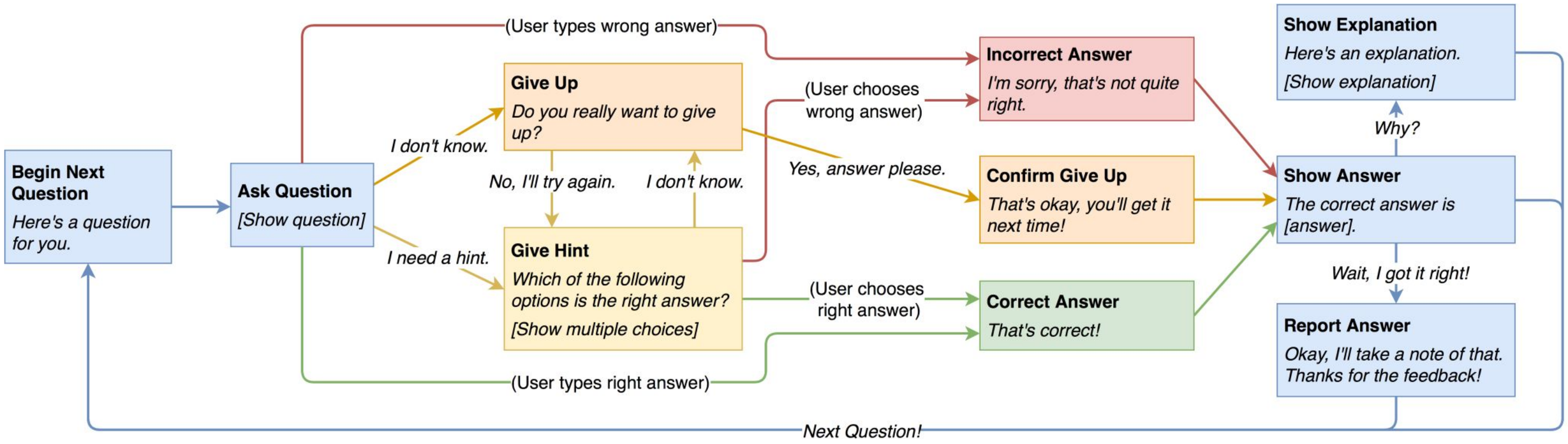
The Big Picture



https://lilianweng.github.io/posts/2023-06-23-agent/

Conversation Flow

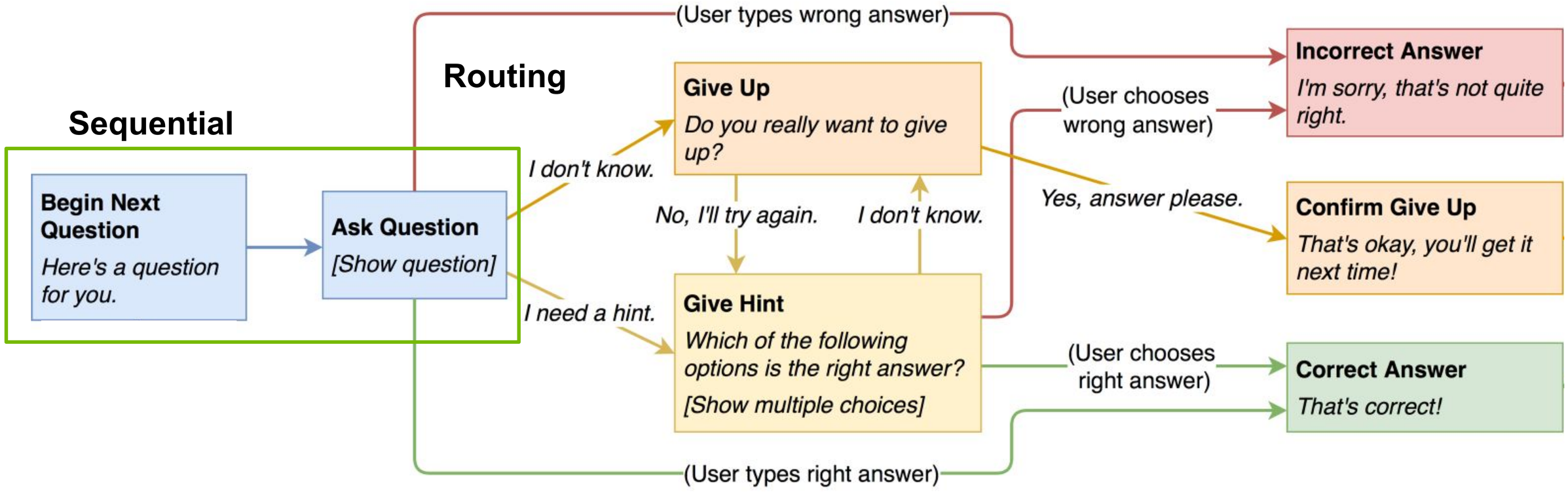
Finite State/Markov Chain Perspective



http://ai.stanford.edu/blog/quizbot/

Conversation Flow

Finite State/Markov Chain Perspective



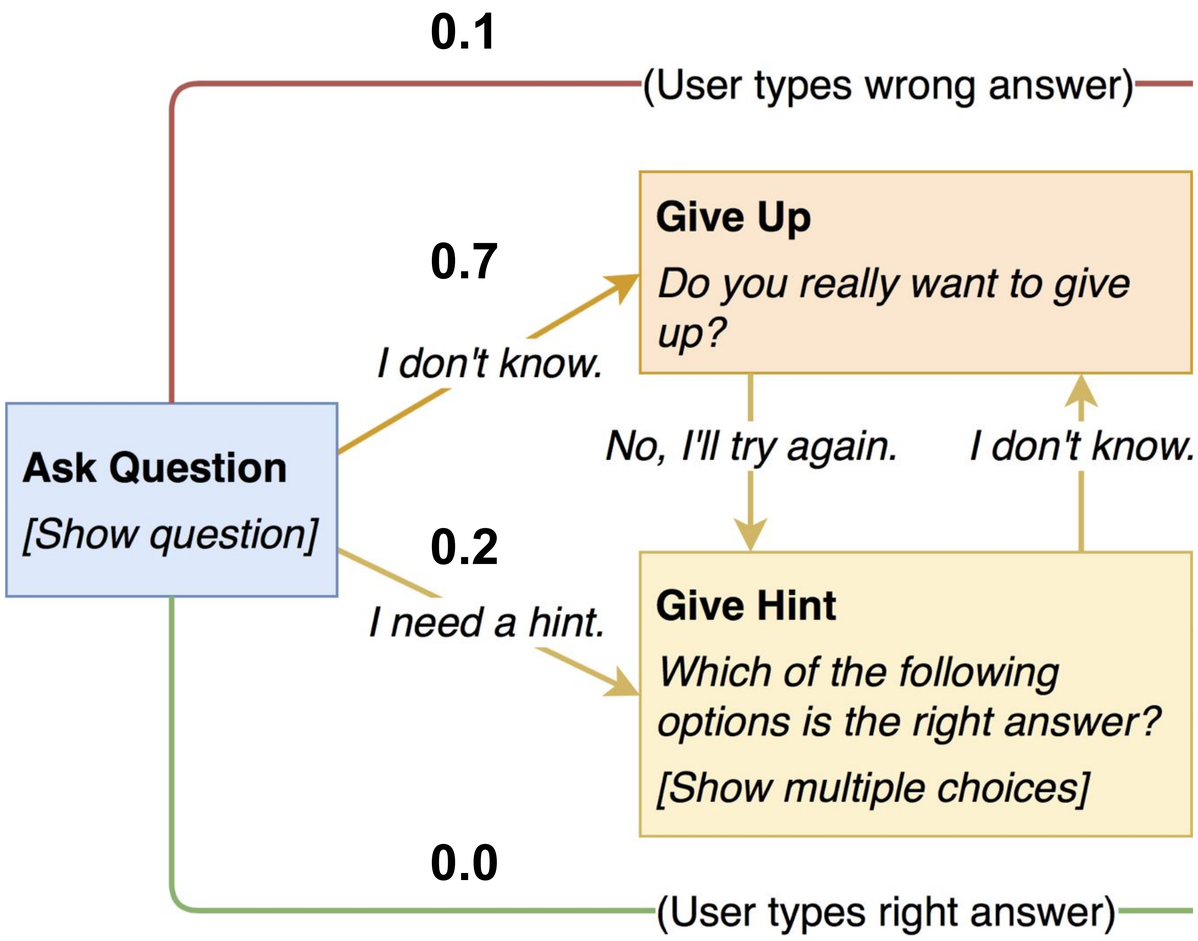
http://ai.stanford.edu/blog/quizbot/

Conversation Flow

Finite State/Markov Chain Perspective

Powerful LLM As State Transition Mechanism

- Let stateful model have specific system msg
- Let the model predict its state transitions
- Control option space in finite state premise
- Include base idle state and tooling access



$$P(\ S_t\ = X_i \mid S_{t-1},\ msg = \text{"I have no clue"}\ )$$

http://ai.stanford.edu/blog/quizbot/

Conversation Flow

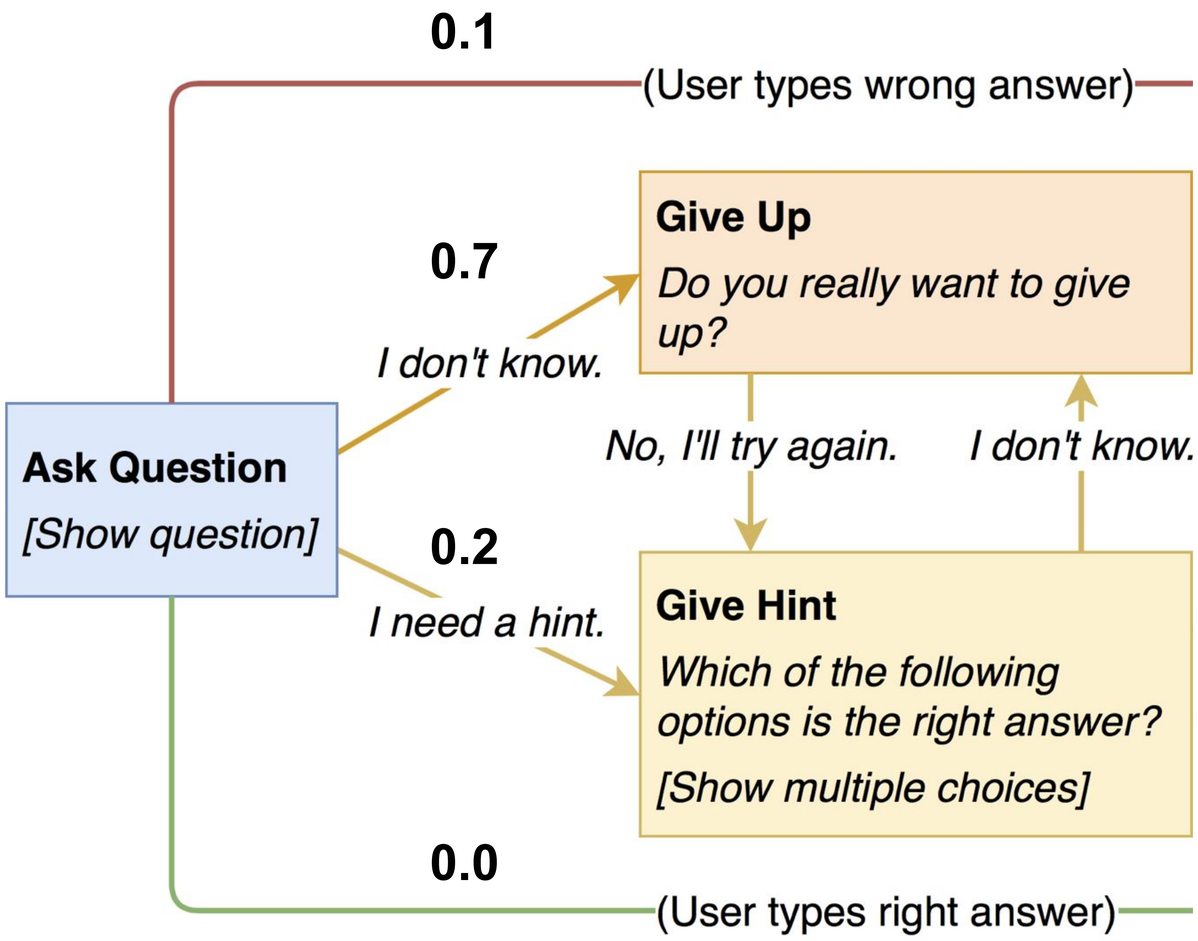
Finite State/Markov Chain Perspective

Powerful LLM As State Transition Mechanism

- Let stateful model have specific system msg
- Let the model predict its state transitions
- Control option space in finite state premise
- Include base idle state and tooling access

Simple LLM As State Transition Mechanism

- Fine-Tuned Zero-Shot Classifier
- Specialized Fine-Tuned Models
- Simple LLMs and State Checks



$$P(\ S_t\ = X_i \mid S_{t-1},\ msg = \text{"I have no clue"}\ )$$

http://ai.stanford.edu/blog/quizbot/

Implement Your Own Agent

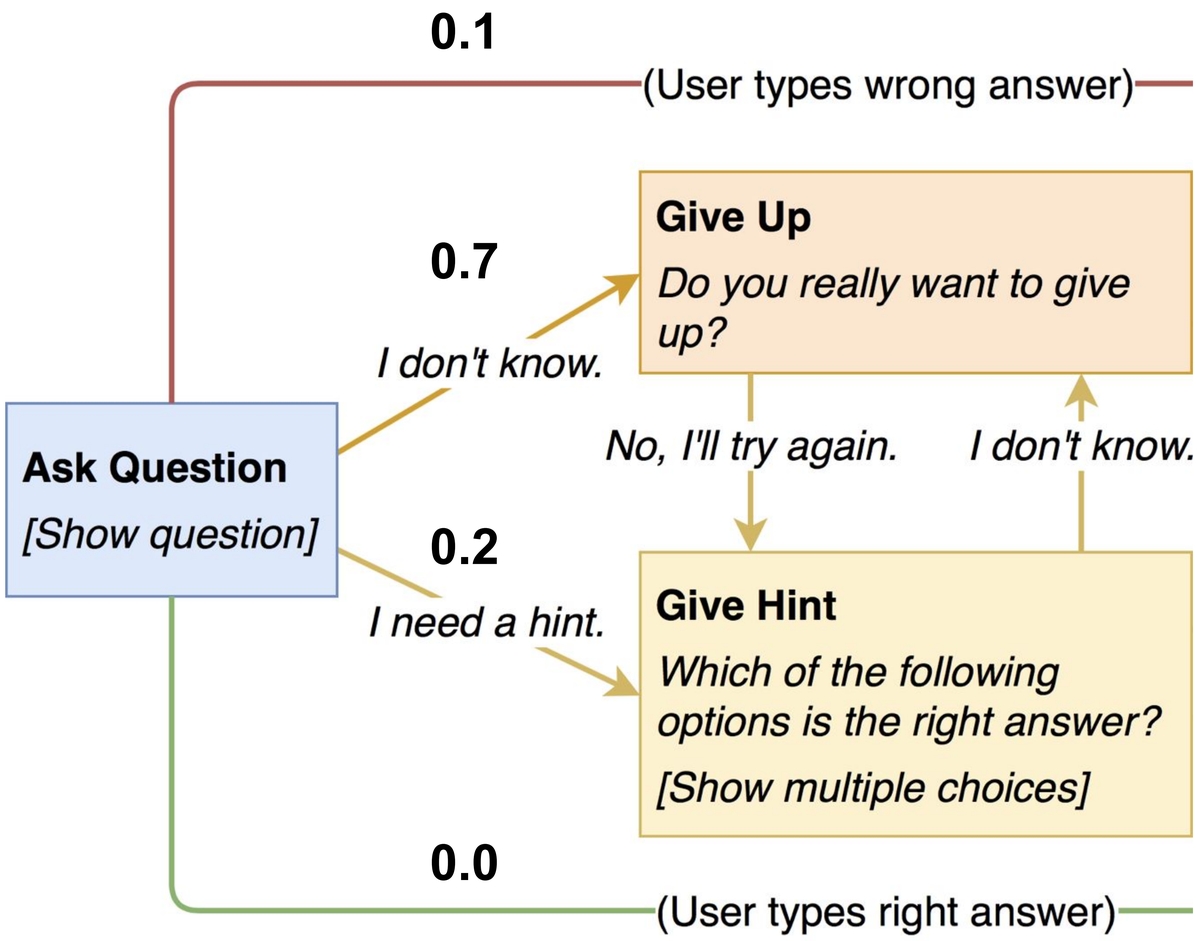
Final Assessment

What You Have

- Llama-2 Chat Agent
- Queries For User Input

What You Need

- Remember conversation info
- Keep track of user toxicity
- Keep track of user emotion
- Support python code generation
- Support image question-answering

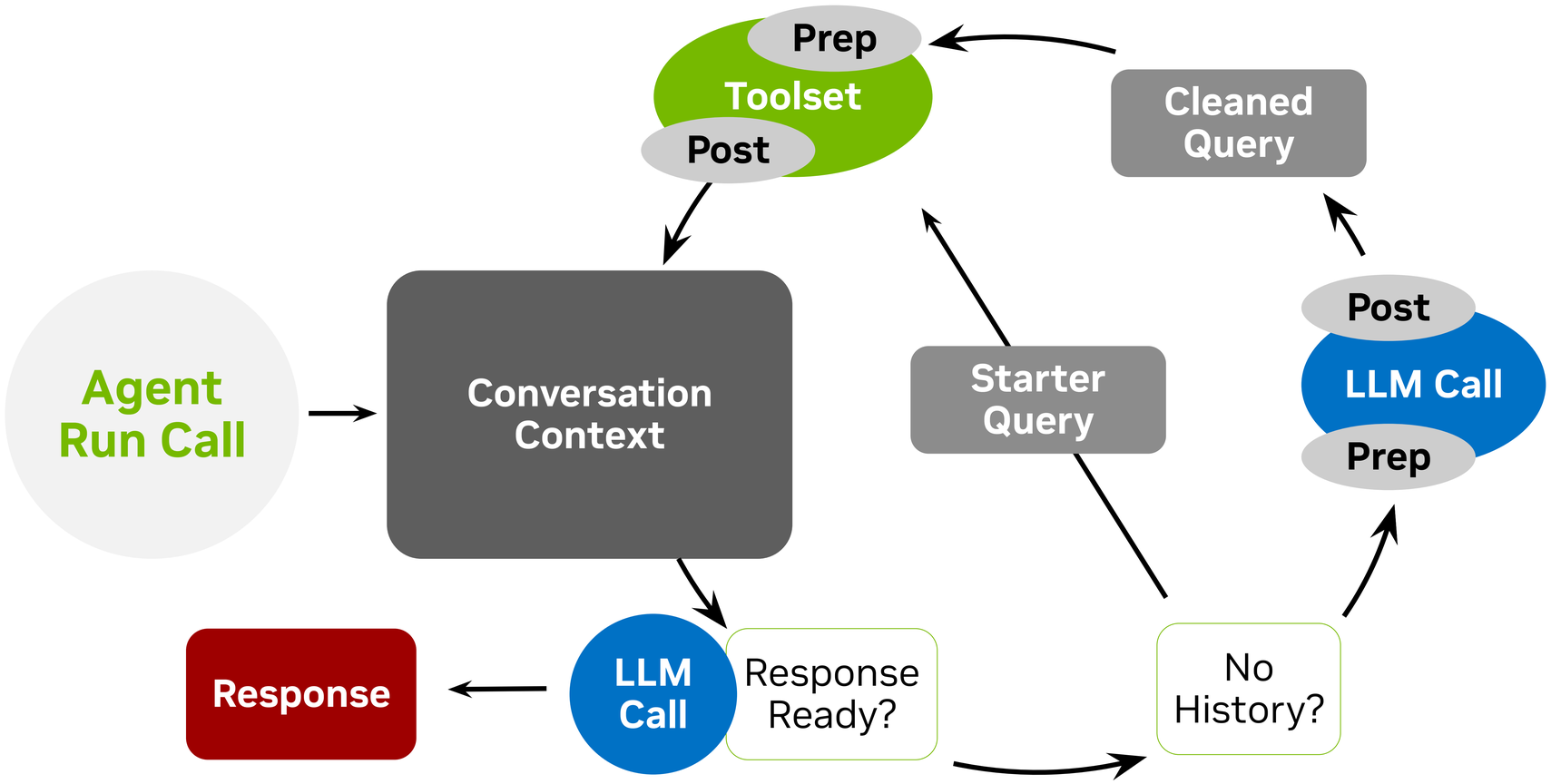


$$P(\ S_t\ = X_i \mid S_{t-1}, msg = \text{"I have no clue"}\ )$$

http://ai.stanford.edu/blog/quizbot/

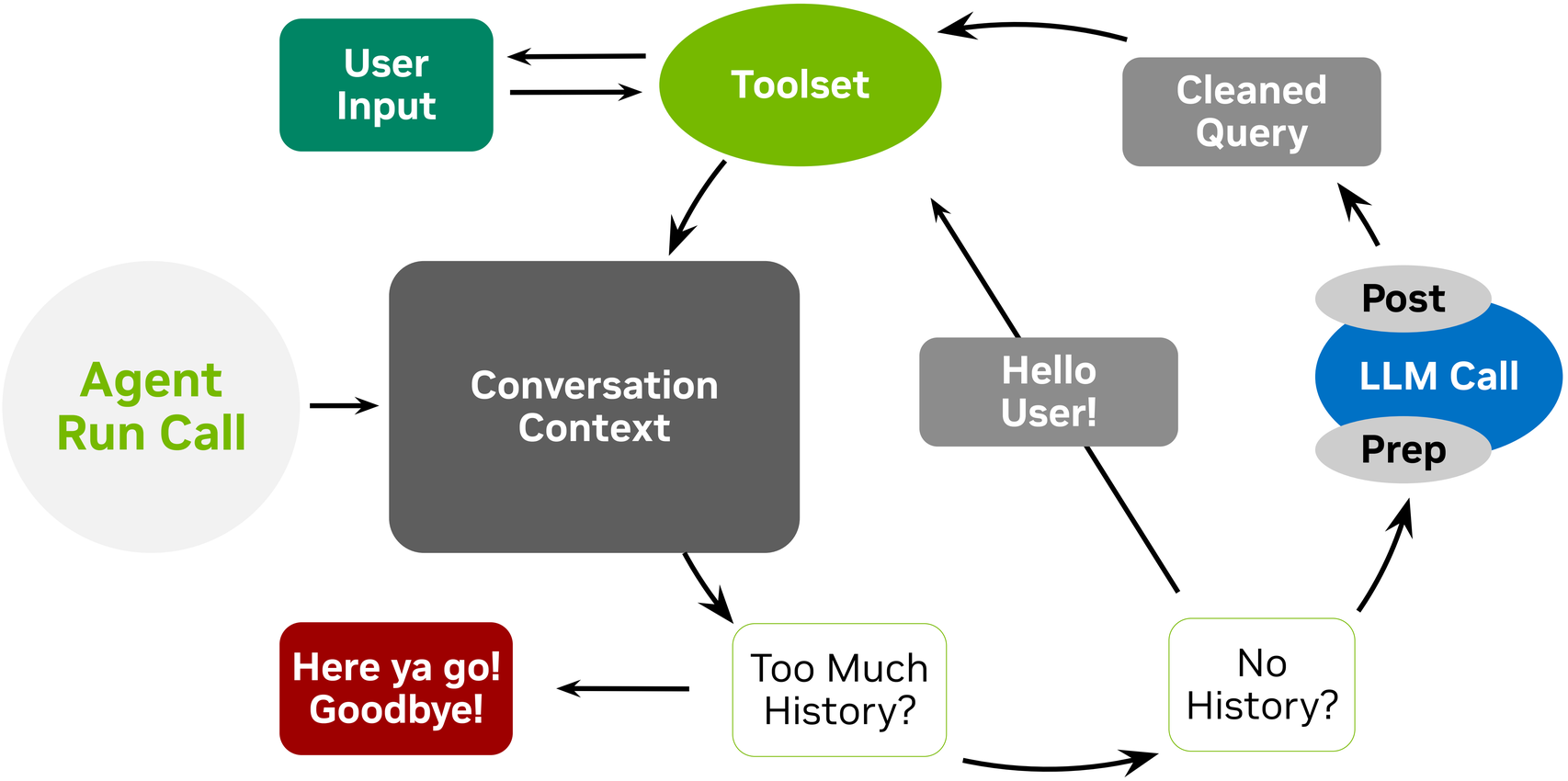
General Agent Event Loop

Final Assessment



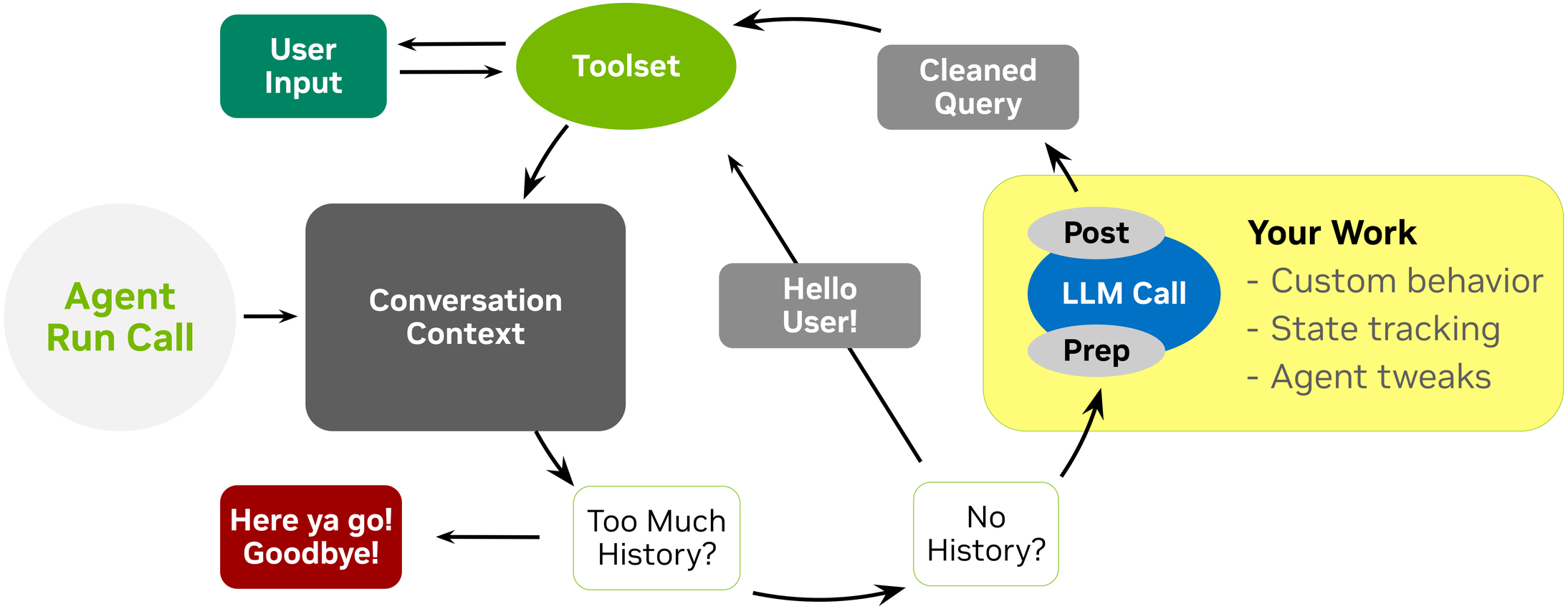
Your Agent Event Loop

Final Assessment



Your Agent Contributions

Final Assessment



Implement Your Own Agent

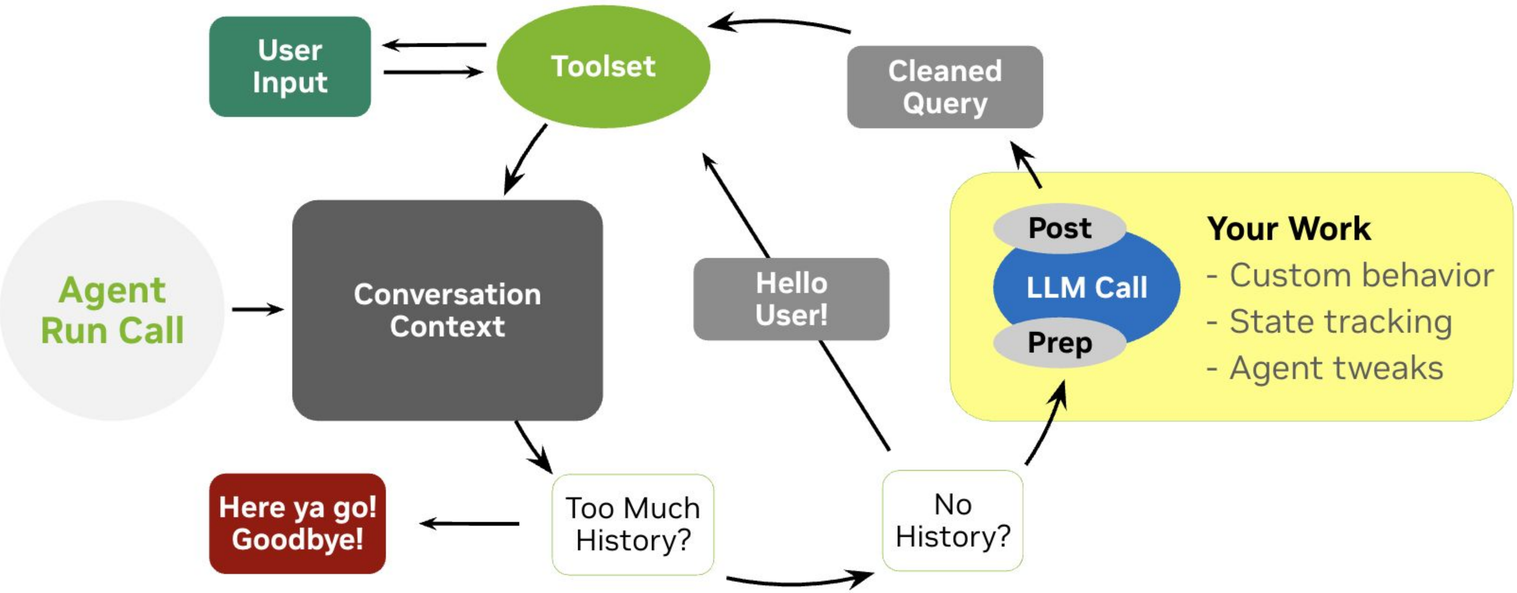
Final Assessment

What You Have

- Llama-2 Chat Agent
- Queries For User Input

What You Need

- Remember conversation info
- Keep track of user toxicity
- Keep track of user emotion
- Support python code generation
- Support image question-answering



Thank You!