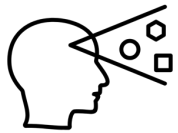
IMEN574/IDSC574 Programming for Data Science

20. **Multimodal Models II**

Spring 2026

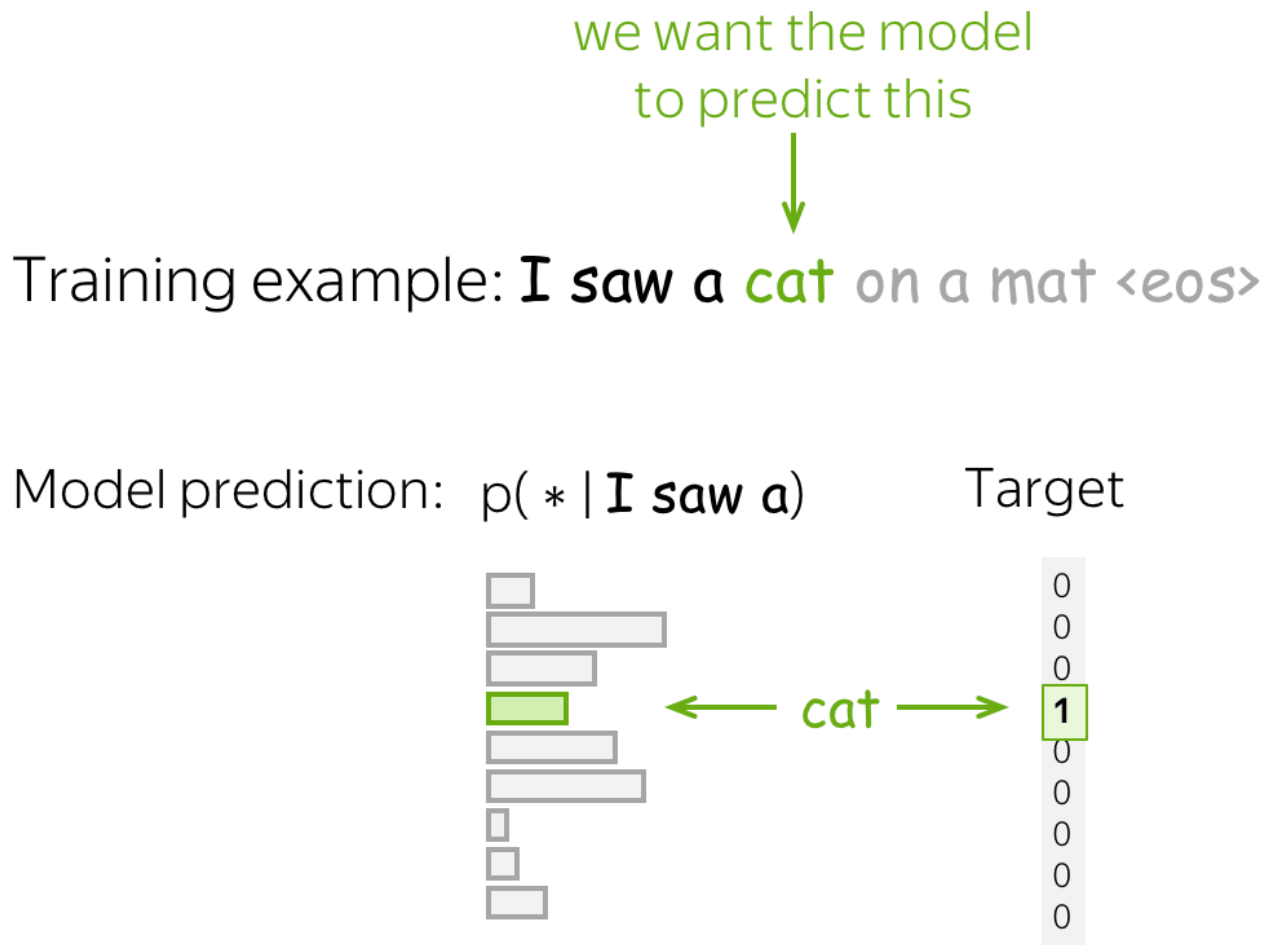
Instructor: **Sangwoo Mo**

Language Generation

	Perception 	Generation 	Action 
Vision	DINO, CLIP, SAM, etc.		
Language	BERT	LLM	

Language Generation

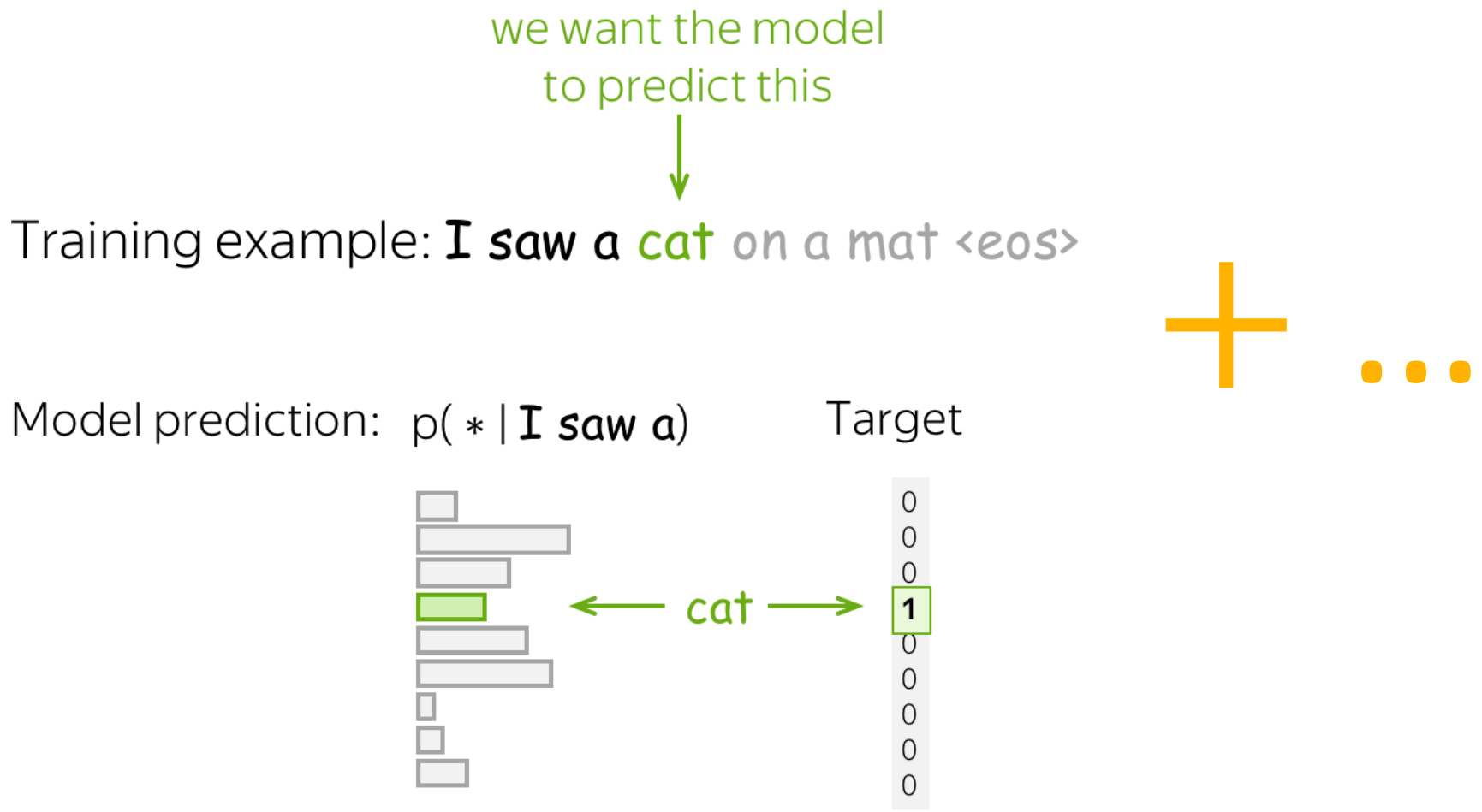
= Repetition of Next Token Prediction



Brown et al. Language Models are Few-Shot Learners. NeurIPS 2020.

Language Generation

In practice, Pre-train + SFT + RL (alignment / reasoning)



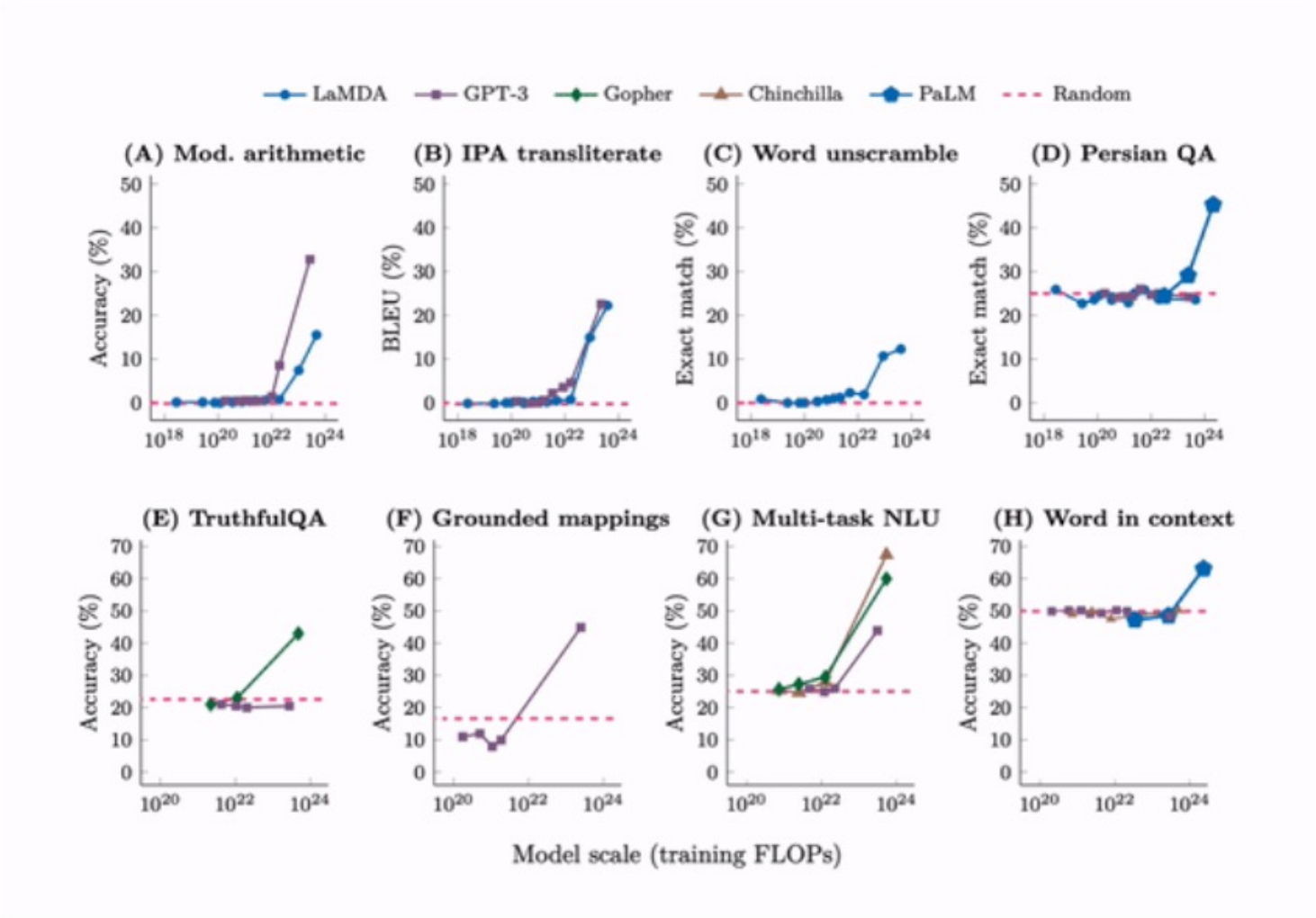
Brown et al. Language Models are Few-Shot Learners. NeurIPS 2020.

LLM

Some abilities emerge with scale

Having the right perspective is crucial

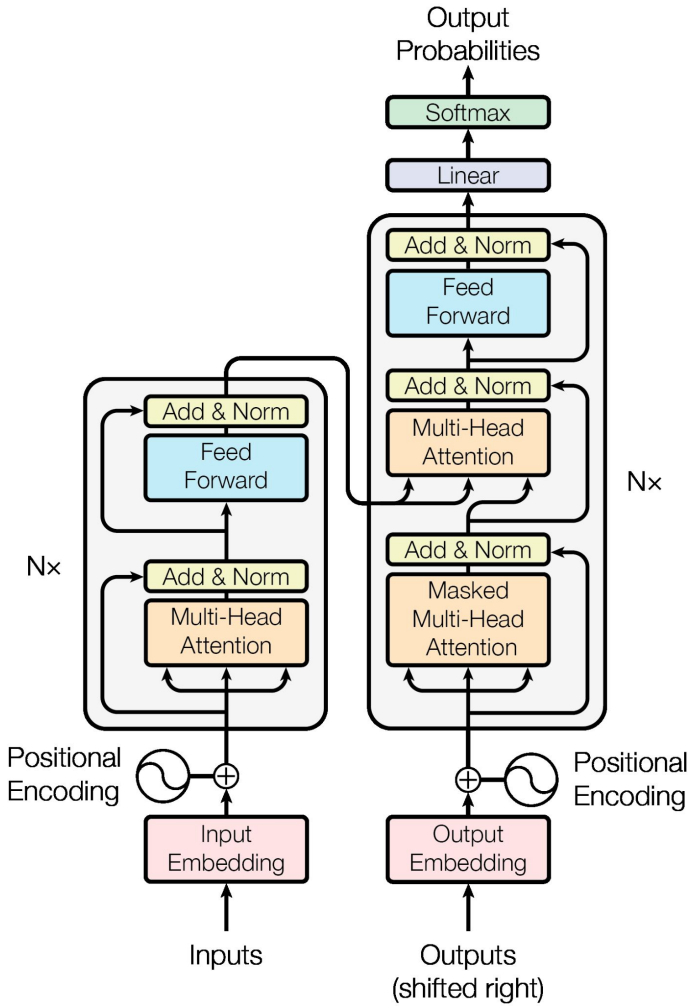
LLM



[Emergent Abilities of Large Language Models](#)
Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph et al. (2022)

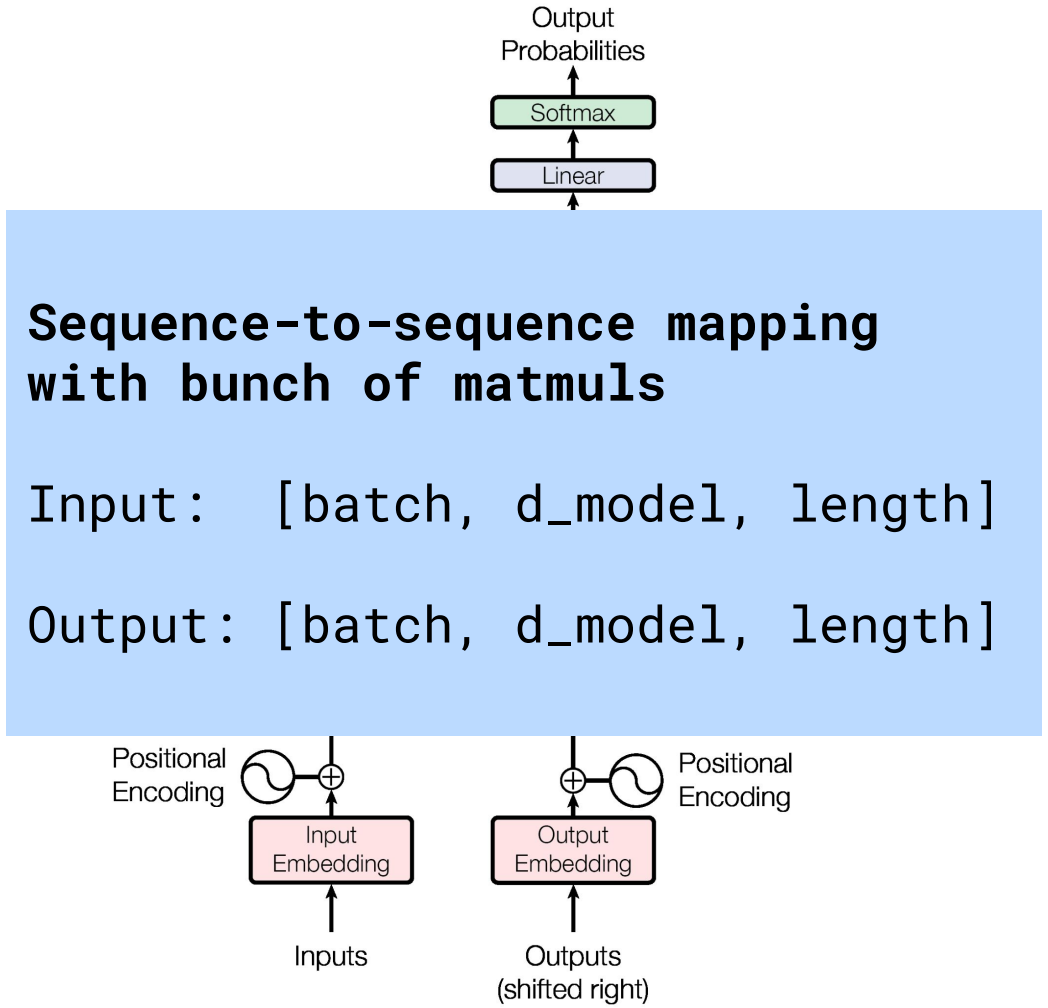
LLM

All LLMs so far use Transformer architecture



LLM

Let’s take a “functional” viewpoint on the Transformer



LLM

Process

“Many words don't map to one token: indivisible.”

Shape

[ ]

LLM

Process

“Many words don't map to one token: indivisible.”

⬇ Tokenization

Unicode characters like emojis may be split.

[7085, 2456, 836, 470, 3975, 284, 530, 11241, 25, 773, 452, 12843, 13]

Shape

[ ]

⬇

[length]

LLM

Process

“Many words don't map to one token: indivisible.”

⬇ Tokenization

Unicode characters like emojis may be split.

[7085, 2456, 836, 470, 3975, 284, 530, 11241, 25, 773, 452, 12843, 13]

⬇ Embedding

2.3	-3.2	8.3	5.4	2.1	3.9	-8.9	3.8	3.9	3.3
4.5	5.9	4.5	7.1	1.0	5.3	5.0	3.1	0.7	5.0
…	…	…	…	…	…	…	…	…	…
3.8	1.2	3.8	9.0	9.3	3.1	4.2	0.8	9.2	5.8

Shape

[ ]

⬇

[length]

⬇

[d_model, length]

LLM

Process

“Many words don't map to one token: indivisible.”

⬇ Tokenization

Unicode characters like emojis may be split.

[7085, 2456, 836, 470, 3975, 284, 530, 11241, 25, 773, 452, 12843, 13]

⬇ Embedding

2.3	-3.2	8.3	5.4	2.1	3.9	-8.9	3.8	3.9	3.3
4.5	5.9	4.5	7.1	1.0	5.3	5.0	3.1	0.7	5.0
…	…	…	…	…	…	…	…	…	…
3.8	1.2	3.8	9.0	9.3	3.1	4.2	0.8	9.2	5.8

⬇ N Transformer layers

3.2	-2.3	3.8	4.5	1.2	9.3	-9.8	8.3	9.3	3.3
5.4	9.5	5.4	1.7	0.1	3.5	0.5	1.3	7.0	0.5
…	…	…	…	…	…	…	…	…	…
8.3	2.1	8.3	0.9	3.9	1.3	2.4	8.0	2.9	8.5

Shape

[ ]

⬇

[length]

⬇

[d_model, length]

⬇

[d_model, length]

LLM

Process

Shape

“Many words don't map to one token: indivisible.”

[ ]

⬇ Tokenization

⬇

Unicode characters like emojis may be split.

[7085, 2456, 836, 470, 3975, 284, 530, 11241, 25, 773, 452, 12843, 13]

[length]

⬇ Embedding

⬇

2.3	-3.2	8.3	5.4	2.1	3.9	-8.9	3.8	3.9	3.3
4.5	5.9	4.5	7.1	1.0	5.3	5.0	3.1	0.7	5.0
…	…	…	…	…	…	…	…	…	…
3.8	1.2	3.8	9.0	9.3	3.1	4.2	0.8	9.2	5.8

[d_model, length]

⬇ N Transformer layers

⬇

3.2	-2.3	3.8	4.5	1.2	9.3	-9.8	8.3	9.3	3.3
5.4	9.5	5.4	1.7	0.1	3.5	0.5	1.3	7.0	0.5
…	…	…	…	…	…	…	…	…	…
8.3	2.1	8.3	0.9	3.9	1.3	2.4	8.0	2.9	8.5

[d_model, length]

⬇ Loss function (predict next token given previous)

⬇

2.6

[ ]

LLM

Original sentence

Many words don't map to one token: indivisible.

LLM

Original sentence

```
Many words don't map to one token: indivisible.
```

Given “many”, predict the next token

```
Many
```

apple: 0.01
don: 0.001
…
intelligence: 0.00001
…
words: 0.02

LLM

Original sentence

```
Many words don't map to one token: indivisible.
```

Given “many”, predict the next token

```
Many
```

apple: 0.01
don: 0.001
…
intelligence: 0.00001
…
words: 0.02

Given “many words”, predict the next token

```
Many words
```

apple: 0.00003
don: 0.03
…
intelligence: 0.00001
…
words: 0.0000001

LLM

Original sentence

```
Many words don't map to one token: indivisible.
```

Given “many”, predict the next token

```
Many
```

apple: 0.01
don: 0.001
…
intelligence: 0.00001
…
words: 0.02

Given “many words”, predict the next token

```
Many words
```

apple: 0.00003
don: 0.03
…
intelligence: 0.00001
…
words: 0.0000001

Probability of a sentence is a product of conditional probabilities. Maximize this.

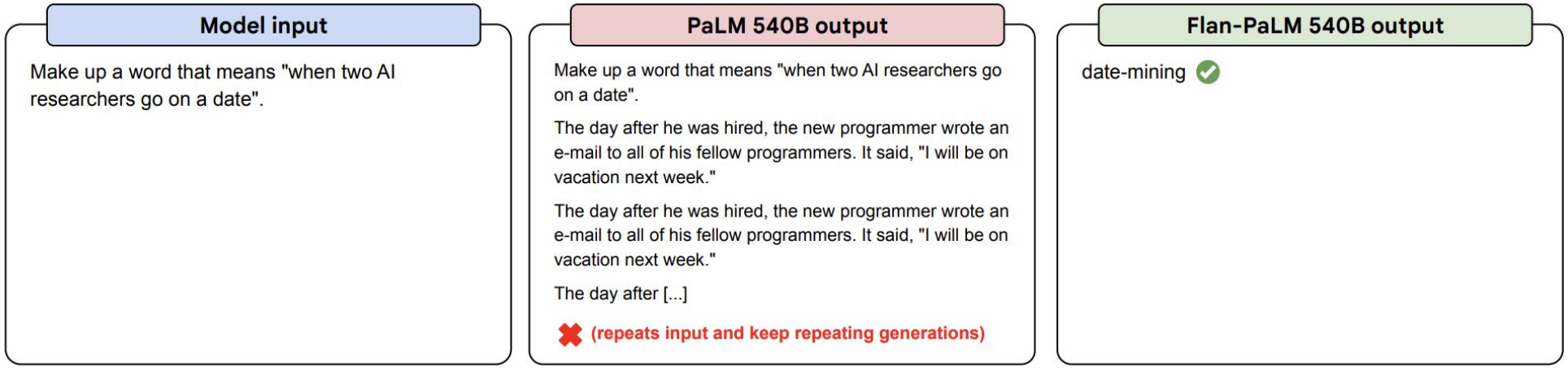
LLM

Scaling doesn’t solve all problems

We also need post-training

LLM

We can’t talk to the pretrained model directly



[Scaling Instruction-Finetuned Language Models](#)
Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus et al. (2022)

LLM

Hack: we can frame the question so that the answer is the next token

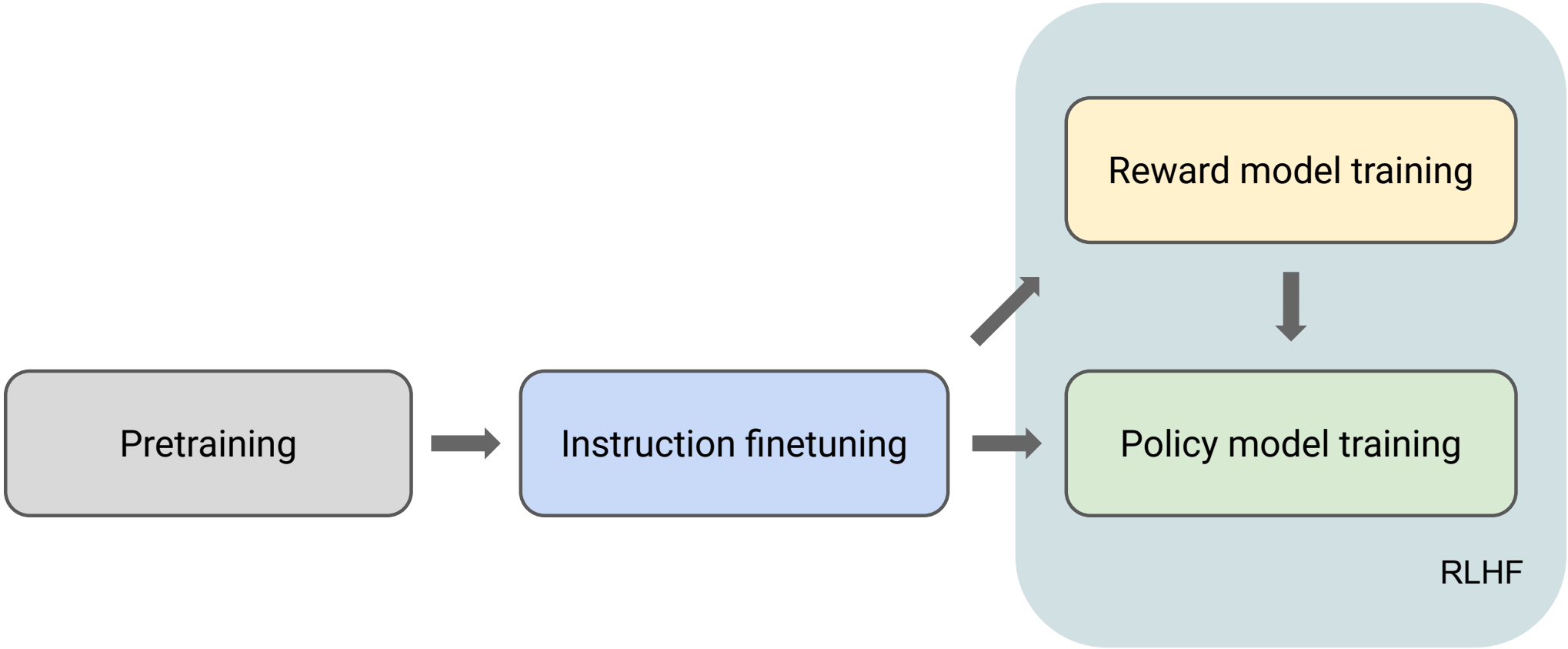
Q: The square root of x is the cube root of y. What is y to the power of 2, if x = 4?

A:



Pretrained model just predicts the next token, which happens to be the answer

LLM

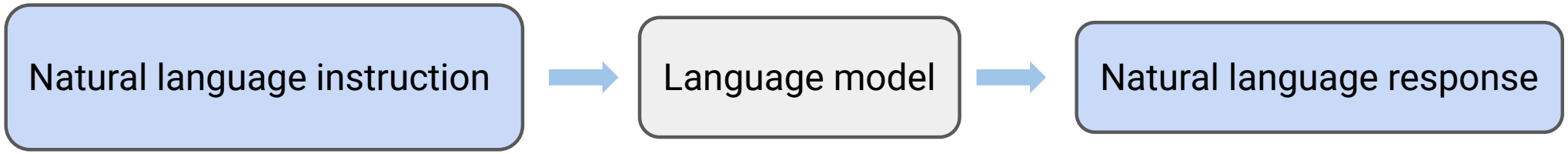

Reward model training
Pretraining
Instruction finetuning
Policy model training
RLHF

LLM

Instruction finetuning

*Frame **all** tasks in the form of*

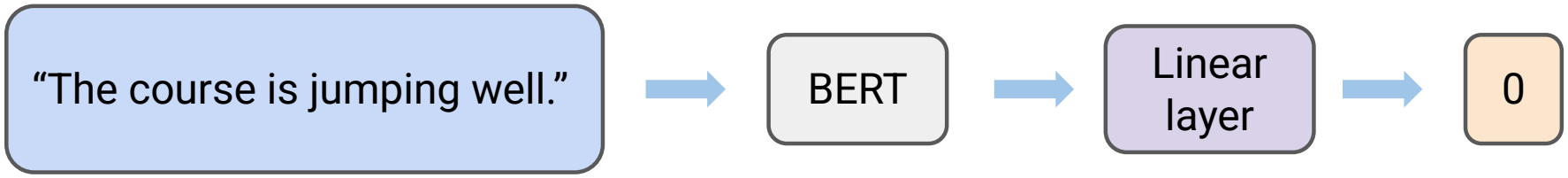
natural language instruction to natural language response mapping



LLM

Input: text

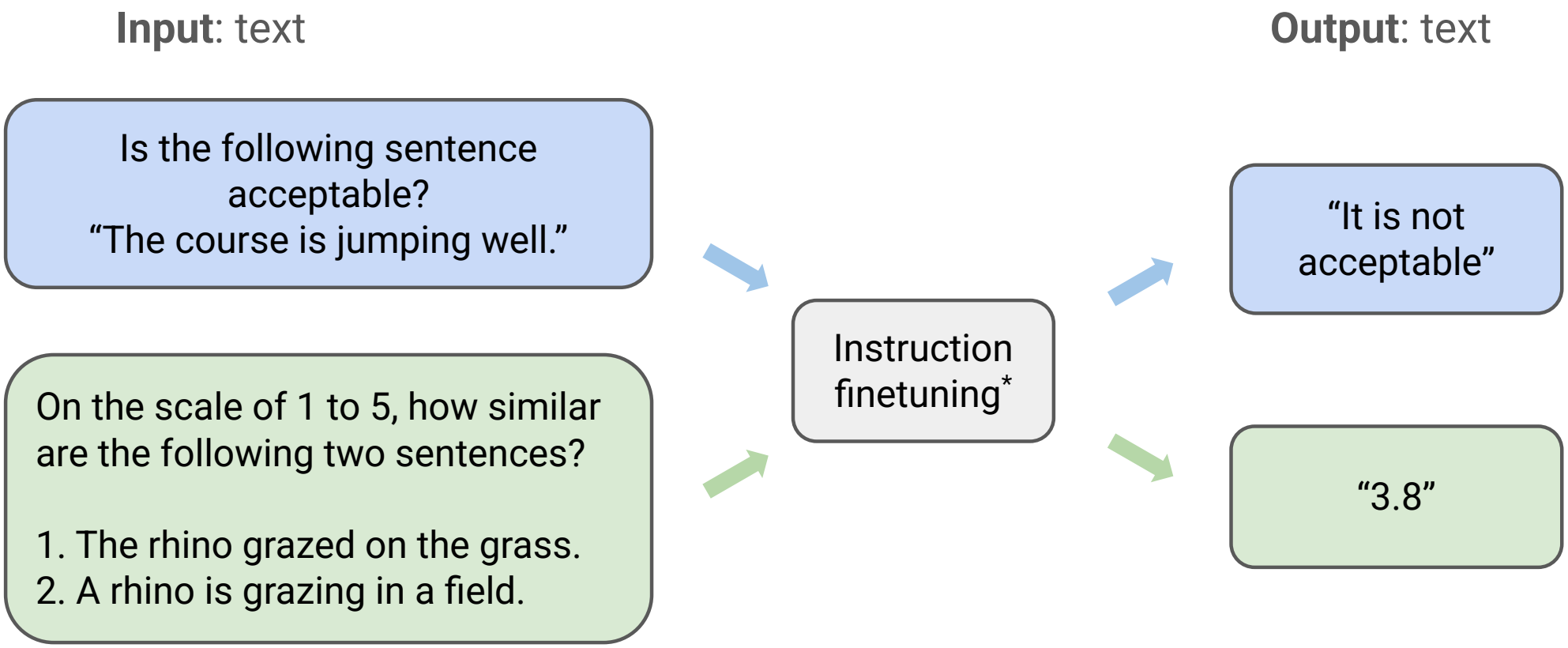
Output: label



Task specific linear layer is necessary

Devlin et al. (2018)

LLM

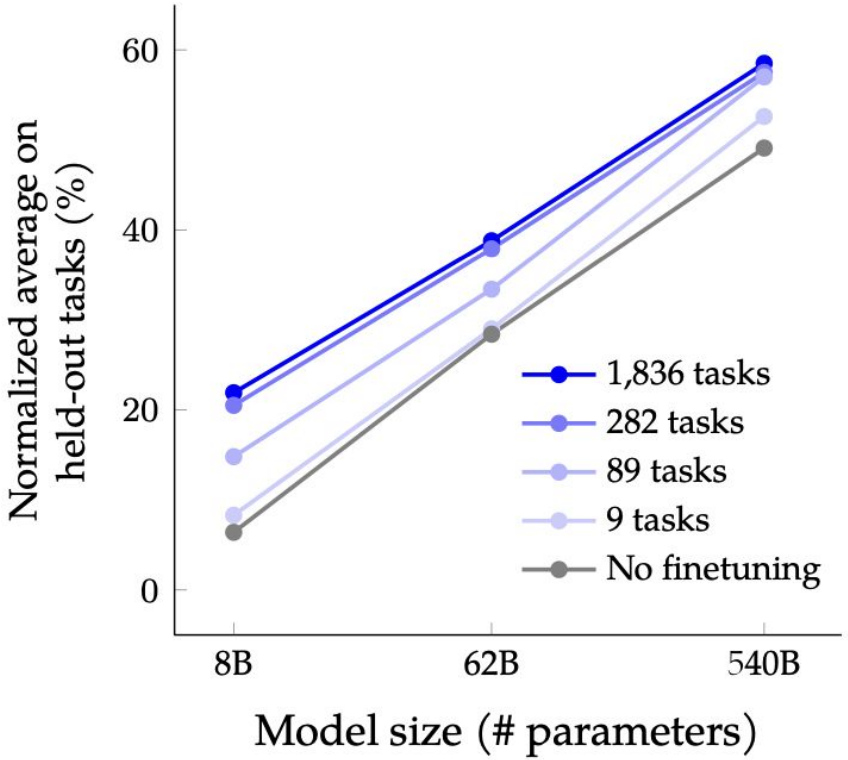


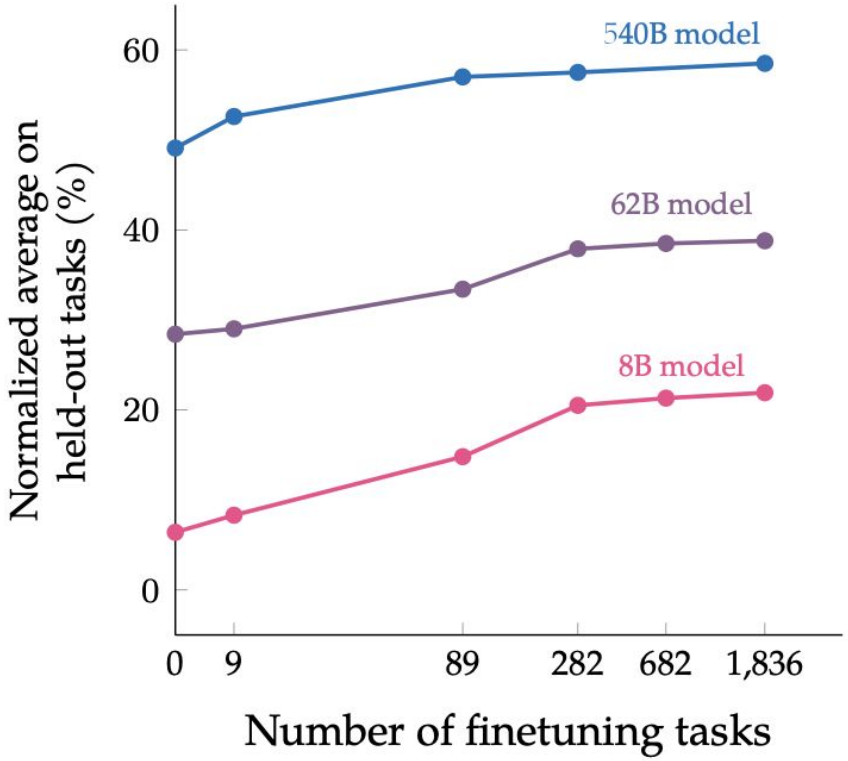
Tasks are unified. So for an unseen task, the model just needs to respond to the natural language instruction

*Wei et al. (2021), Sanh et al. (2021), Ouyang et al. (2022)

LLM

Scaling the number of tasks and model size improves the performance

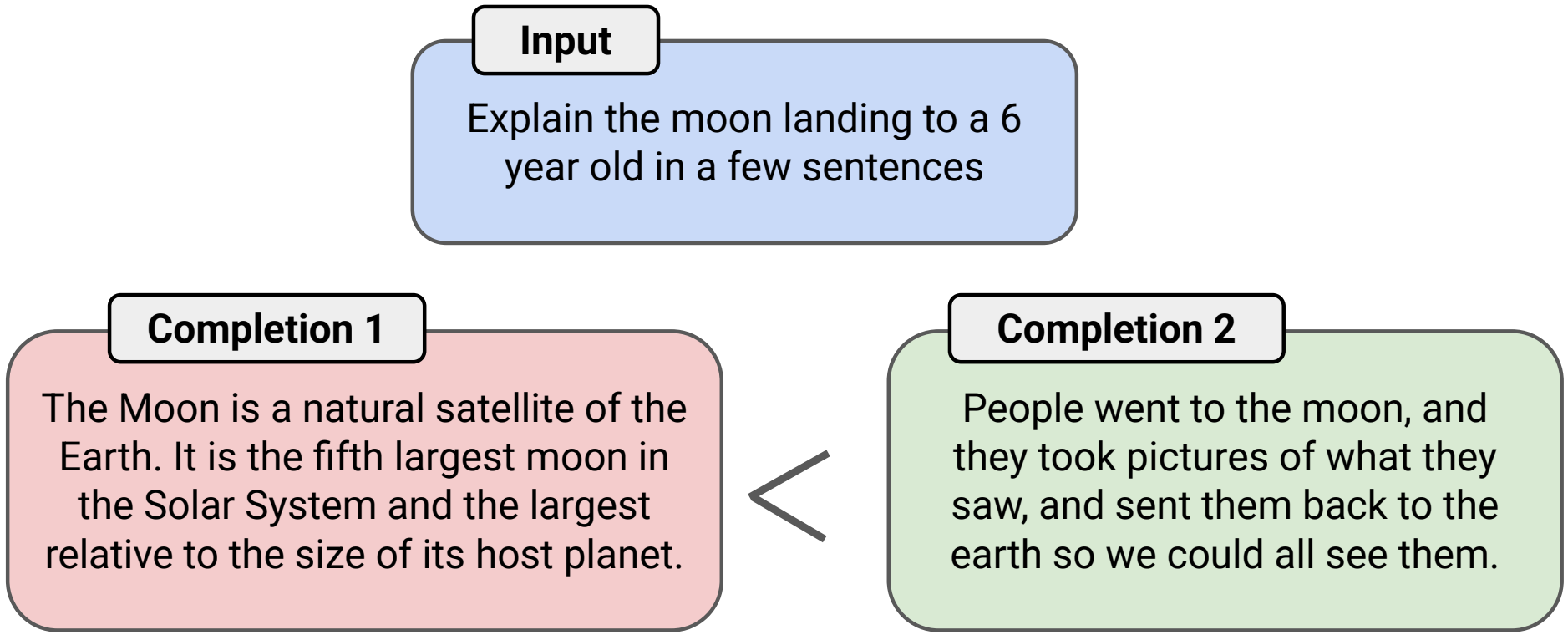




[Scaling Instruction-Finetuned Language Models]
Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus et al. (2022)

LLM

Reward Model (RM) training data: which completion is better?



Humans label which completion is preferred.

This setup aims to align models to the human preference

LLM

Reward Model (RM) training objective function

Let p_{ij} be the probability that completion y_i is better than completion y_j

Bradley–Terry model (1952): log odds that completion y_i is favored over y_j is modeled as difference in the rewards

$$\log \frac{p_{ij}}{1 - p_{ij}} = r(x, y_i; \phi) - r(x, y_j; \phi)$$

$$p_{ij} = \frac{e^{r(x,y_i;\phi)-r(x,y_j;\phi)}}{1 + e^{r(x,y_i;\phi)-r(x,y_j;\phi)}} = \sigma(r(x, y_i; \phi) - r(x, y_j; \phi))$$

$$\max_{\phi} \sum_{x,y_i,y_j \in D} \log p_{ij}$$

LLM

Policy model objective function

Once we have a reward model, we can use it in RL to learn the language model parameters that maximizes the expected reward

$$J(\theta) = \mathbb{E}_{(X,Y)\sim D_{\pi_\theta}}[r(X, Y; \phi)]$$

where $X = (X_1, \ldots, X_S)$ is the prompt and $Y = (Y_1, \ldots, Y_T)$ is the completion sampled from the policy model.

LLM

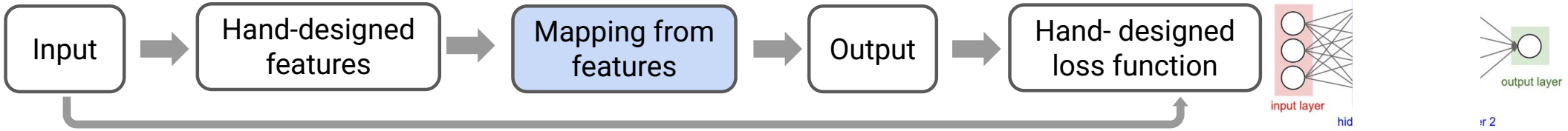
Learnable part of the system

Rule-based systems



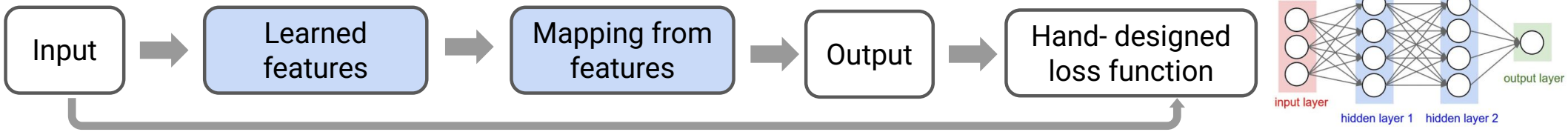
Classical machine learning

logistic regression

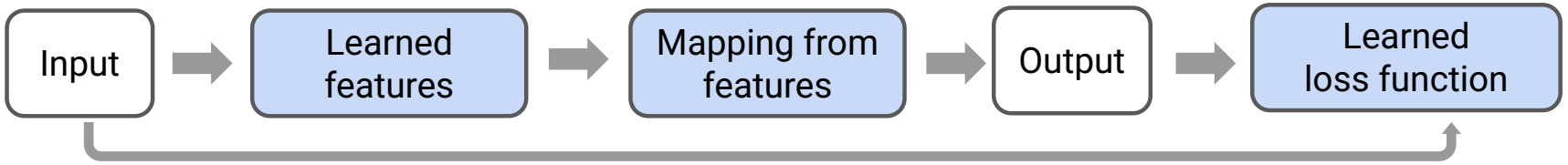


Deep learning: (self-)supervised learning

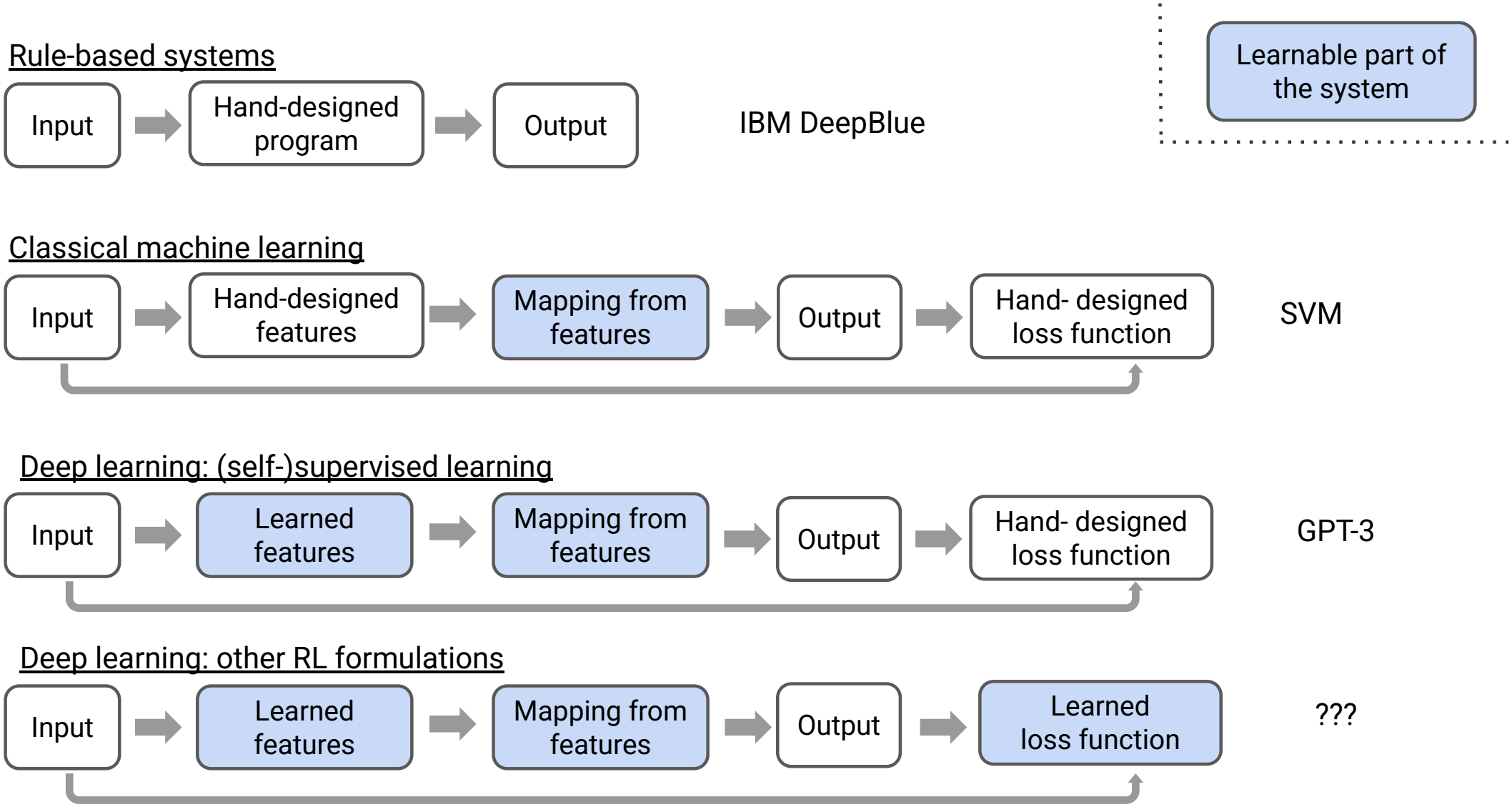
Feedforward neural net



Deep learning: other RL formulations

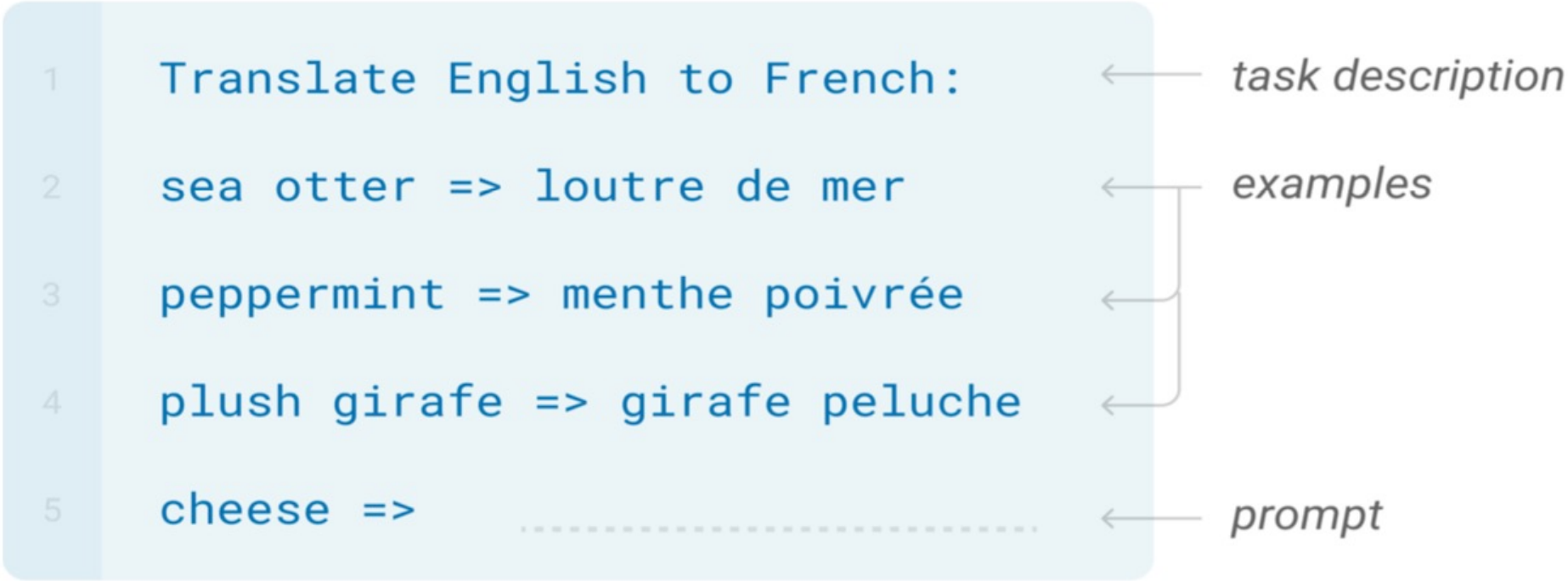


LLM


Learnable part of the system
Rule-based systems
Input
Hand-designed program
Output
IBM DeepBlue
Classical machine learning
Input
Hand-designed features
Mapping from features
Output
Hand- designed loss function
SVM
Deep learning: (self-)supervised learning
Input
Learned features
Mapping from features
Output
Hand- designed loss function
GPT-3
Deep learning: other RL formulations
Input
Learned features
Mapping from features
Output
Learned loss function
???

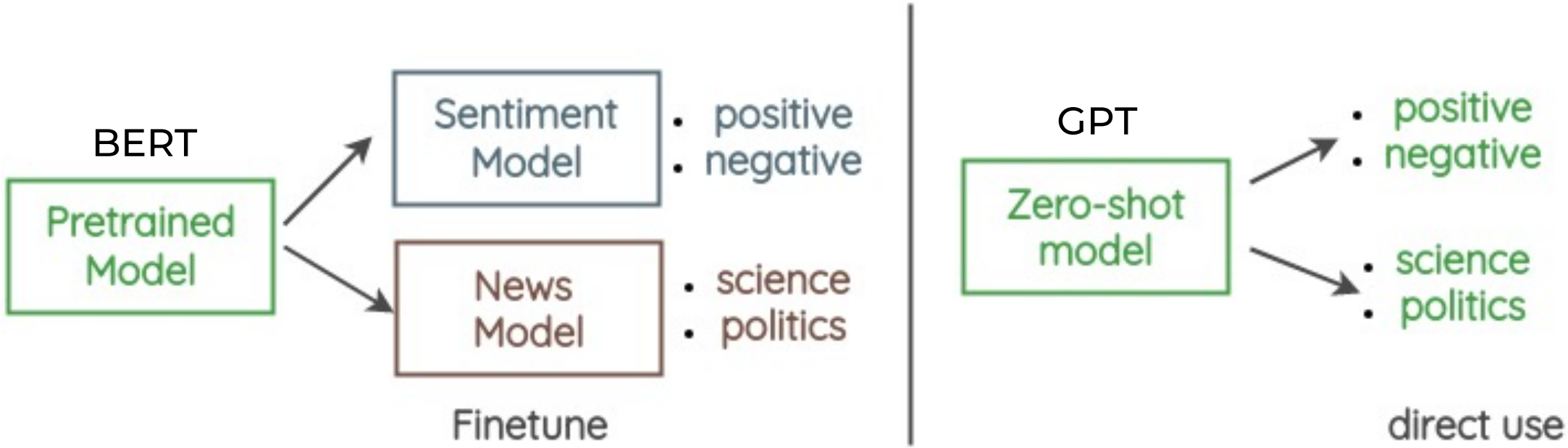
In-Context Learning (ICL): New Paradigm of FM

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



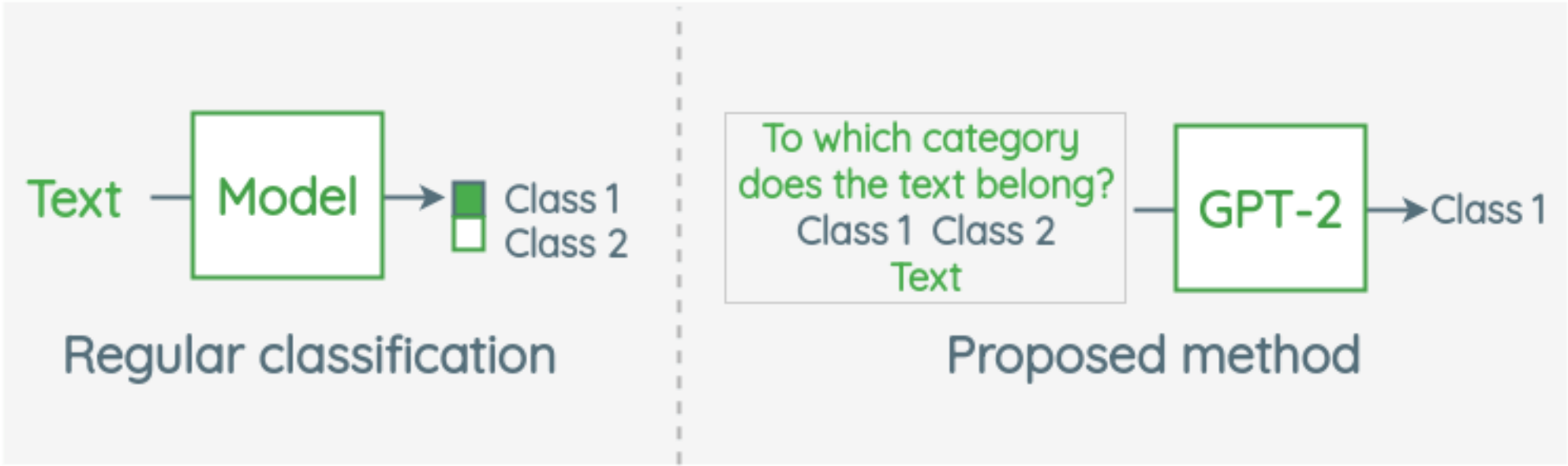
Brown et al. Language Models are Few-Shot Learners. NeurIPS 2020.

In-Context Learning (ICL): New Paradigm of FM

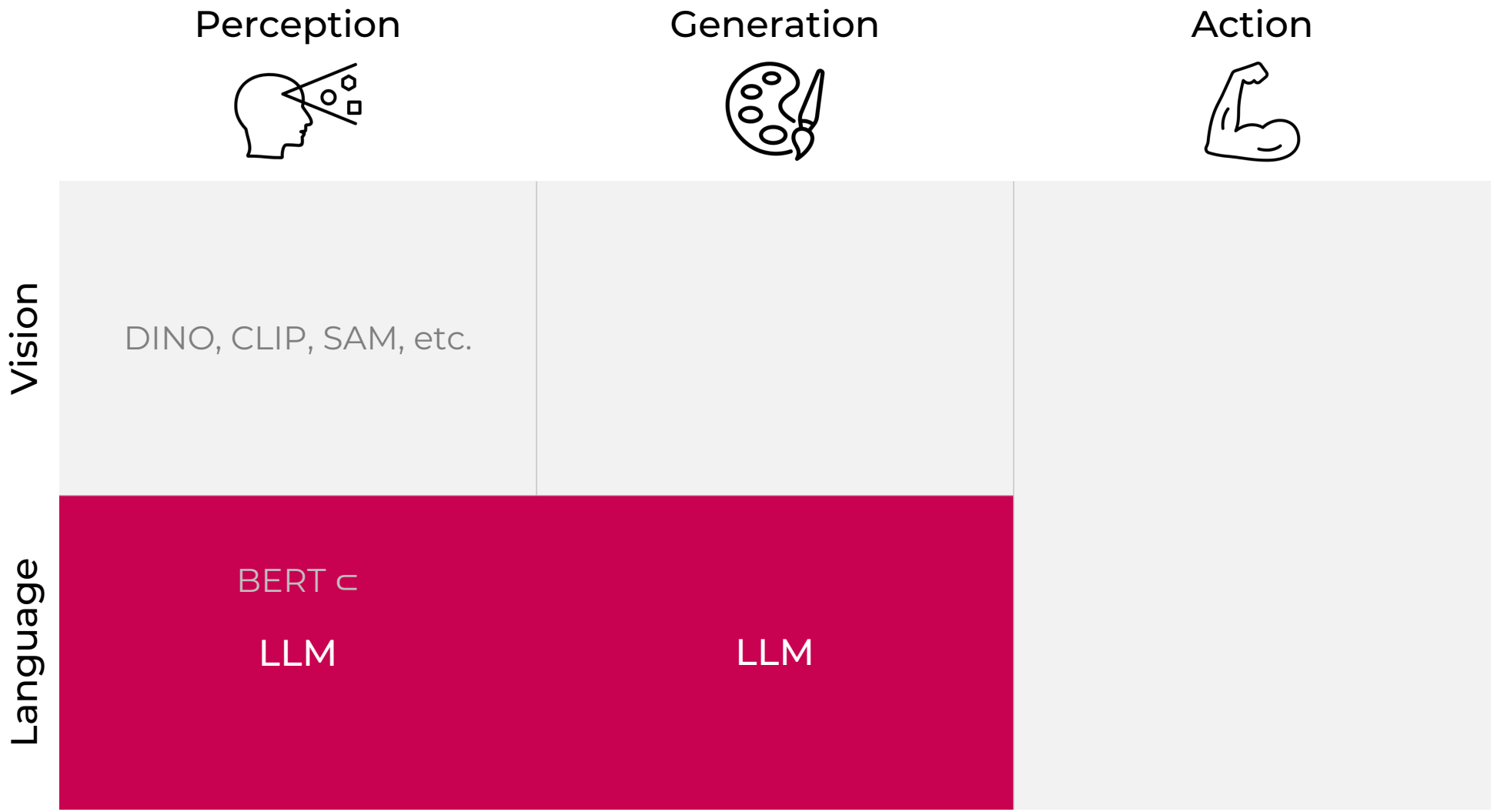


Brown et al. Language Models are Few-Shot Learners. NeurIPS 2020.

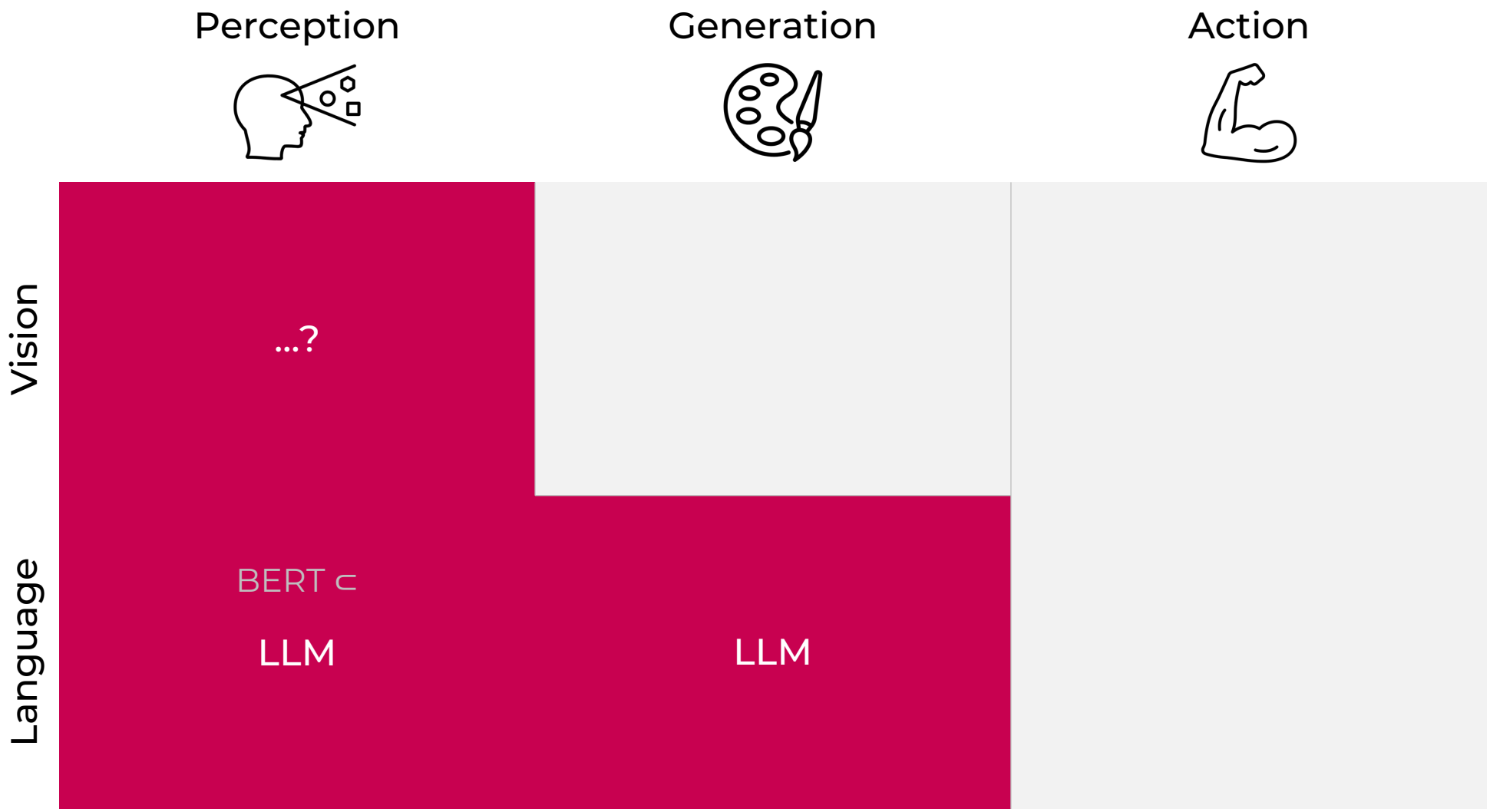
Understanding ⊂ Generation



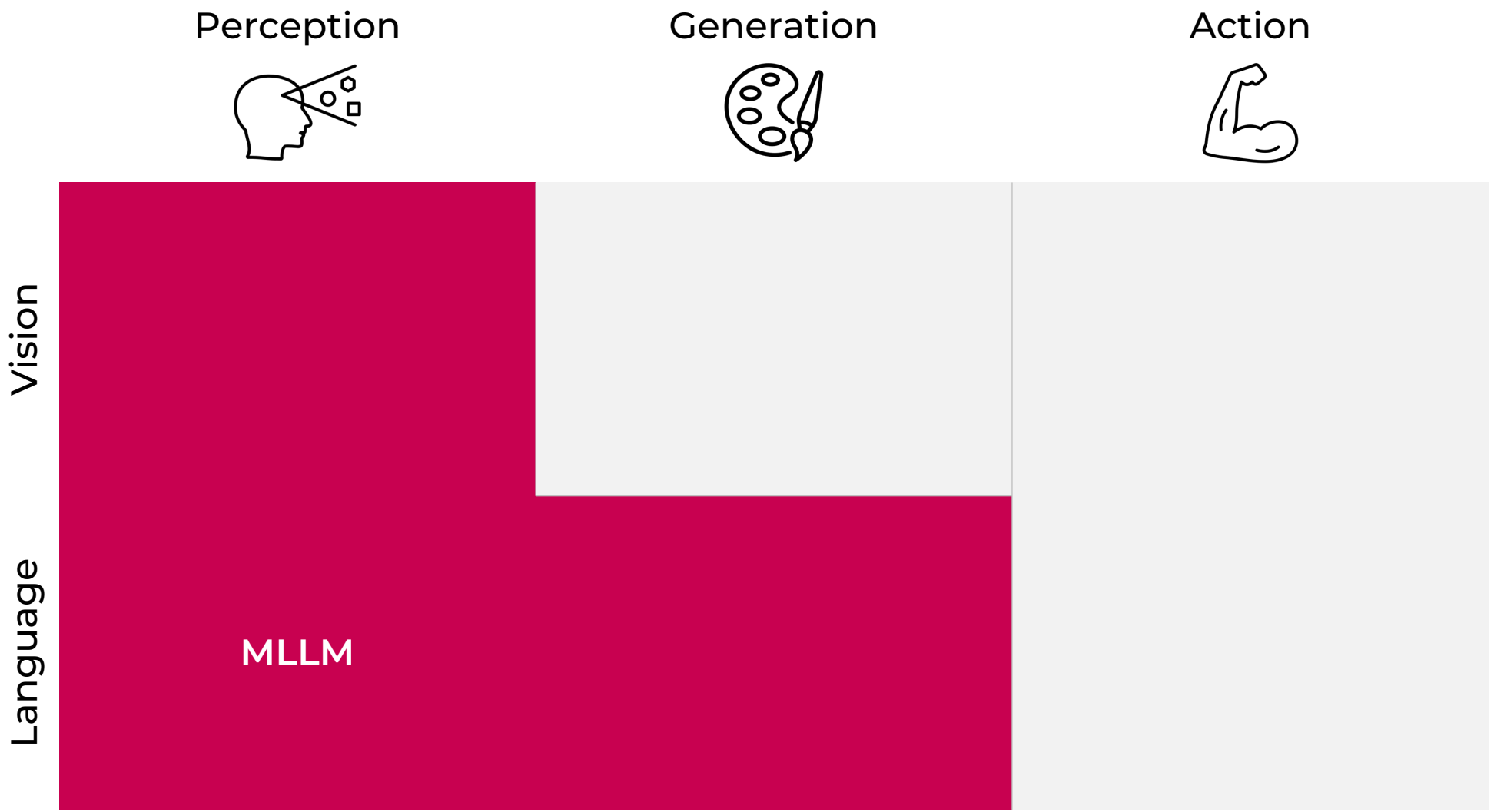
Understanding ⊂ Generation



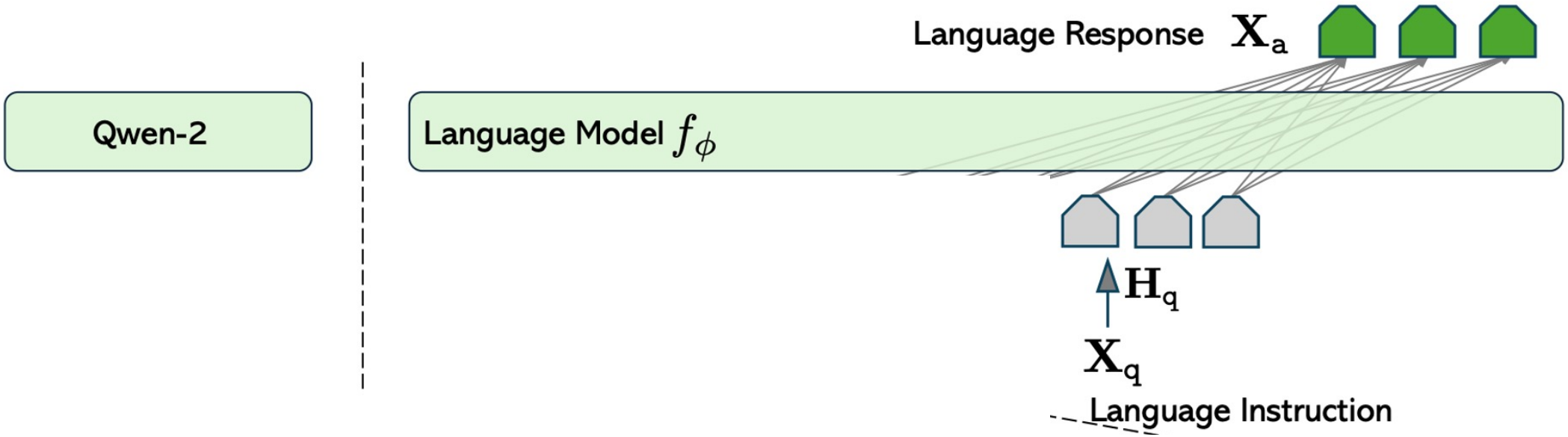
LLM + Visual Perception?



Multimodal LLM

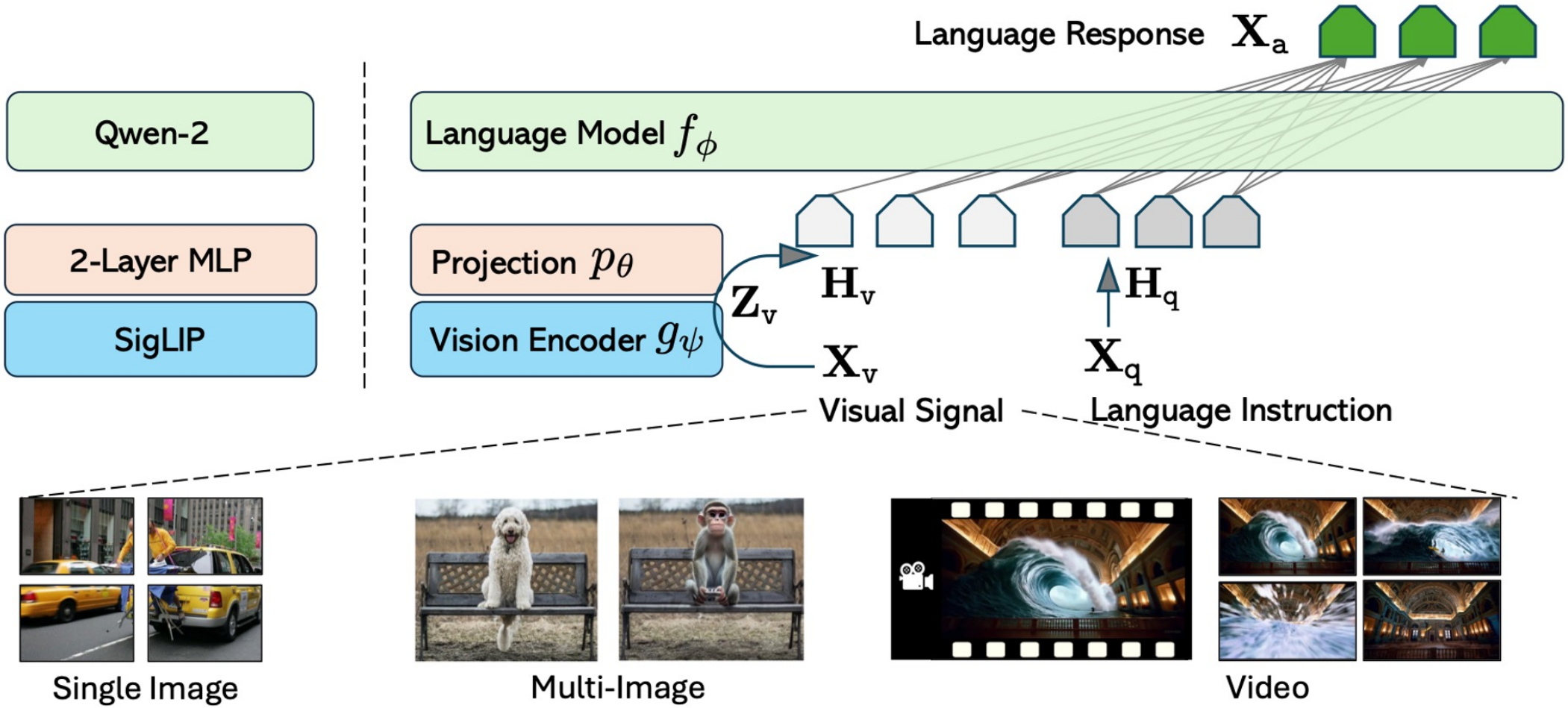


Multimodal LLM



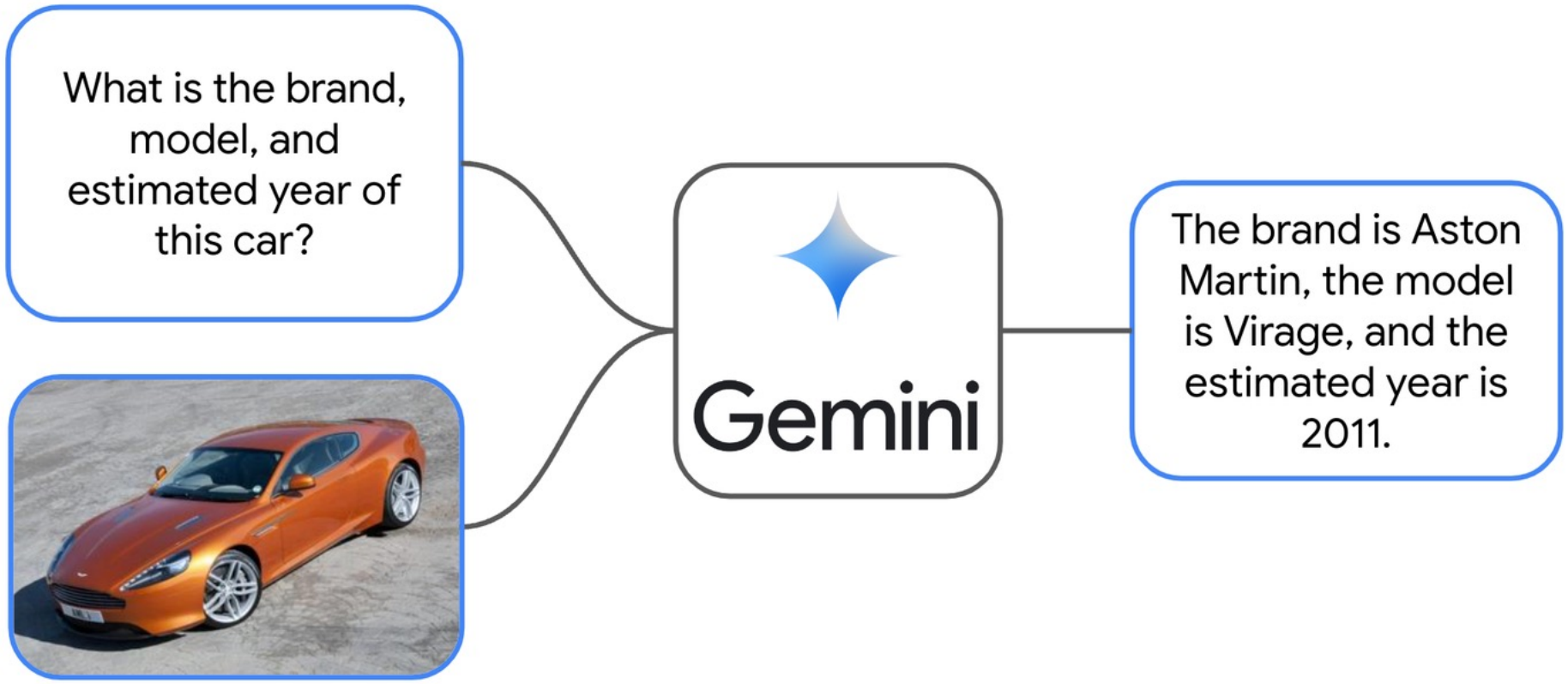
Liu et al. **Visual Instruction Tuning**. NeurIPS 2023.

Multimodal LLM



Liu et al. **Visual Instruction Tuning**. NeurIPS 2023.

Multimodal LLM: Visual Reasoning



Liu et al. **Visual Instruction Tuning**. NeurIPS 2023.

Progress So Far – Part IV

