CSCE 638 Natural Language Processing Foundation and Techniques

Lecture 11: Model Efficiency

Kuan-Hao Huang

Spring 2026

Quiz 2

- Feb 25 (Wednesday)
- Coverage: mainly Lecture 6 to 10
  - Naturally include some concepts in Lecture 1 to 5
- In-class, 20 minutes, closed book, no cheat sheet
- Written quiz, 5 questions
  - Please bring a pen
- Tips
  - Get familiar with formula
  - Understand the intuition behind the formula and the design
  - Know the pros and cons of different approaches

Course Project

- Each team should have 3–4 members of your choice
- Two possible tracks of projects
  - Research Track
  - Application Track
    - Application track will present first at the end of semester

Project Proposal

- Due: Mar 6
- Page limit: 2 pages (excluding reference)
- Format: ACL style

Project Proposal – Research Track

- Example topics
  - Selecting an existing problem discussed in class and developing new ideas
  - Identify unresolved challenges from a paper and improve the approach
  - Participate in ongoing shared tasks and present the techniques you apply
- Proposal
  - **Introduction:** project scope, challenges, novelty, expected contribution
  - **Related work:** related literature, current research progress, what's missing
  - **Methodology:** detailed problem definition, proposed approach
  - **Experiments:** planned experiments, datasets, baselines, expected results
  - **Expected timeline:** timeline
- Expectation: a conference workshop level submission

Project Proposal – Application Track

- Example topics
  - An NLP system with UI and multiple features to solve a real application
  - A Chrome extension that applies NLP techniques to assist users in real time
  - Develop an App with compelling features that requires NLP techniques
- Proposal
  - **Introduction:** project scope, importance, challenges, expected contribution
  - **Related work:** related existing applications, what's missing
  - **Designs:** detailed application design, required NLP techniques and models
  - **Outcomes:** planned outcomes and features, planned demo, evaluation metrics
  - **Expected timeline:** timeline
- Expectation: a produce prototype

Project Proposal – Example

~~Train NLP models to predict the sentiment of reddit comments~~

- Identify the uniqueness of reddit comments: lots of acronym, jargons, and newly created words
- Challenge: existing models are not good at those “not common” words → show simple evidence
- Propose methods to make NLP models better on those cases
- Show experimental evidence

Research Track

- Not only train NLP models but also build a Chrome extension to display the sentiment in real time
- Visualize the level of sentiment and indicate important keywords
- Support checking users’ historical comment sentiment
- Provide a slider to allow checking change over time

Application Track

Team Sign-Up

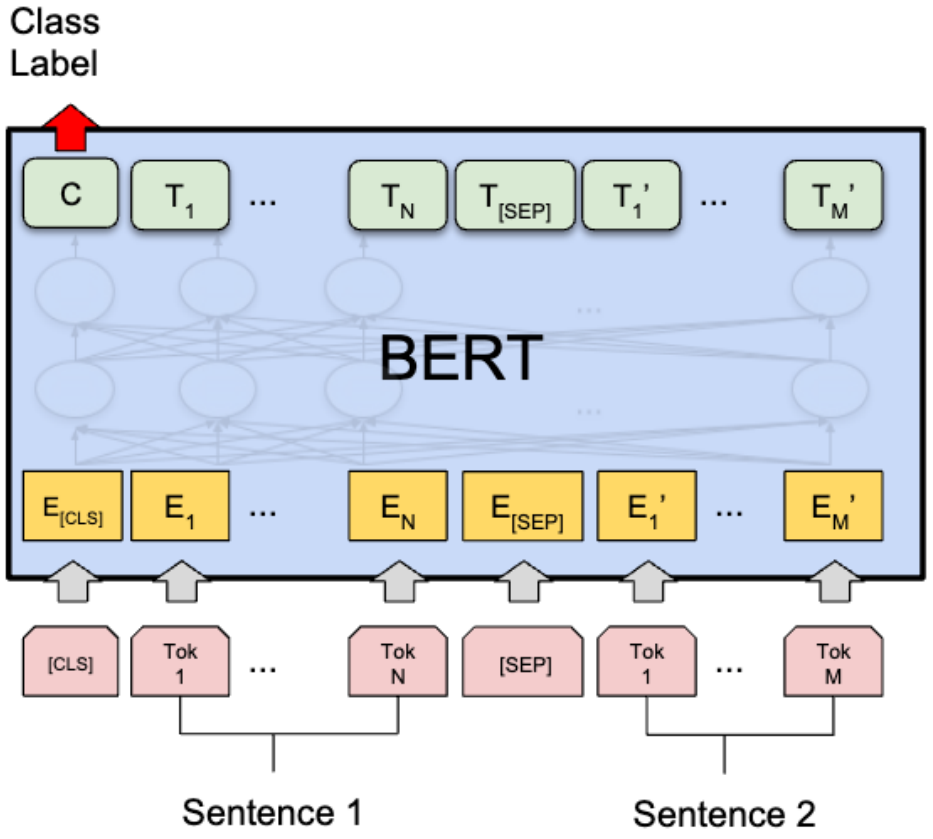
- https://docs.google.com/spreadsheets/d/1qUZPFI4wciToJsXye8-WN4L7xVG38IWdS2GCCzmu84A/edit?usp=sharing

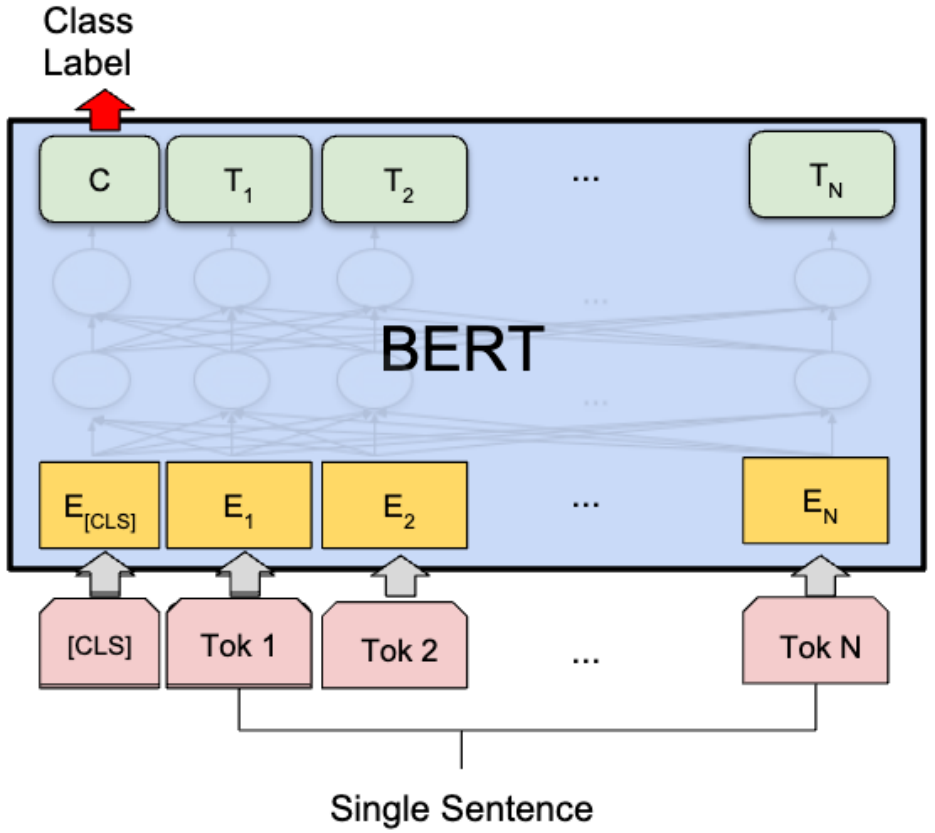
Lecture Plan

- Parameter-Efficient Fine-Tuning
  - Prompt Tuning, Prefix Tuning, Adapter
  - Low-Rank Adaptation (LoRA)
- Efficient Architecture
  - Mixture of Experts (MoE)
- Model Compression
  - Pruning, Quantization
  - Distillation
- Inference
  - KV Cache

Pre-Trained Models Provide Good Initialization

- Pre-training provides a weight initialization for continuing fine-tuning

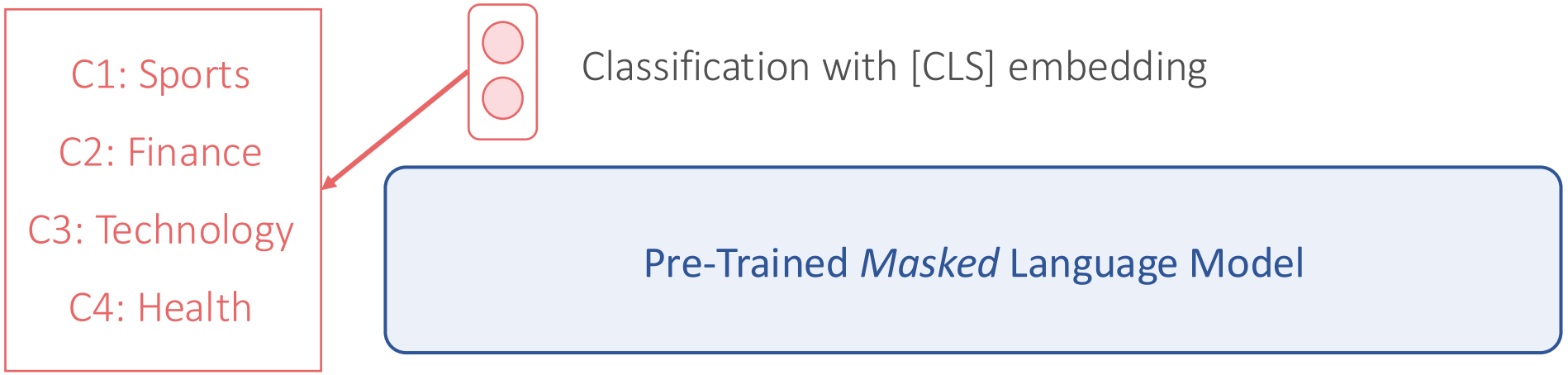




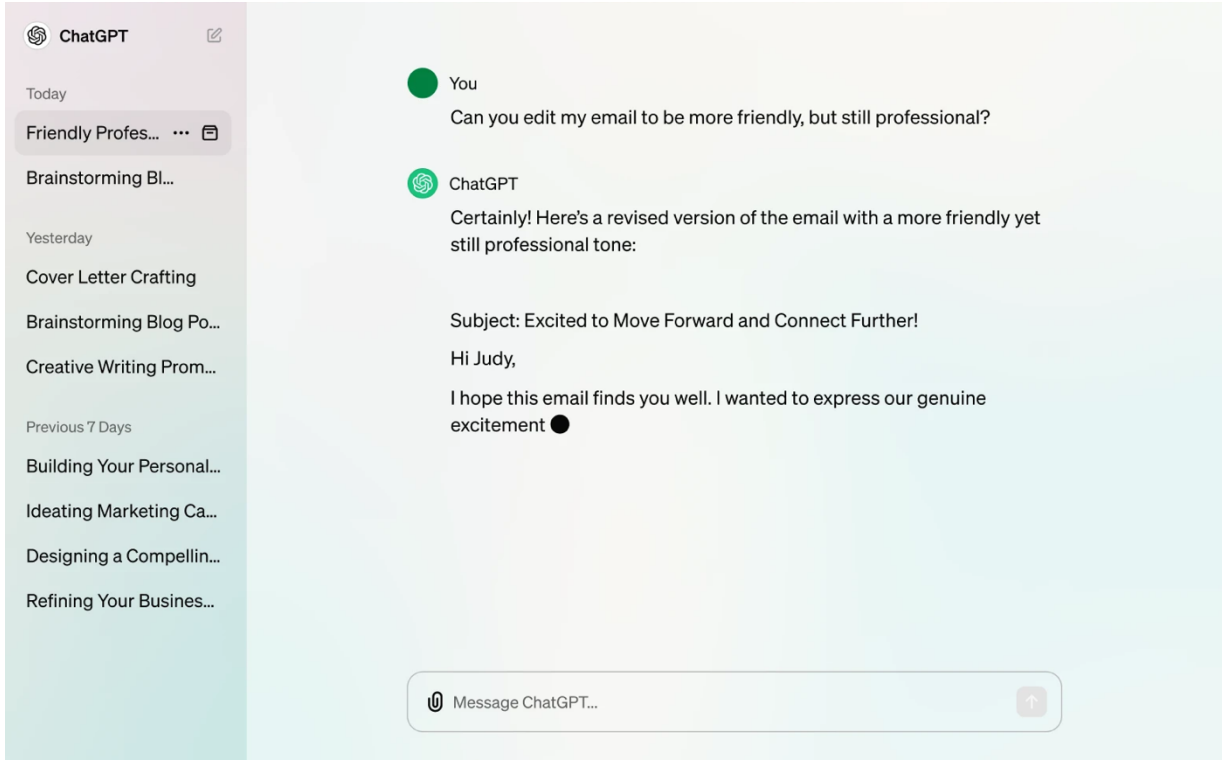
Classification with [CLS] Embedding

Topic Classification

The Houston Rockets won an intense overtime game	Sports
Bitcoin hit a new all-time high this week	Finance
Tesla launched a new self-driving software update	Technology
Flu cases are rising in several major cities	Health



Large Language Models with Classifiers

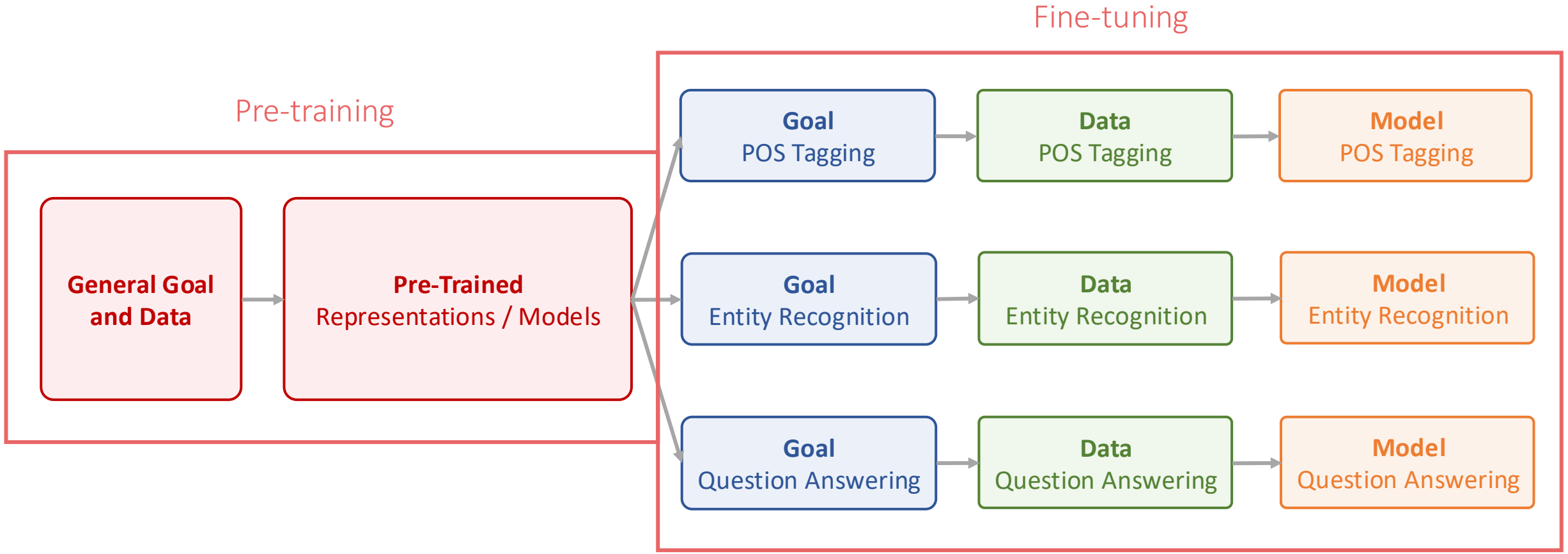


Normal model, math mode, code mode, …

Enable search, enable calculator, …

Ethical issue, harmful prompts, …

Pre-Training and Fine-Tuning

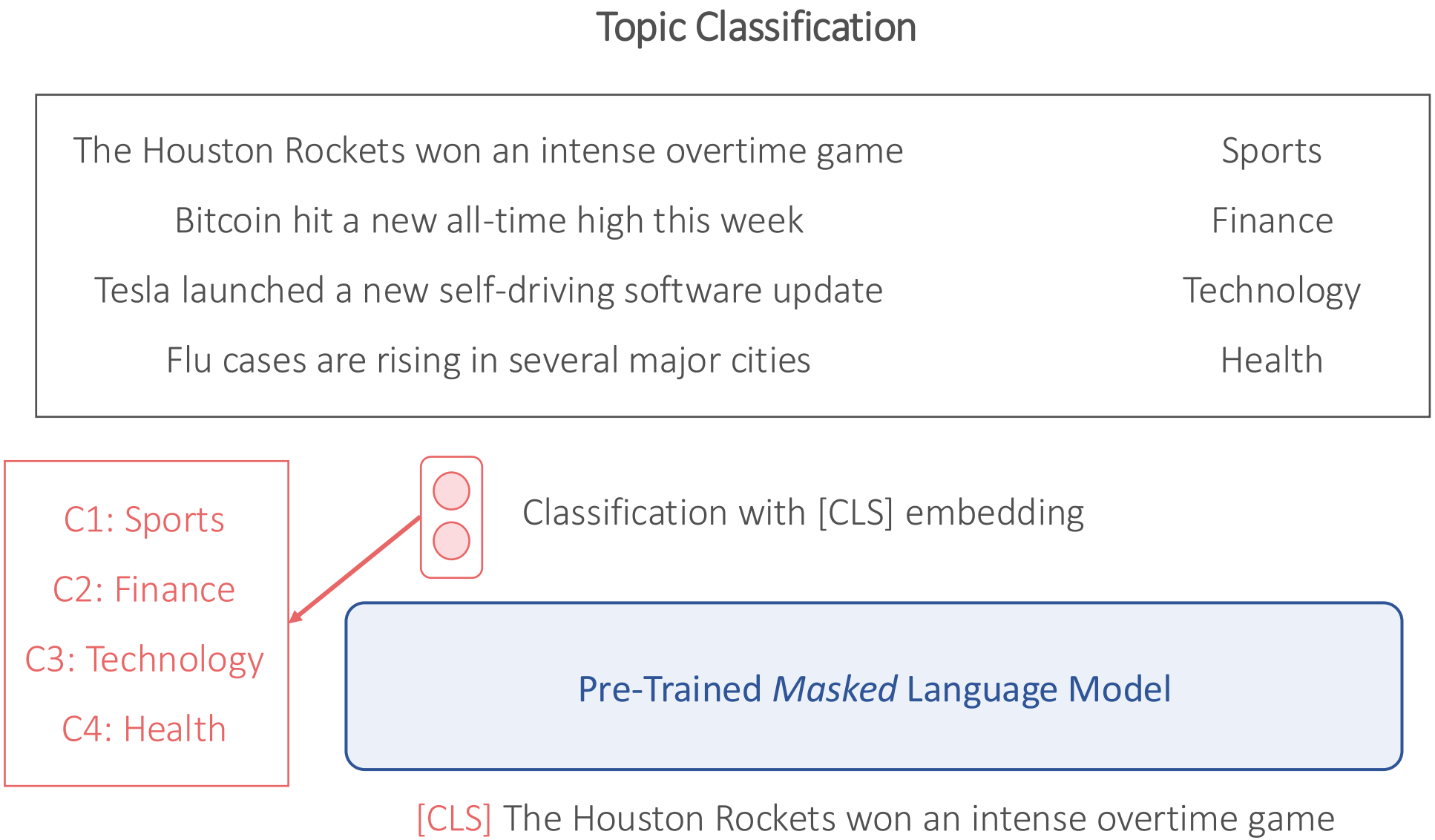


Saving the classifiers requires storage space!

Parameter-Efficient Fine-Tuning (PEFT)

- Motivation
  - Do not fine-tune the whole model
  - Update only a small subset of parameters
  - Keep most parameters frozen
  - Achieve comparable performance to full fine-tuning
- Advantages
  - Lower GPU memory usage
  - Faster training
  - Reduced storage cost
  - Easier deployment across tasks

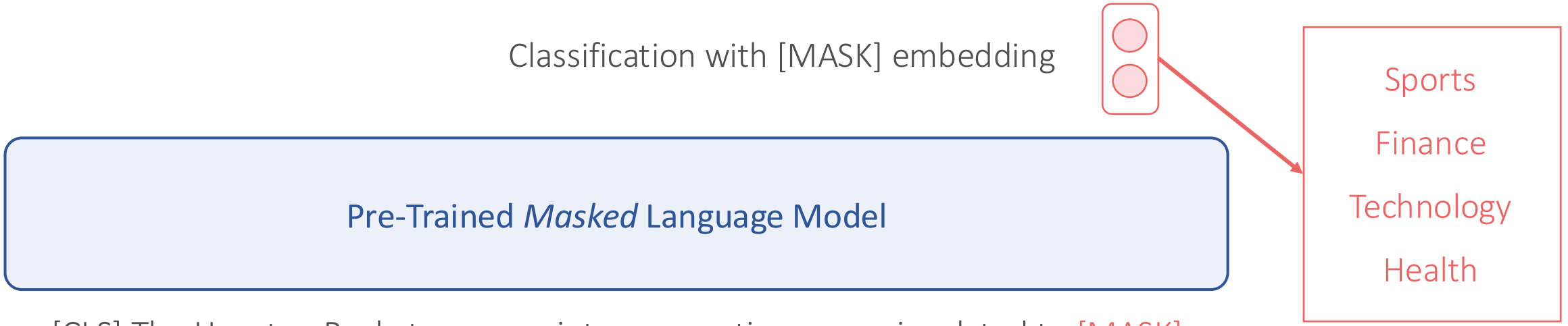
Classification with [CLS] Embedding



Classification with [MASK] Embedding

Topic Classification

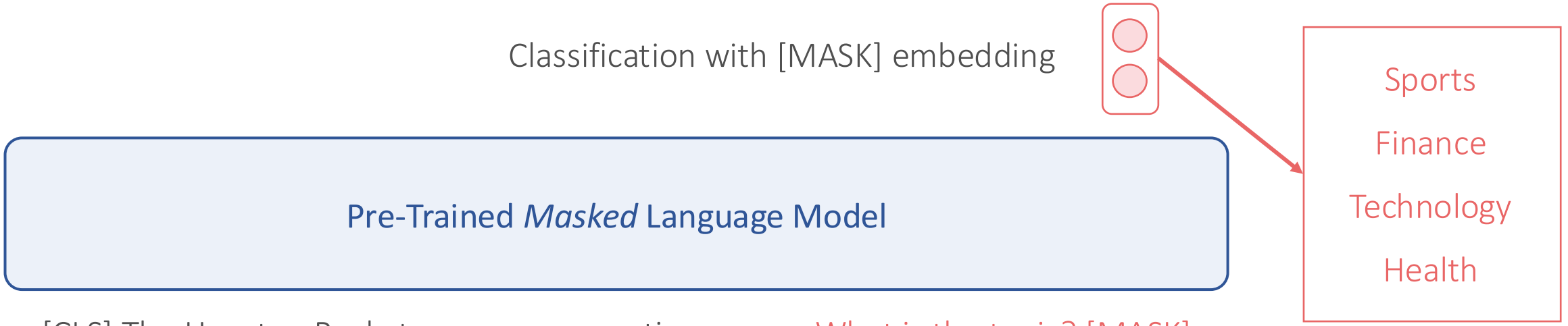
The Houston Rockets won an intense overtime game	Sports
Bitcoin hit a new all-time high this week	Finance
Tesla launched a new self-driving software update	Technology
Flu cases are rising in several major cities	Health



Classification with [MASK] Embedding and Prompt

Topic Classification

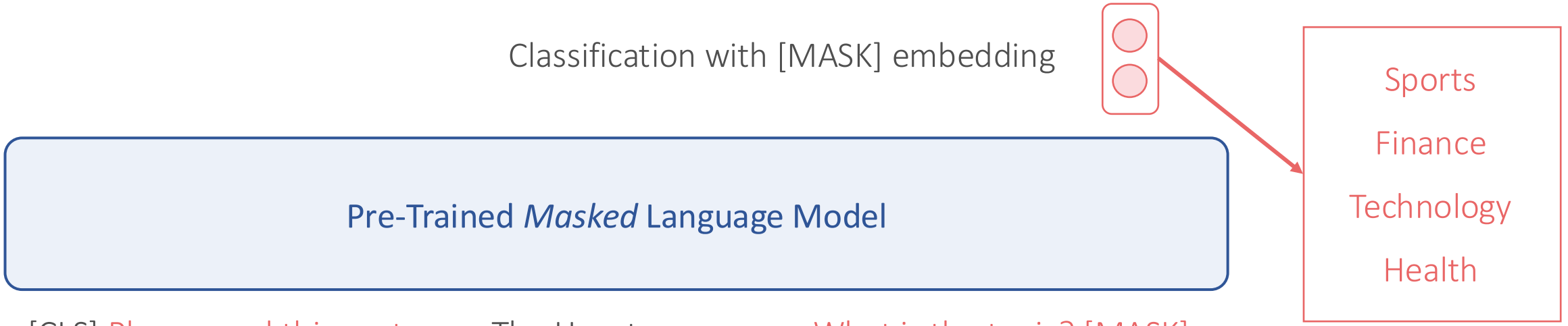
The Houston Rockets won an intense overtime game	Sports
Bitcoin hit a new all-time high this week	Finance
Tesla launched a new self-driving software update	Technology
Flu cases are rising in several major cities	Health



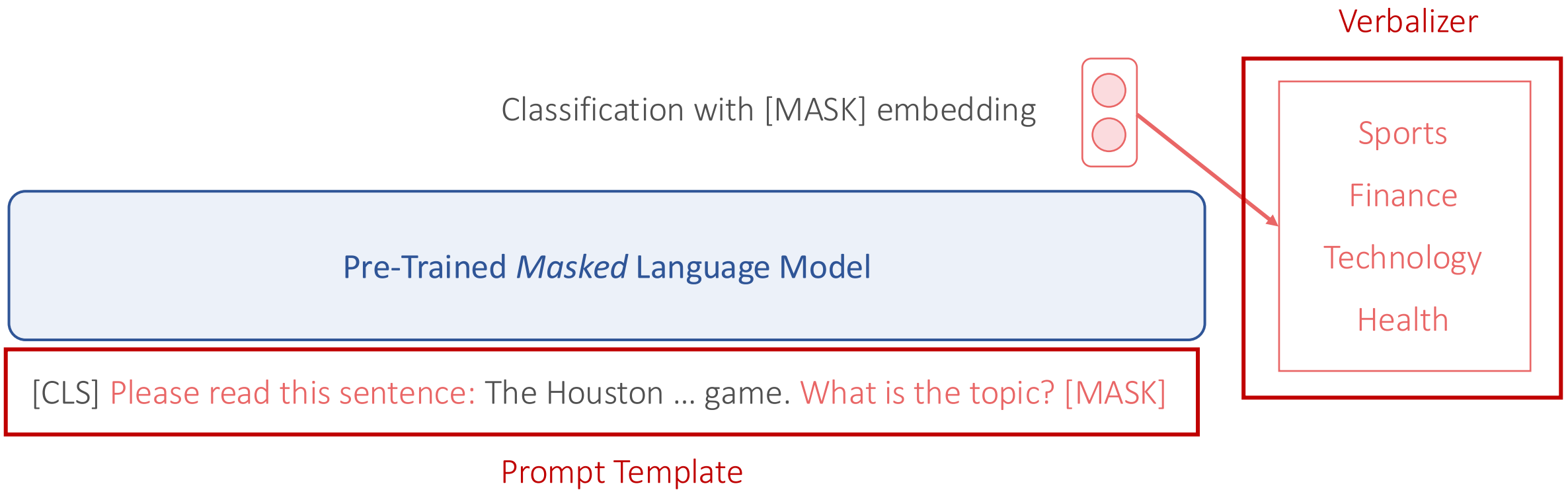
Classification with [MASK] Embedding and Prompt

Topic Classification

The Houston Rockets won an intense overtime game	Sports
Bitcoin hit a new all-time high this week	Finance
Tesla launched a new self-driving software update	Technology
Flu cases are rising in several major cities	Health

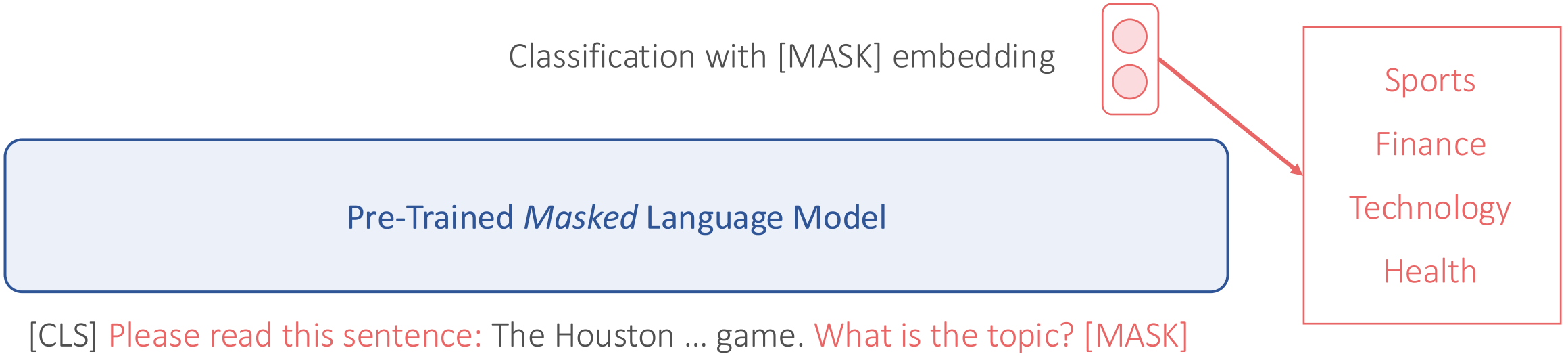


Prompt Tuning



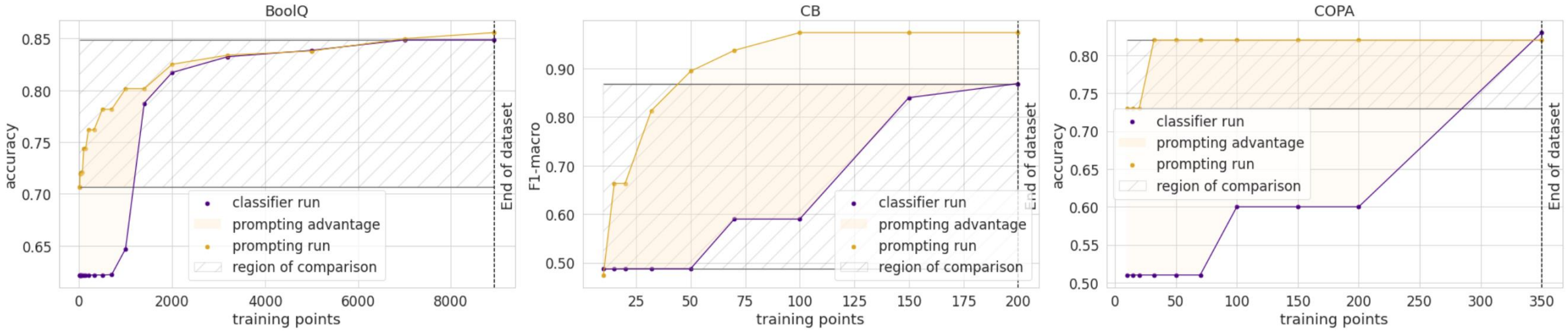
Prompt Tuning

- Better utilize label semantics and pre-trained knowledge
  - Verbalizer
- Can make zero-shot predictions



Prompt Tuning

- Better utilize label semantics and pre-trained knowledge
  - Verbalizer
- Can make zero-shot predictions
- Require less training examples



How Many Data Points is a Prompt Worth? 2021

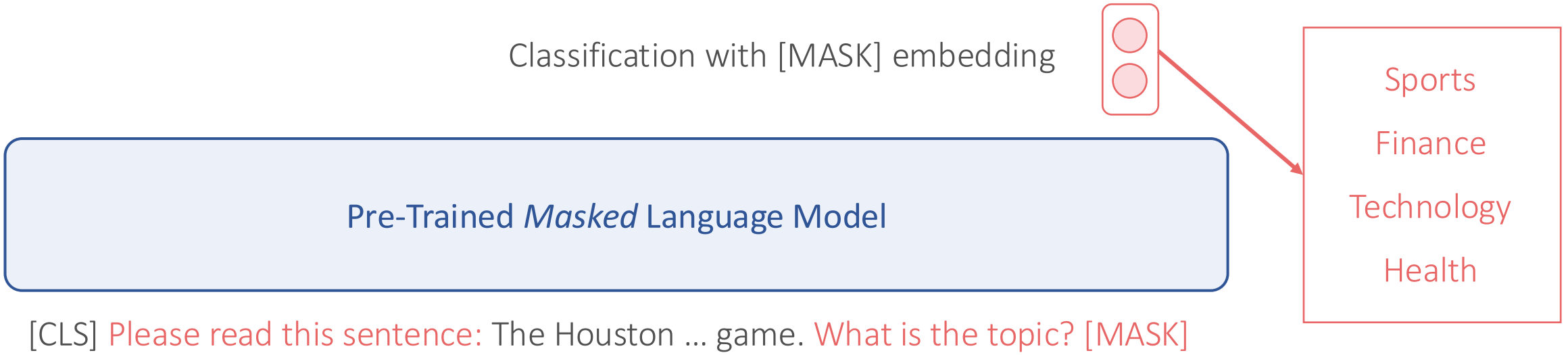
Issues of Discrete/Hard Prompts

- Manually design prompts can be difficult
  - Which one is the best?
- Pre-trained models are sensitive to prompts

Prompt	P@1
[X] is located in [Y]. *(original)*	31.29
[X] is located in which country or state? [Y].	19.78
[X] is located in which country? [Y].	31.40
[X] is located in which country? In [Y].	51.08

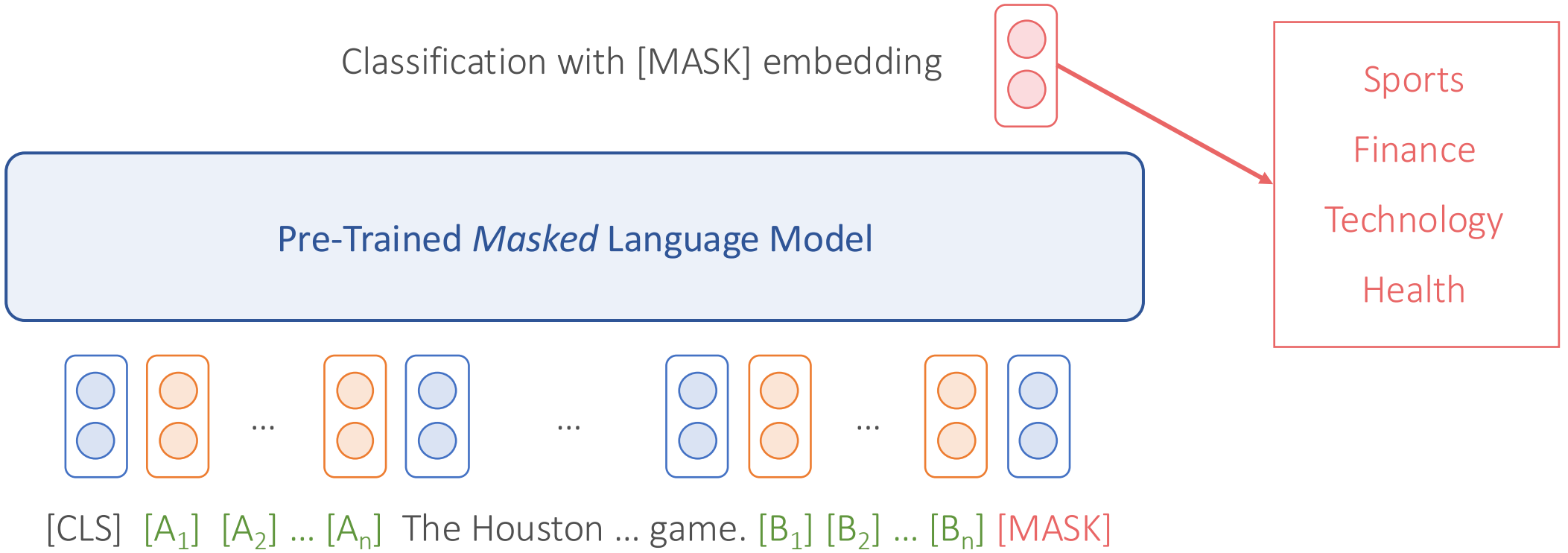
GPT Understands, Too, 2021

Hard Prompt Tuning



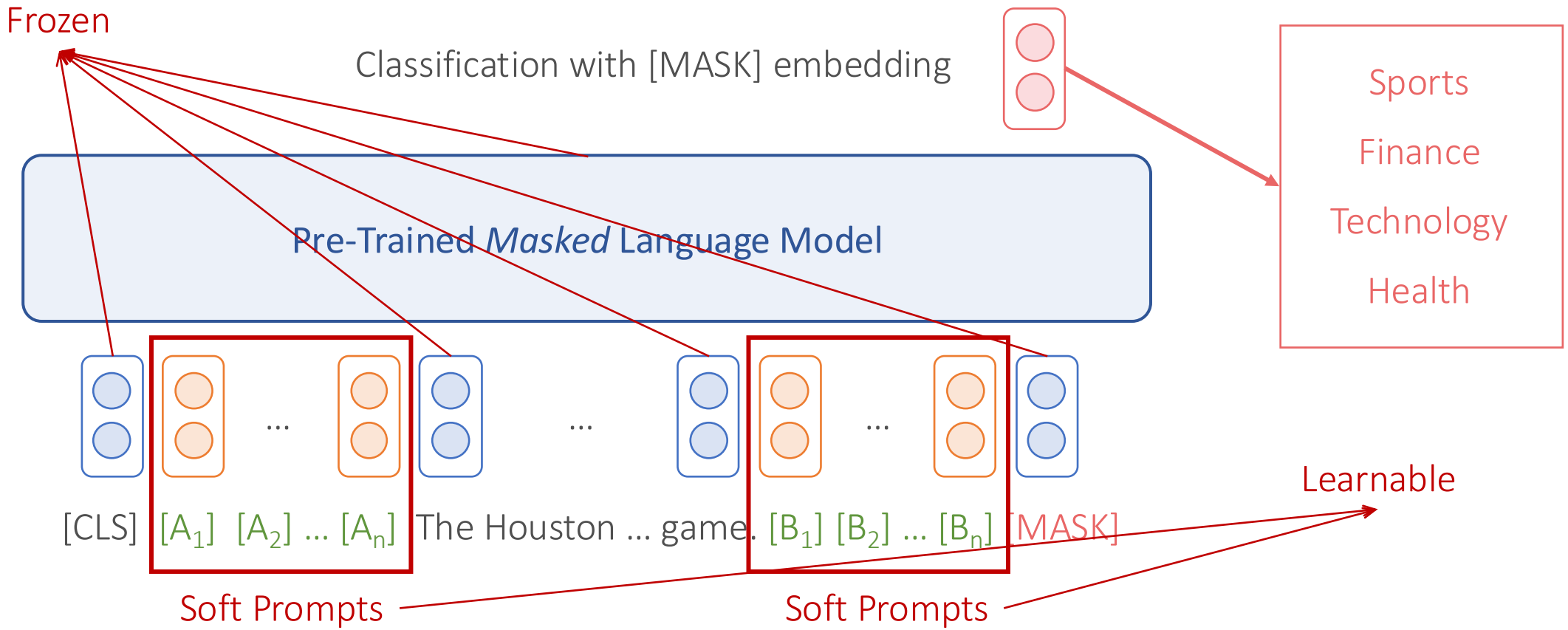
Soft Prompt Tuning

- Let model learn good prompts by itself



Soft Prompt Tuning

- Let model learn good prompts by itself

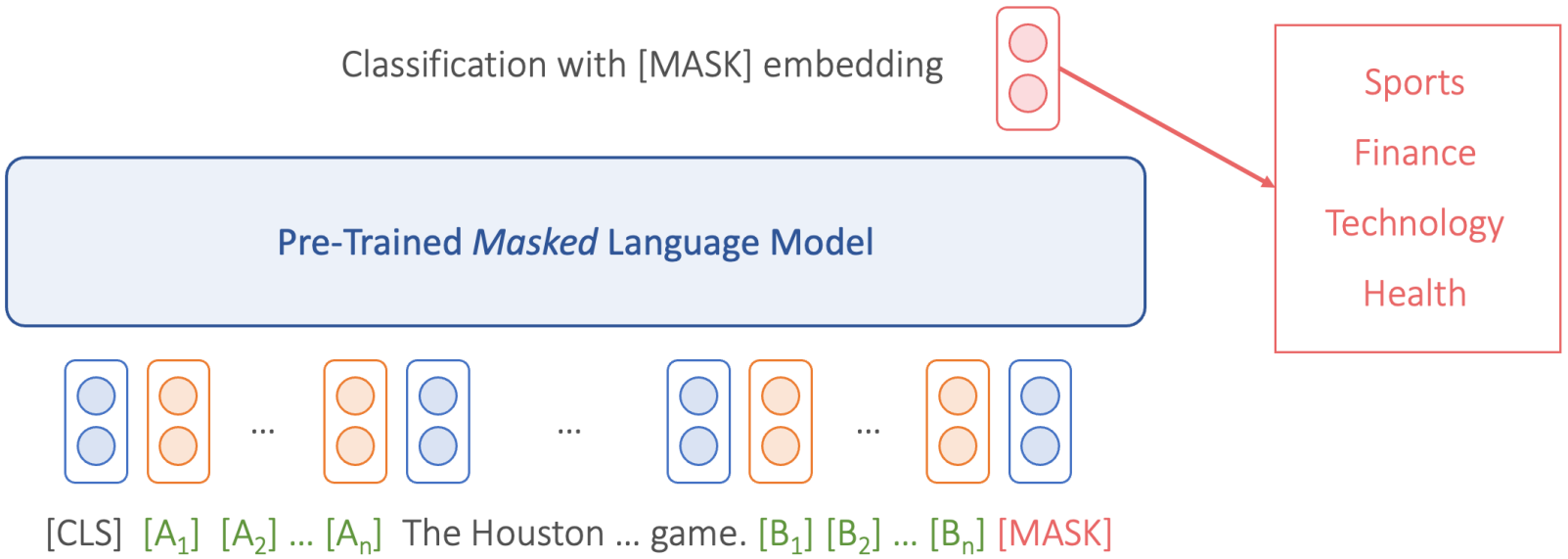


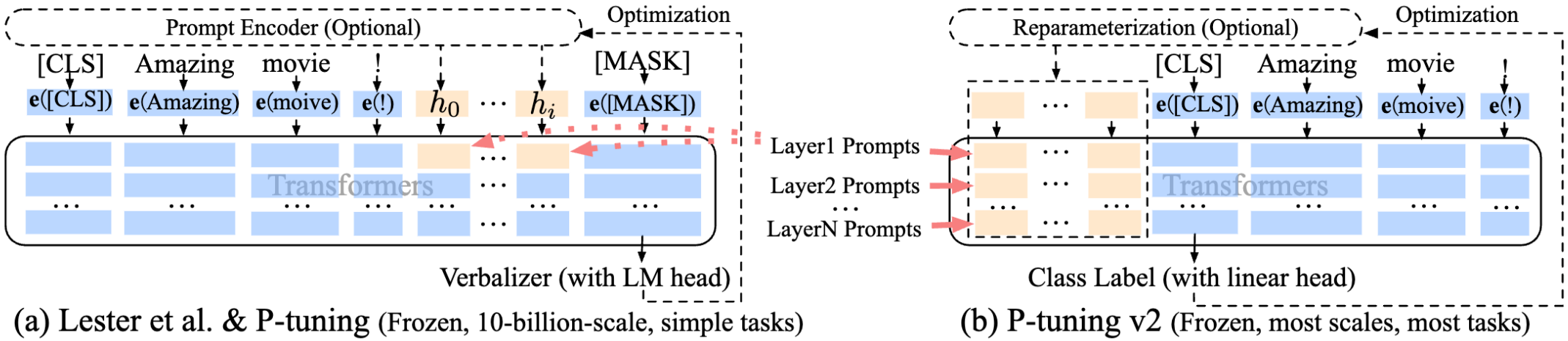
Soft Prompt Tuning

Prompt	$\mathcal{D}_{dev}$ Acc.
Does [PRE] agree with [HYP]? [MASK].	57.16
Does [HYP] agree with [PRE]? [MASK].	51.38
Premise: [PRE] Hypothesis: [HYP] Answer: [MASK].	68.59
[PRE] question: [HYP]. true or false? answer: [MASK].	70.15
P-tuning	76.45

GPT Understands, Too, 2021

From Prompt Tuning to Prefix Tuning



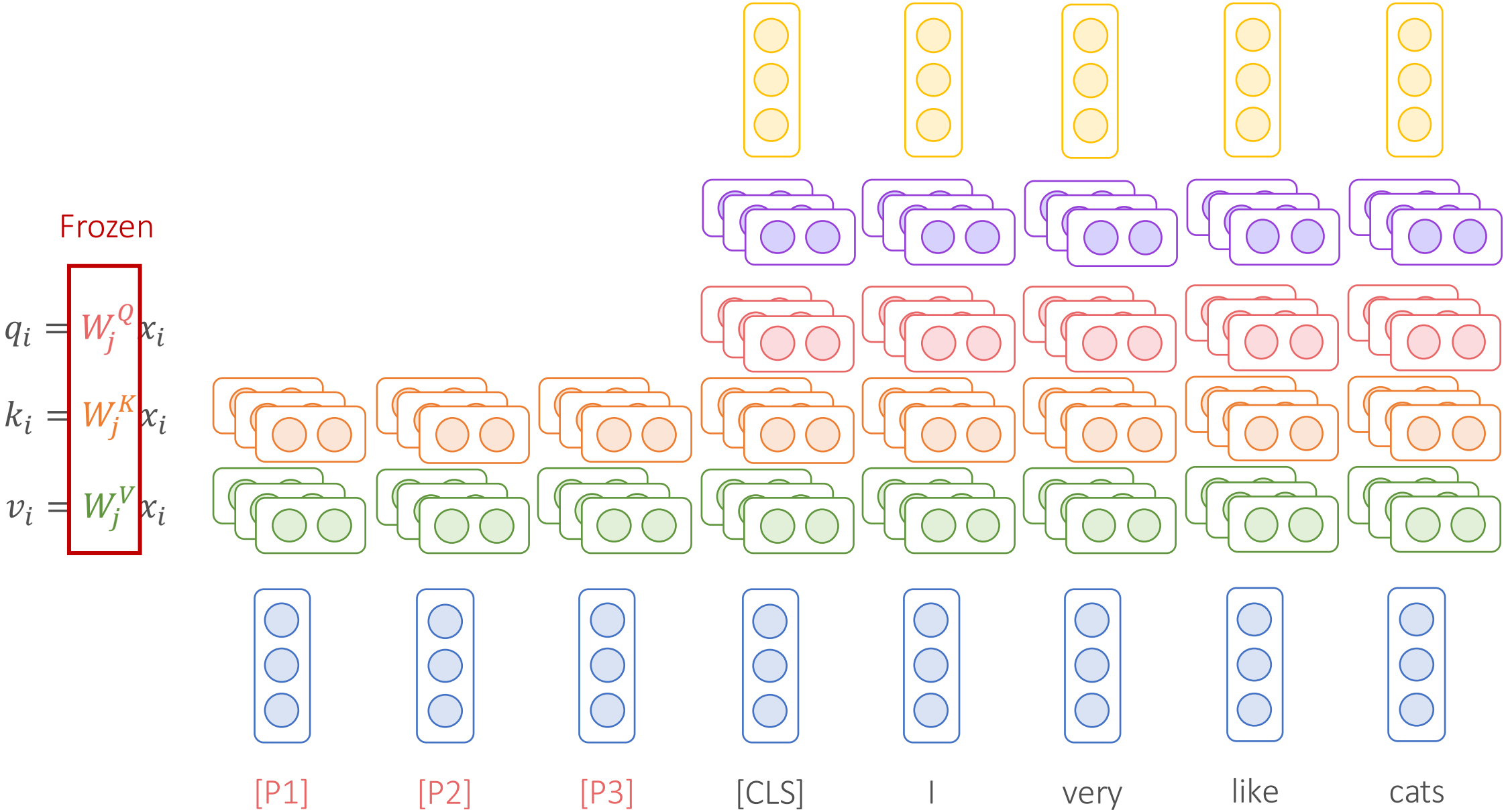


(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

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

P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks, 2021

Prefix Tuning

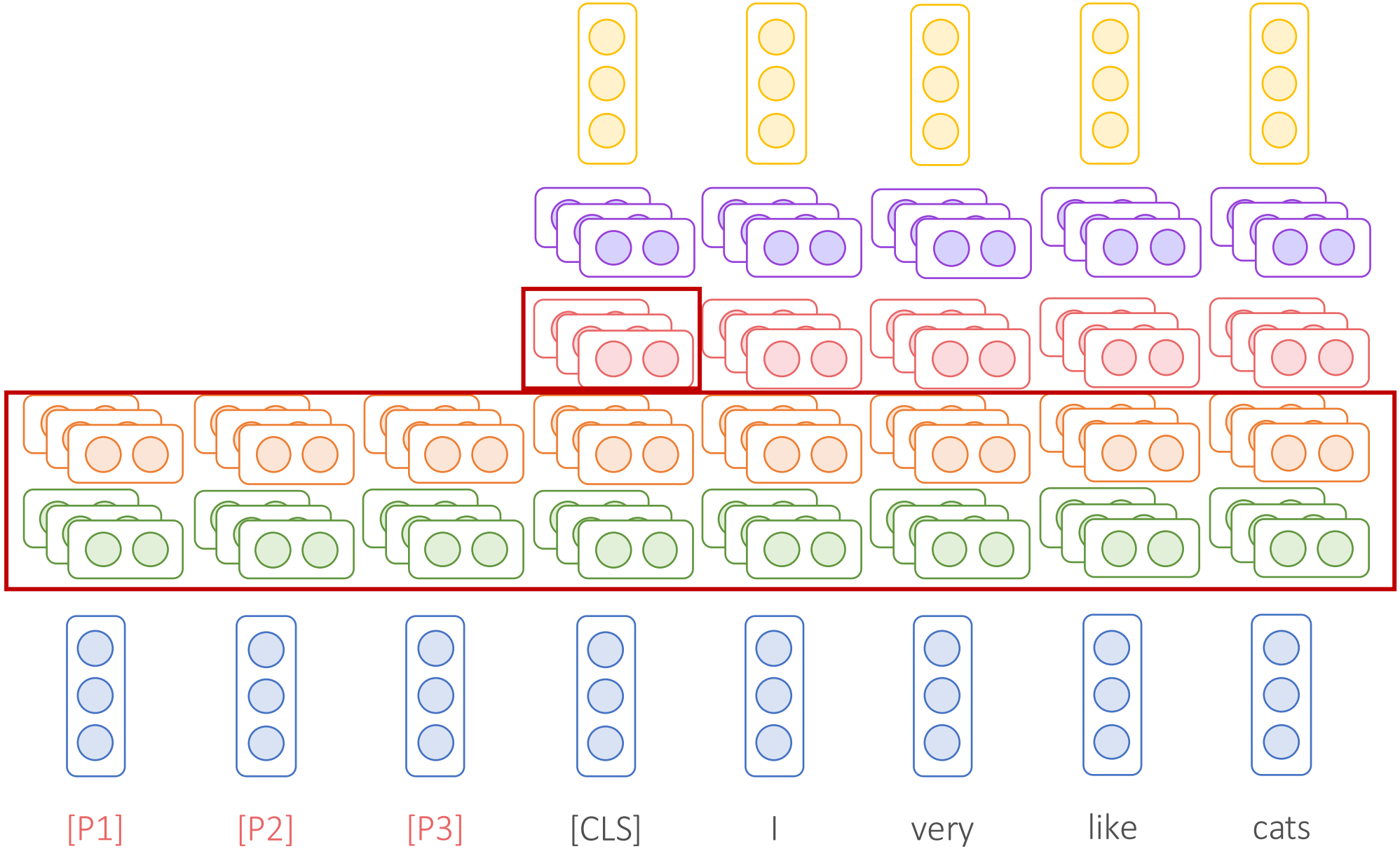


Prefix Tuning

$q_i = W_j^Q x_i$

$k_i = W_j^K x_i$

$v_i = W_j^V x_i$

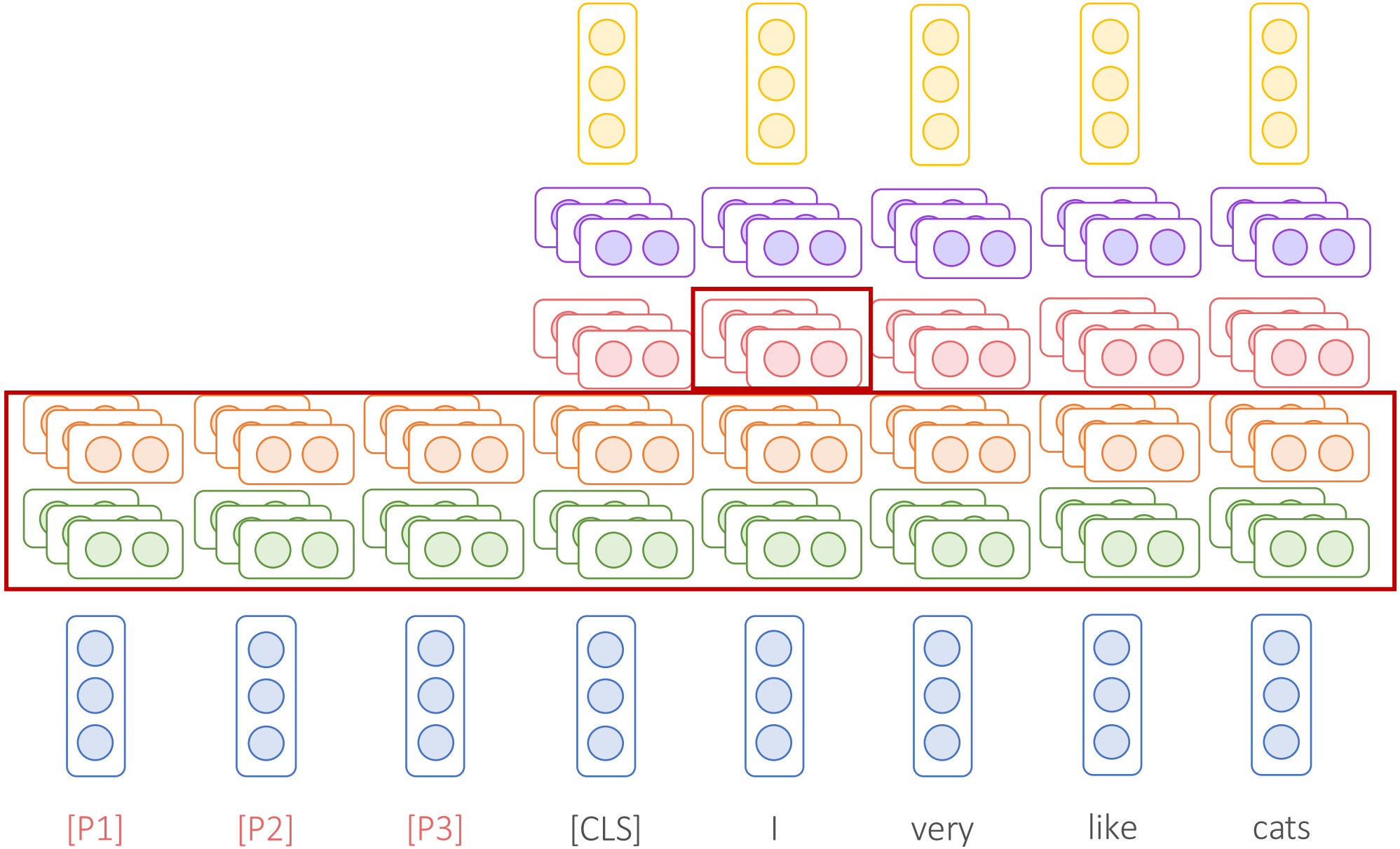


Prefix Tuning

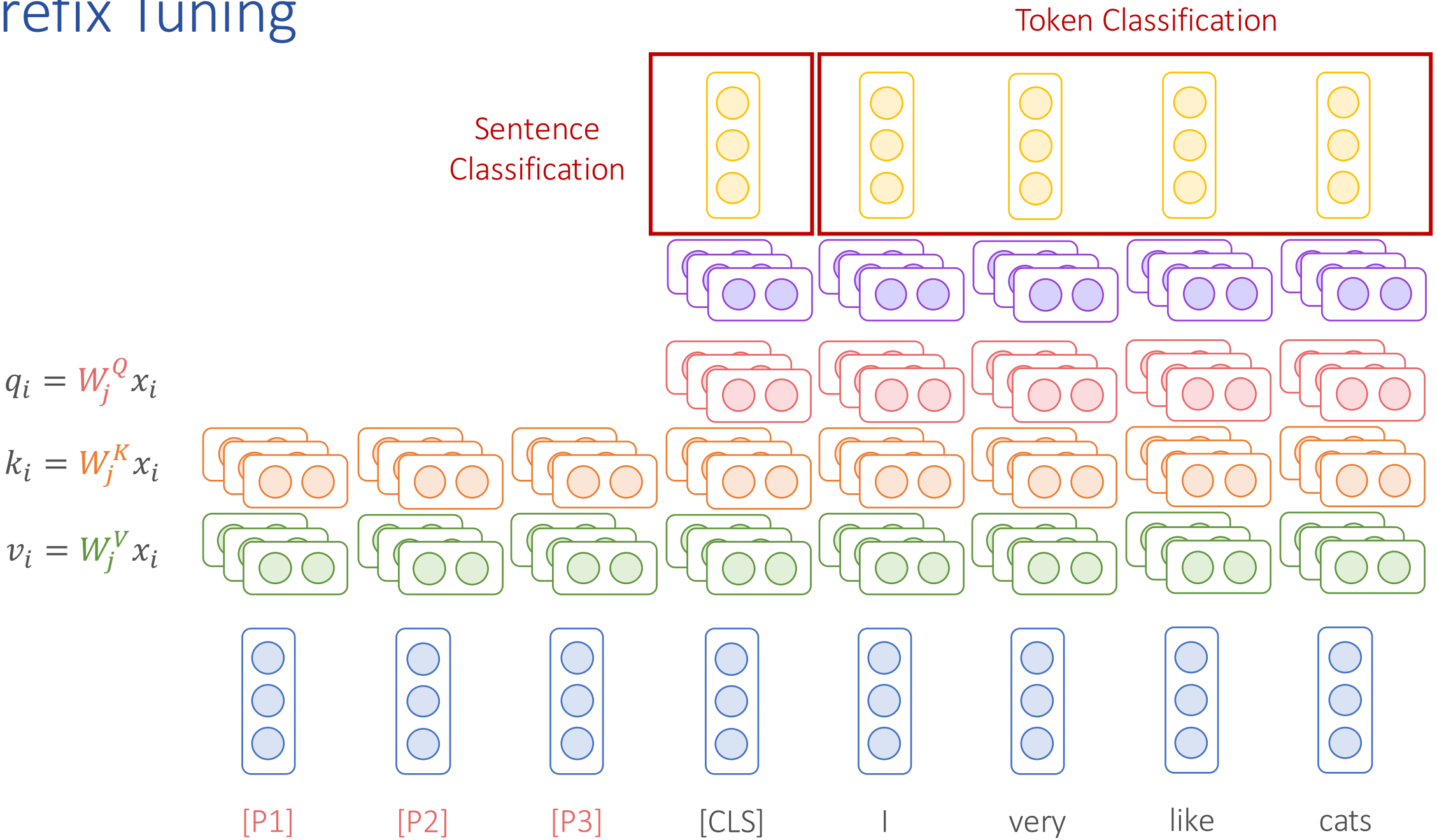
$q_i = W_j^Q x_i$

$k_i = W_j^K x_i$

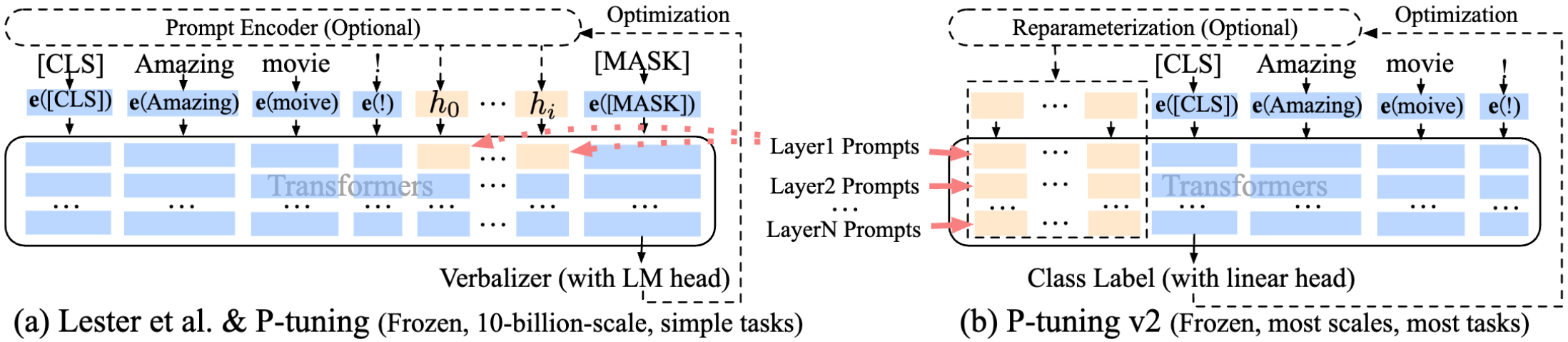
$v_i = W_j^V x_i$



Prefix Tuning



Prefix Tuning

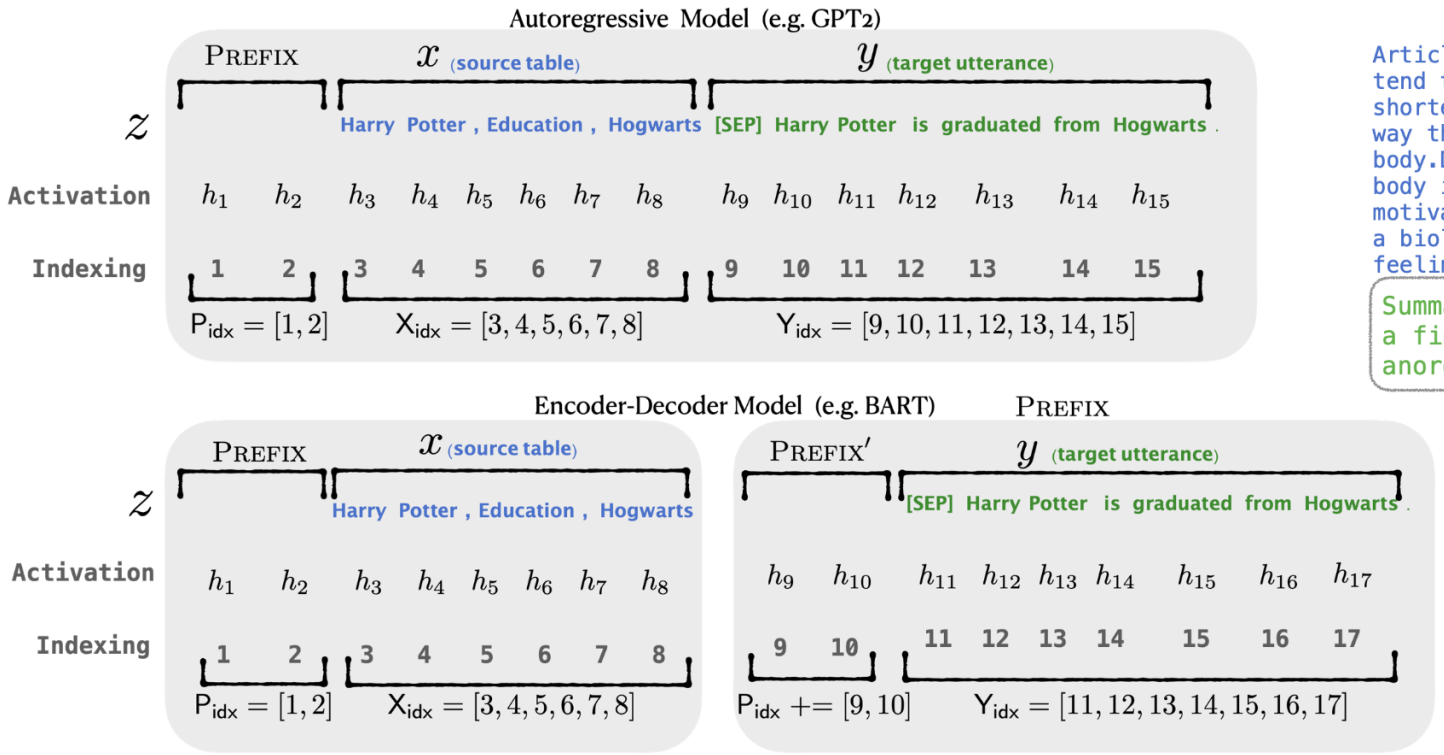


(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)

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

P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks, 2021

Prefix Tuning for Generation



Summarization Example

```
Article: Scientists at University College London discovered people
tend to think that their hands are wider and their fingers are
shorter than they truly are.They say the confusion may lie in the
way the brain receives information from different parts of the
body.Distorted perception may dominate in some people, leading to
body image problems ... [ignoring 308 words] could be very
motivating for people with eating disorders to know that there was
a biological explanation for their experiences, rather than
feeling it was their fault."

Summary: The brain naturally distorts body image -
a finding which could explain eating disorders like
anorexia, say experts.
```

Table-to-text Example

```
Table: name[Clowns] customer-
rating[1 out of 5] eatType[coffee
shop] food[Chinese] area[riverside]
near[Clare Hall]

Textual Description: Clowns is a
coffee shop in the riverside area
near Clare Hall that has a rating
1 out of 5 . They serve Chinese
food .
```


Prefix-Tuning: Optimizing Continuous Prompts for Generation, 2021

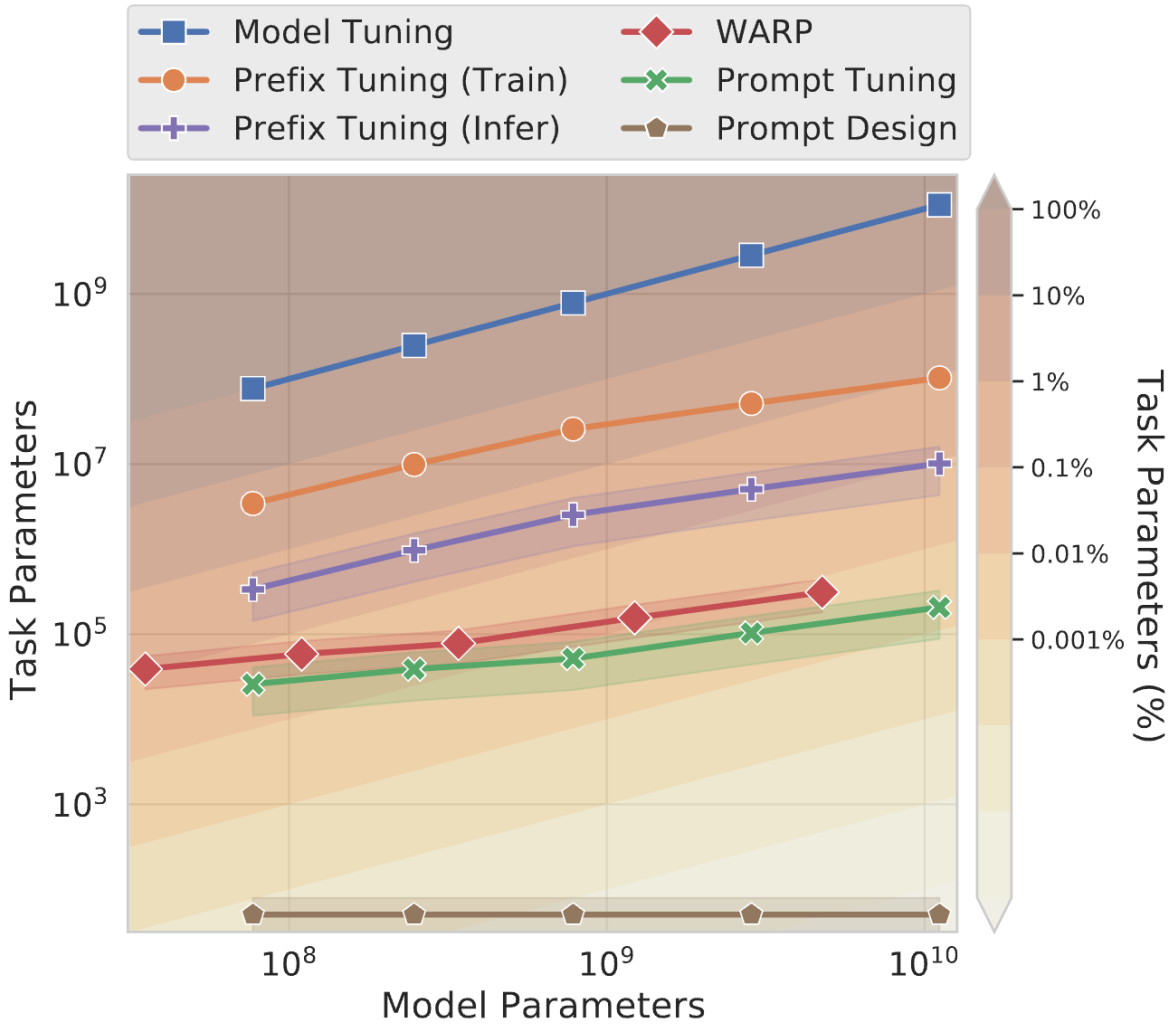
Prefix Tuning

	#Size	BoolQ			CB			COPA			MultiRC (F1a)		
		FT	PT	PT-2	FT	PT	PT-2	FT	PT	PT-2	FT	PT	PT-2
$\text{BERT}_{\text{large}}$	335M	**77.7**	67.2	75.8	**94.6**	80.4	**94.6**	69.0	55.0	**73.0**	70.5	59.6	**70.6**
$\text{RoBERTa}_{\text{large}}$	355M	**86.9**	62.3	84.8	98.2	71.4	**100**	**94.0**	63.0	93.0	**85.7**	59.9	82.5
$\text{GLM}_{\text{xlarge}}$	2B	**88.3**	79.7	87.0	**96.4**	76.4	**96.4**	**93.0**	92.0	91.0	84.1	77.5	**84.4**
$\text{GLM}_{\text{xxlarge}}$	10B	88.7	**88.8**	**88.8**	**98.7**	98.2	96.4	**98.0**	**98.0**	**98.0**	**88.1**	86.1	**88.1**

	#Size	ReCoRD (F1)			RTE			WiC			WSC		
		FT	PT	PT-2	FT	PT	PT-2	FT	PT	PT-2	FT	PT	PT-2
$\text{BERT}_{\text{large}}$	335M	70.6	44.2	**72.8**	70.4	53.5	**78.3**	74.9	63.0	**75.1**	**68.3**	64.4	**68.3**
$\text{RoBERTa}_{\text{large}}$	355M	89.0	46.3	**89.3**	86.6	58.8	**89.5**	**75.6**	56.9	73.4	63.5	**64.4**	63.5
$\text{GLM}_{\text{xlarge}}$	2B	91.8	82.7	**91.9**	**90.3**	85.6	**90.3**	**74.1**	71.0	72.0	**95.2**	87.5	92.3
$\text{GLM}_{\text{xxlarge}}$	10B	**94.4**	87.8	92.5	**93.1**	89.9	**93.1**	**75.7**	71.8	74.0	**95.2**	94.2	93.3

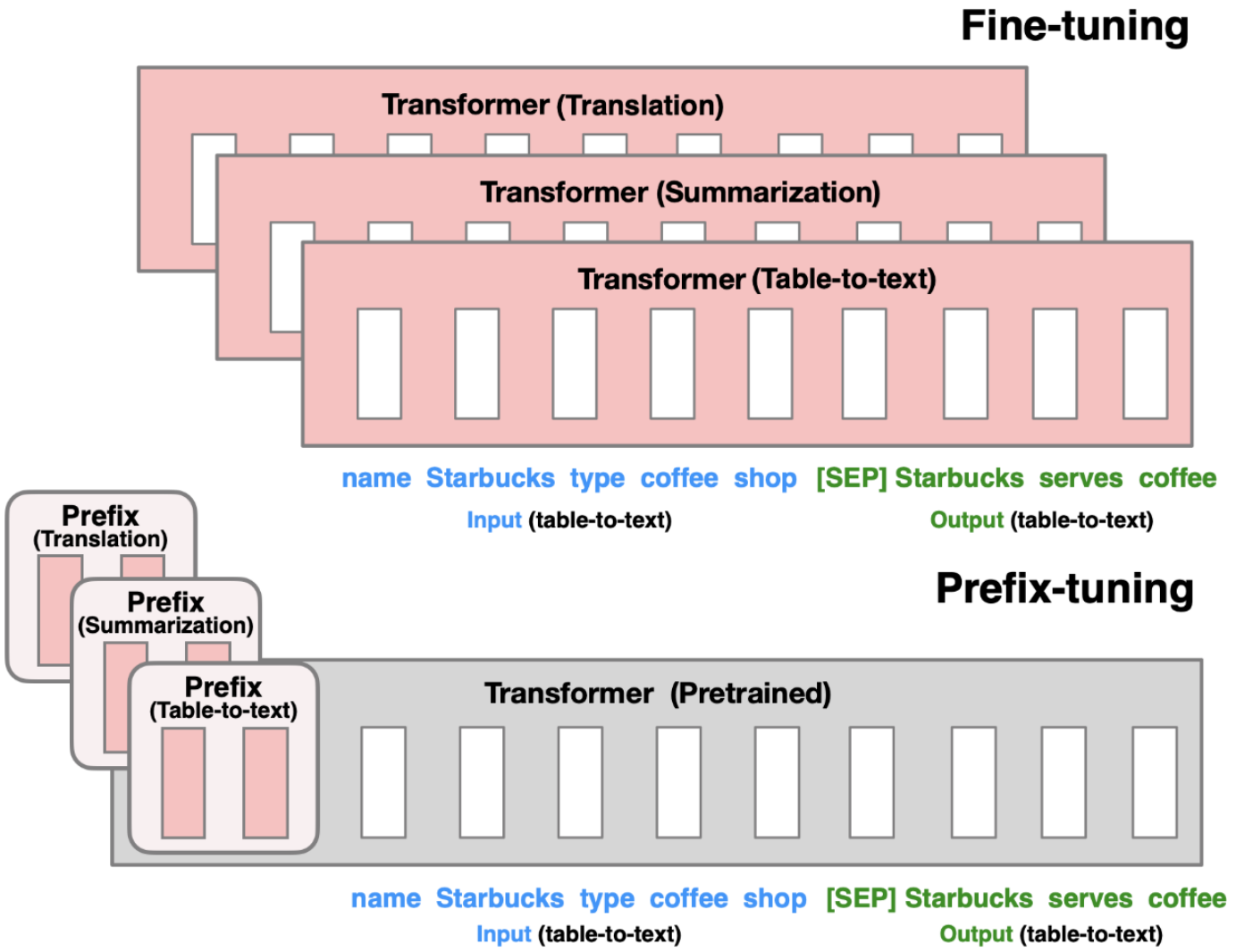
P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks, 2021

Prefix Tuning – Parameter-Efficient



The Power of Scale for Parameter-Efficient Prompt Tuning, 2021

Prefix Tuning – Parameter-Efficient

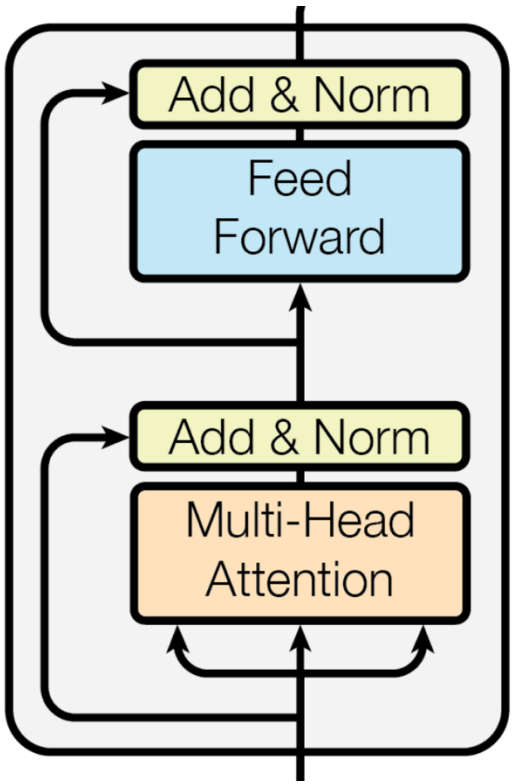


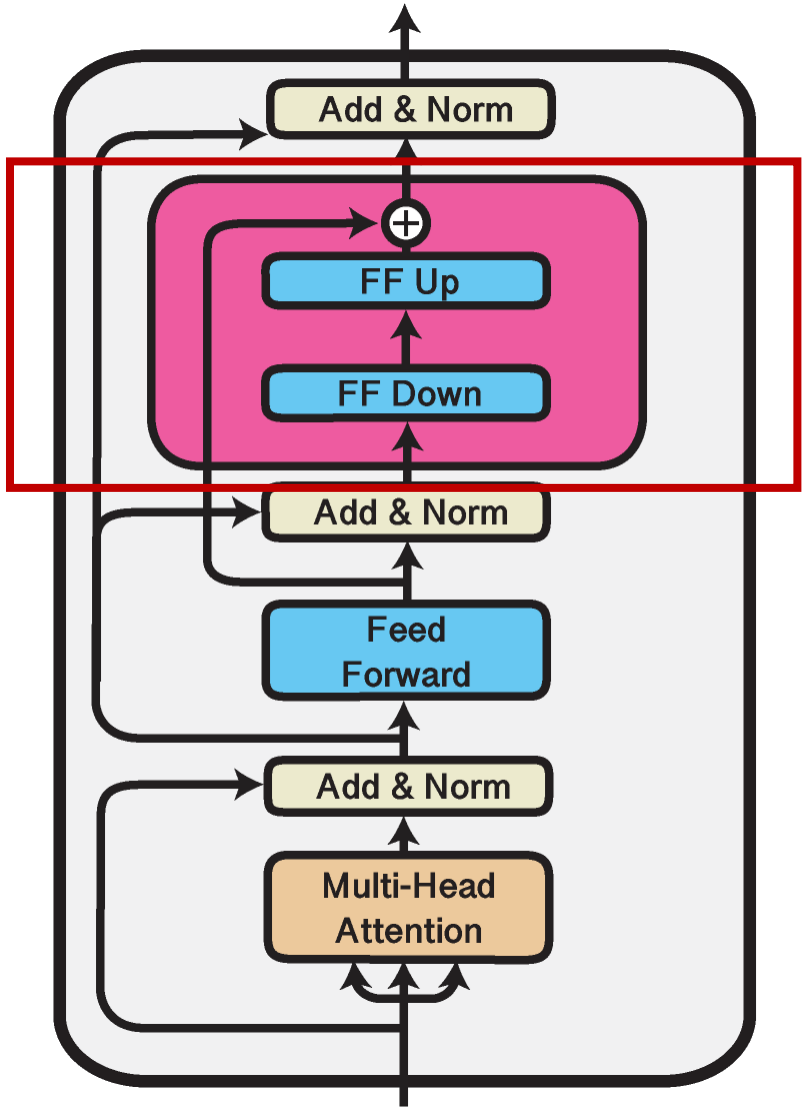
Prefix-Tuning: Optimizing Continuous Prompts for Generation, 2021

Parameter-Efficient Fine-Tuning (PEFT)

- Motivation
  - Do not fine-tune the whole model
  - Update only a small subset of parameters
  - Keep most parameters frozen
  - Achieve comparable performance to full fine-tuning
- Advantages
  - Lower GPU memory usage
  - Faster training
  - Reduced storage cost
  - Easier deployment across tasks

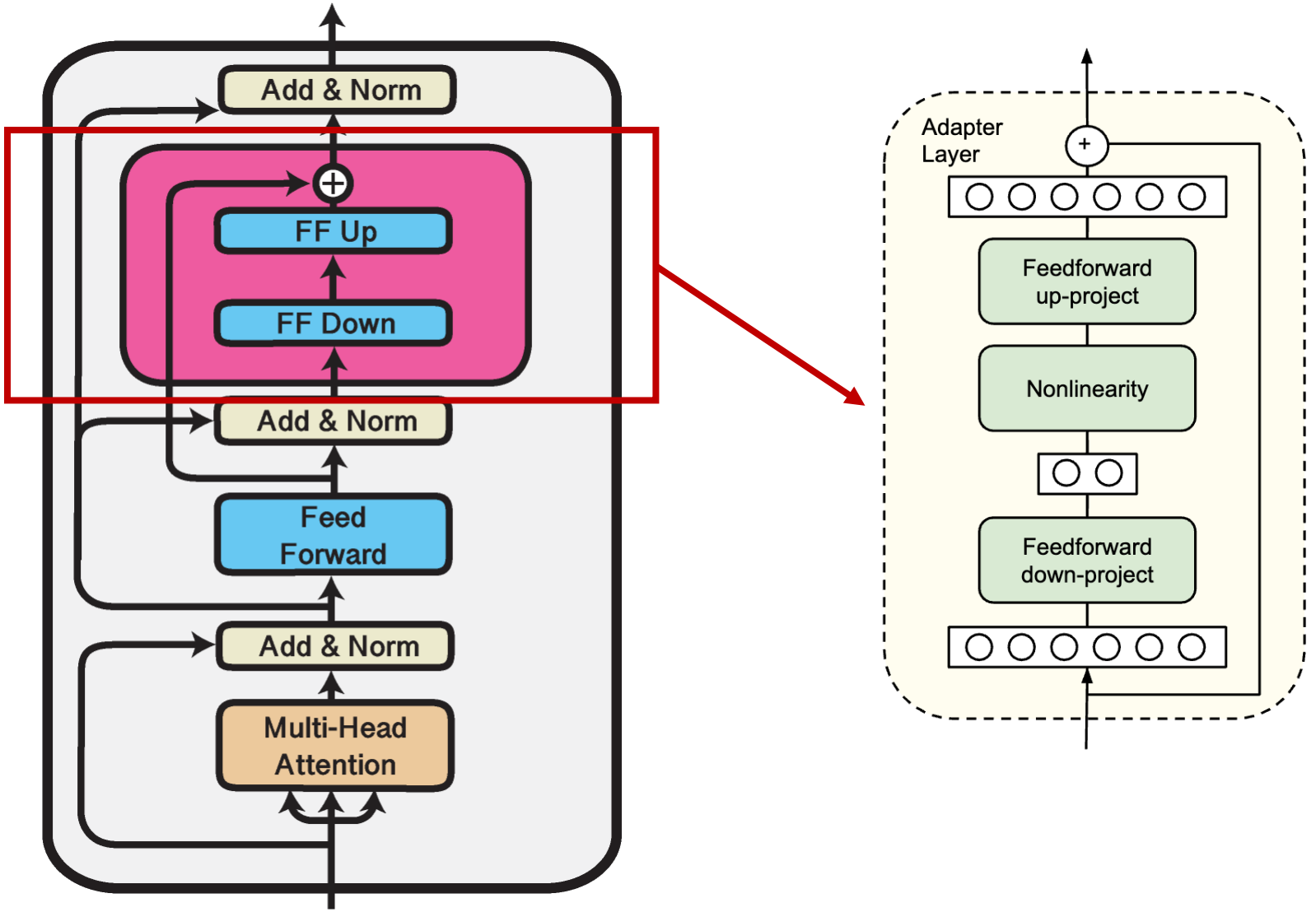
Adapter





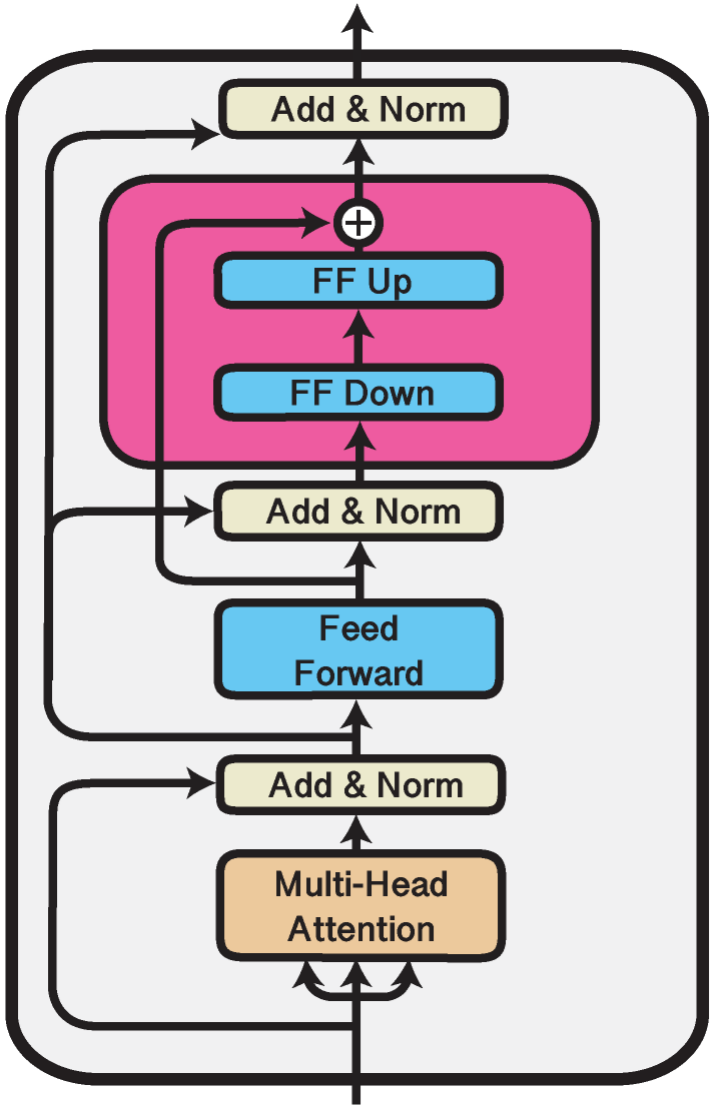
https://adapterhub.ml/blog/2022/03/adapter-transformers-v3-unifying-efficient-fine-tuning/

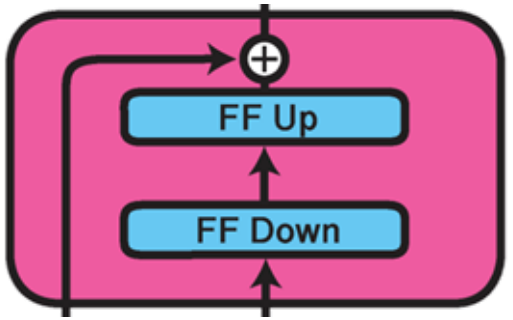
Adapter



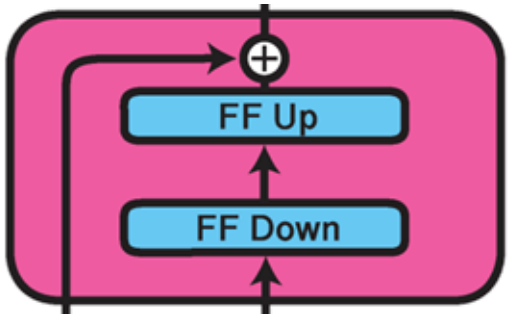
Parameter-Efficient Transfer Learning for NLP, 2019

Adapter

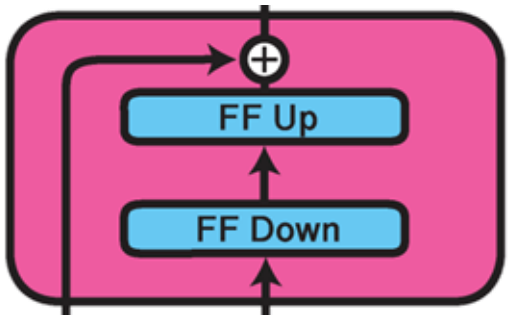




Task 1



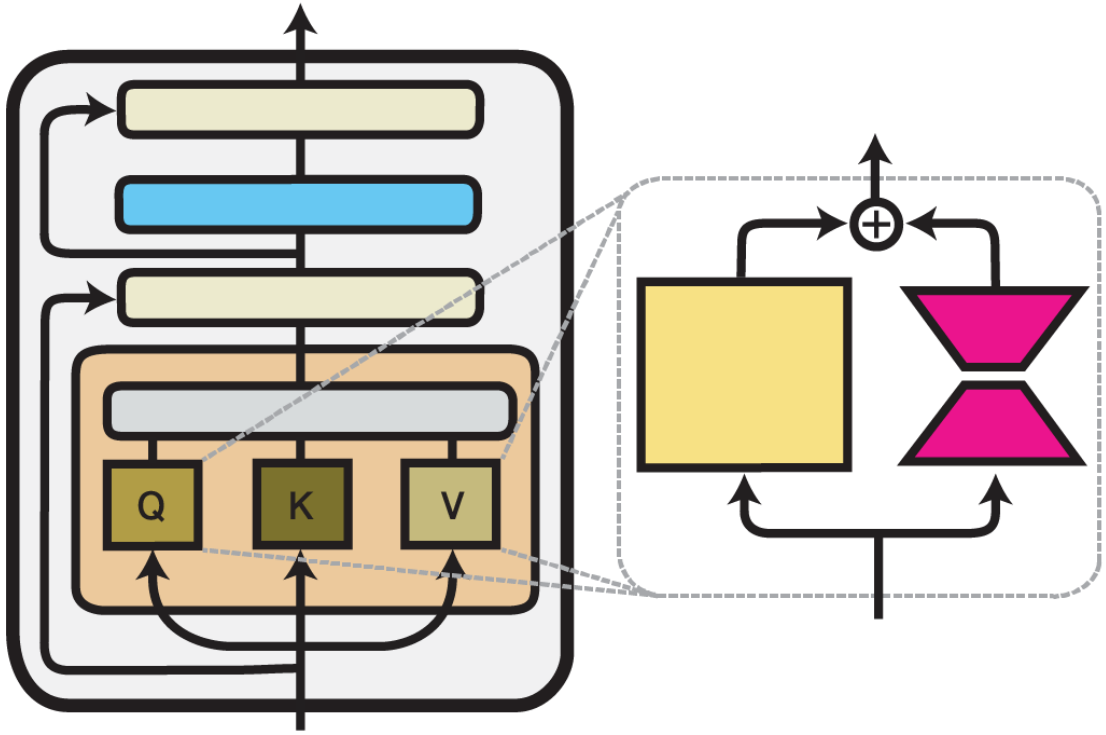
Task 2



Task 3

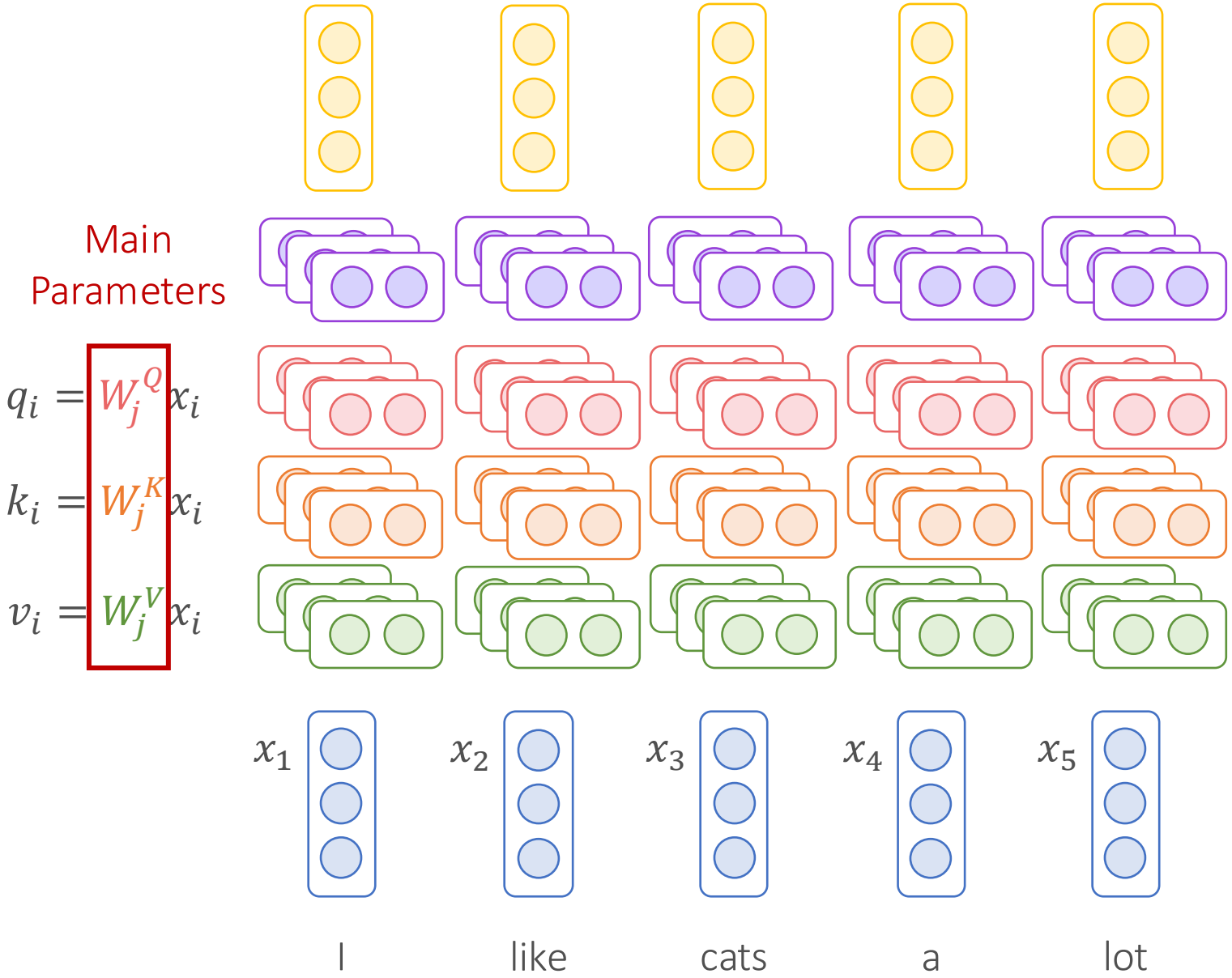
Parameter-Efficient Transfer Learning for NLP, 2019

Low-Rank Adaptation (LoRA)

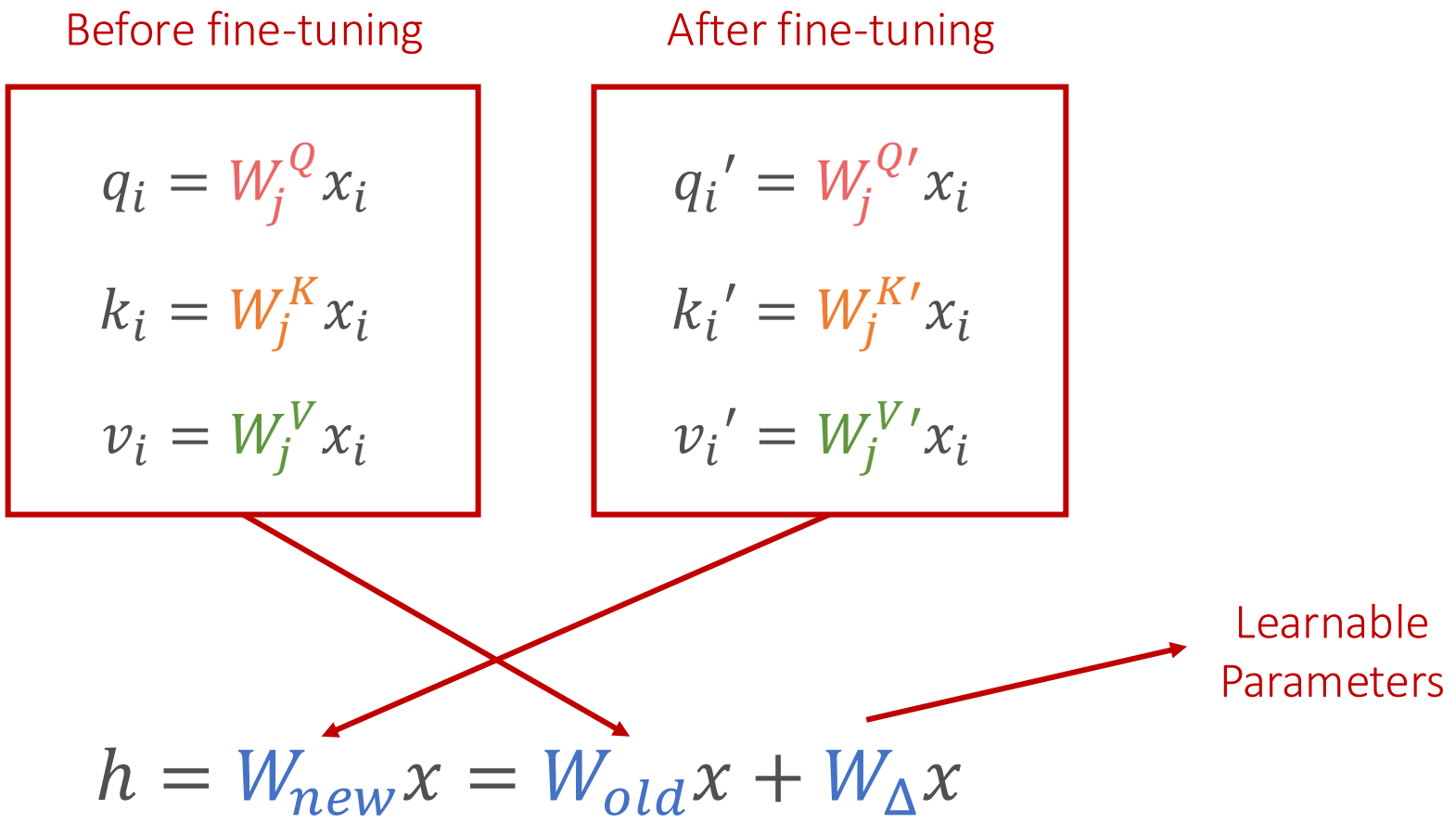


https://docs.adapterhub.ml/methods.html

Low-Rank Adaptation (LoRA)

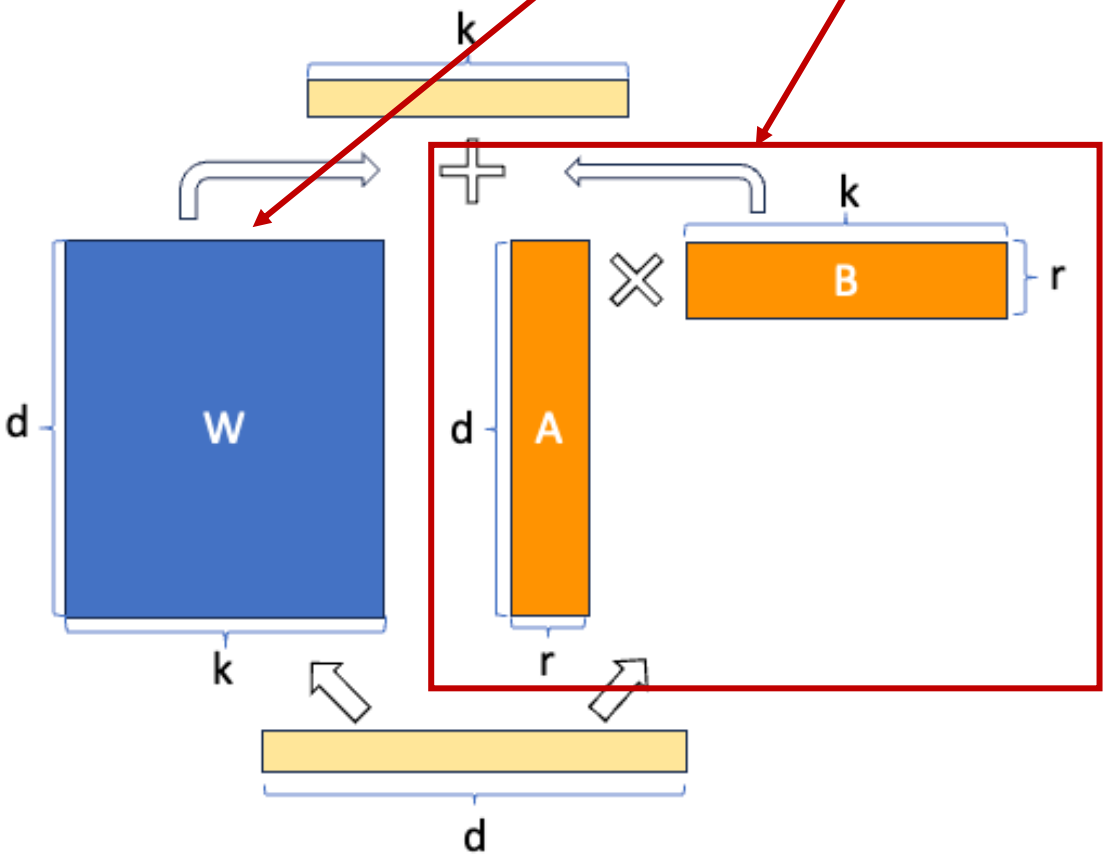


Low-Rank Adaptation (LoRA)



Low-Rank Adaptation (LoRA)

$$h = W_{new}x = W_{old}x + W_{\Delta}x$$



https://dataman-ai.medium.com/fine-tune-a-gpt-lora-e9b72ad4ad3

LoRA: Low-Rank Adaptation

Model & Method	# Trainable Parameters	MNLI	SST-2	MRPC	CoLA	QNLI	QQP	RTE	STS-B	Avg.
RoB_{base} (FT)*	125.0M	**87.6**	94.8	90.2	**63.6**	92.8	**91.9**	78.7	91.2	86.4
RoB_{base} (BitFit)*	0.1M	84.7	93.7	**92.7**	62.0	91.8	84.0	81.5	90.8	85.2
RoB_{base} (Adpt^{D})*	0.3M	$87.1_{\pm.0}$	$94.2_{\pm.1}$	$88.5_{\pm1.1}$	$60.8_{\pm.4}$	$93.1_{\pm.1}$	$90.2_{\pm.0}$	$71.5_{\pm2.7}$	$89.7_{\pm.3}$	84.4
RoB_{base} (Adpt^{D})*	0.9M	$87.3_{\pm.1}$	$94.7_{\pm.3}$	$88.4_{\pm.1}$	$62.6_{\pm.9}$	$93.0_{\pm.2}$	$90.6_{\pm.0}$	$75.9_{\pm2.2}$	$90.3_{\pm.1}$	85.4
RoB_{base} (LoRA)	0.3M	$87.5_{\pm.3}$	$\mathbf{95.1}_{\pm.2}$	$89.7_{\pm.7}$	$63.4_{\pm1.2}$	$\mathbf{93.3}_{\pm.3}$	$90.8_{\pm.1}$	$\mathbf{86.6}_{\pm.7}$	$\mathbf{91.5}_{\pm.2}$	**87.2**
$\text{RoB}_{\text{large}}$ (FT)*	355.0M	90.2	**96.4**	**90.9**	68.0	94.7	**92.2**	86.6	92.4	88.9
$\text{RoB}_{\text{large}}$ (LoRA)	0.8M	$\mathbf{90.6}_{\pm.2}$	$96.2_{\pm.5}$	$\mathbf{90.9}_{\pm1.2}$	$\mathbf{68.2}_{\pm1.9}$	$\mathbf{94.9}_{\pm.3}$	$91.6_{\pm.1}$	$\mathbf{87.4}_{\pm2.5}$	$\mathbf{92.6}_{\pm.2}$	**89.0**
$\text{RoB}_{\text{large}}$ (Adpt^{P})†	3.0M	$90.2_{\pm.3}$	$96.1_{\pm.3}$	$90.2_{\pm.7}$	$\mathbf{68.3}_{\pm1.0}$	$\mathbf{94.8}_{\pm.2}$	$\mathbf{91.9}_{\pm.1}$	$83.8_{\pm2.9}$	$92.1_{\pm.7}$	88.4
$\text{RoB}_{\text{large}}$ (Adpt^{P})†	0.8M	$\mathbf{90.5}_{\pm.3}$	$\mathbf{96.6}_{\pm.2}$	$89.7_{\pm1.2}$	$67.8_{\pm2.5}$	$\mathbf{94.8}_{\pm.3}$	$91.7_{\pm.2}$	$80.1_{\pm2.9}$	$91.9_{\pm.4}$	87.9
$\text{RoB}_{\text{large}}$ (Adpt^{H})†	6.0M	$89.9_{\pm.5}$	$96.2_{\pm.3}$	$88.7_{\pm2.9}$	$66.5_{\pm4.4}$	$94.7_{\pm.2}$	$92.1_{\pm.1}$	$83.4_{\pm1.1}$	$91.0_{\pm1.7}$	87.8
$\text{RoB}_{\text{large}}$ (Adpt^{H})†	0.8M	$90.3_{\pm.3}$	$96.3_{\pm.5}$	$87.7_{\pm1.7}$	$66.3_{\pm2.0}$	$94.7_{\pm.2}$	$91.5_{\pm.1}$	$72.9_{\pm2.9}$	$91.5_{\pm.5}$	86.4
$\text{RoB}_{\text{large}}$ (LoRA)†	0.8M	$\mathbf{90.6}_{\pm.2}$	$96.2_{\pm.5}$	$\mathbf{90.2}_{\pm1.0}$	$68.2_{\pm1.9}$	$\mathbf{94.8}_{\pm.3}$	$91.6_{\pm.2}$	$\mathbf{85.2}_{\pm1.1}$	$\mathbf{92.3}_{\pm.5}$	**88.6**
DeB_{XXL} (FT)*	1500.0M	91.8	**97.2**	92.0	72.0	**96.0**	92.7	93.9	92.9	91.1
DeB_{XXL} (LoRA)	4.7M	$\mathbf{91.9}_{\pm.2}$	$96.9_{\pm.2}$	$\mathbf{92.6}_{\pm.6}$	$\mathbf{72.4}_{\pm1.1}$	$\mathbf{96.0}_{\pm.1}$	$\mathbf{92.9}_{\pm.1}$	$\mathbf{94.9}_{\pm.4}$	$\mathbf{93.0}_{\pm.2}$	**91.3**

LoRA: Low-Rank Adaptation of Large Language Models, 2021

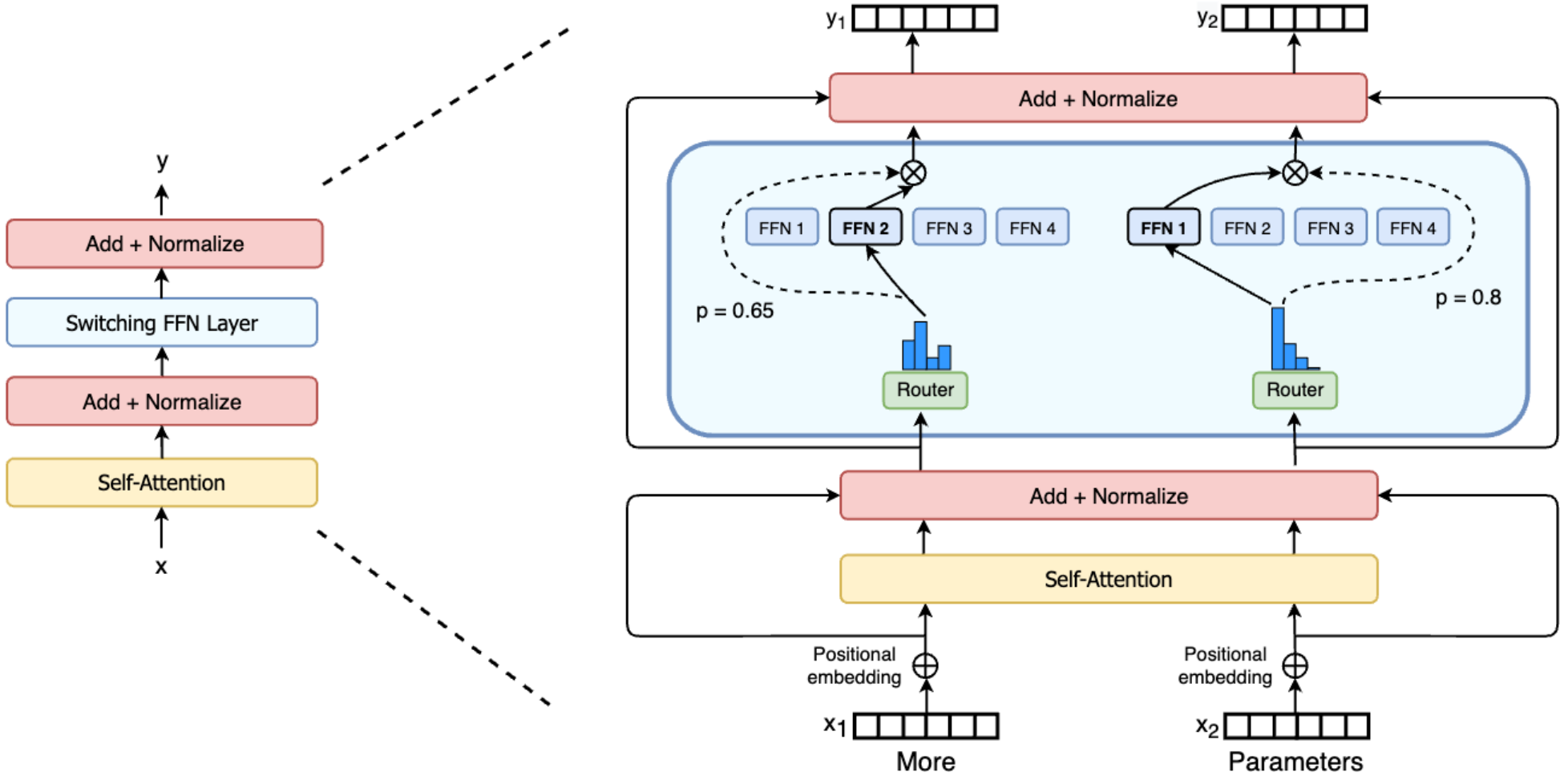
Lecture Plan

- Parameter-Efficient Fine-Tuning
  - Prompt Tuning, Prefix Tuning, Adapter
  - Low-Rank Adaptation (LoRA)
- Efficient Architecture
  - Mixture of Experts (MoE)
- Model Compression
  - Pruning, Quantization
  - Distillation
- Inference
  - KV Cache

Mixture of Experts (MoE)

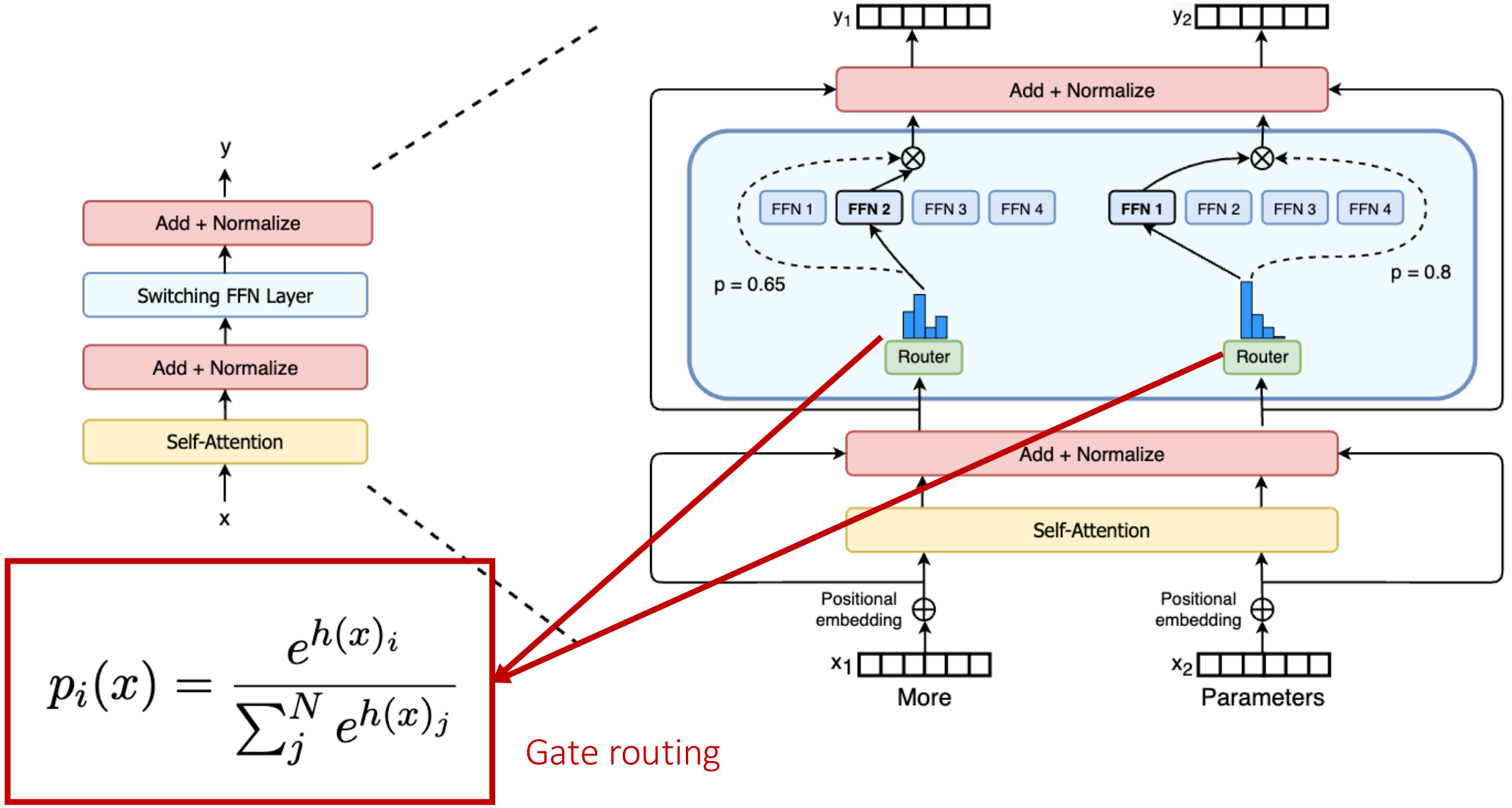
- Transformers with multiple specialized “expert” adapters
- A gating network selects experts per input
- Only only a subset of experts are activated per token/input

Mixture of Experts (MoE)



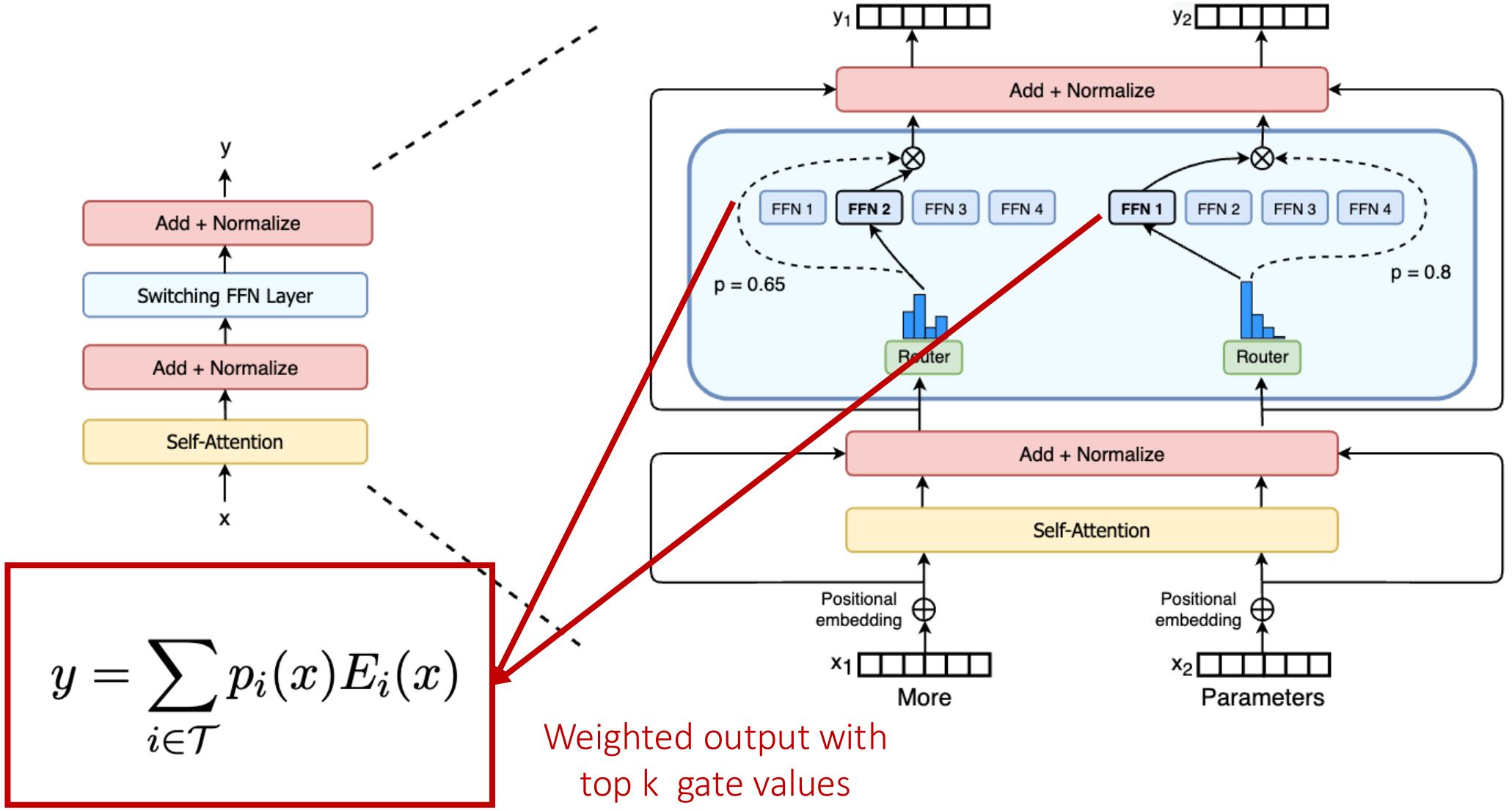
Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity, 2022

Mixture of Experts (MoE)



Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity, 2022

Mixture of Experts (MoE)



Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity, 2022

Mixture of Experts (MoE)

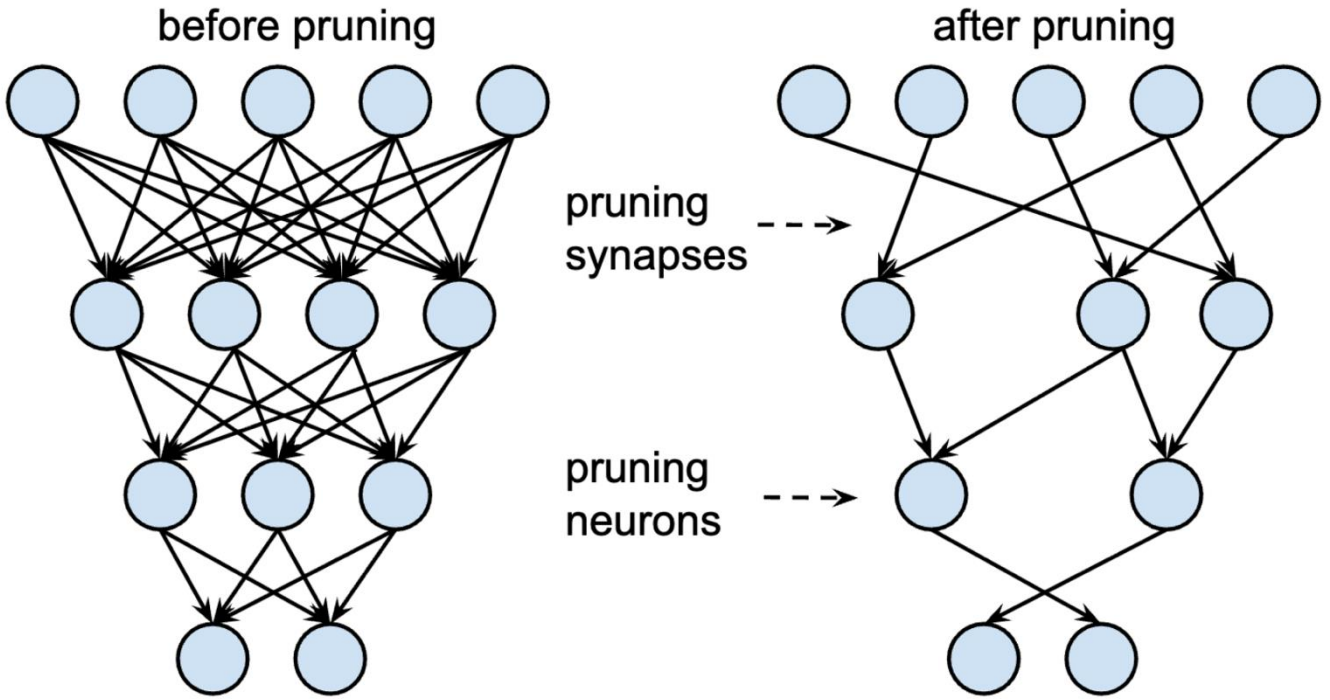
- Transformers with multiple specialized “expert” adapters
- A gating network selects experts per input
- Only only a subset of experts are activated per token/input

Lecture Plan

- Parameter-Efficient Fine-Tuning
  - Prompt Tuning, Prefix Tuning, Adapter
  - Low-Rank Adaptation (LoRA)
- Efficient Architecture
  - Mixture of Experts (MoE)
- Model Compression
  - Pruning, Quantization
  - Distillation
- Inference
  - KV Cache

Pruning

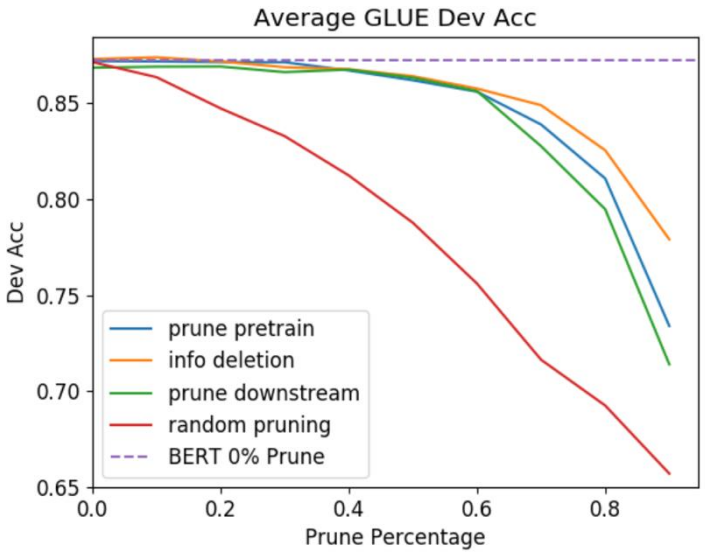
- Reduces model size and computation
- Keeps performance with fewer weights



Learning both Weights and Connections for Efficient Neural Networks, 2015

Transformer with Pruning

- Prune model weights
- Prune neurons
- Prune attention head
- Prune layers



Drop.	SST-2	MNLI	QNLI	QQP	STS-B	RTE	MRPC
				BERT			
0/12	92.43	84.04	91.12	91.07	88.79	67.87	87.99
2/12	92.20 (0.23↓)	83.26 (0.78↓)	89.84 (1.28↓)	90.92 (0.15↓)	88.70 (0.09↓)	62.82 (5.05↓)	86.27 (1.72↓)
4/12	90.60 (1.83↓)	82.51 (1.53↓)	89.68 (1.44↓)	90.63 (0.44↓)	88.64 (0.15↓)	67.87 (0.00)	79.41 (8.58↓)
6/12	90.25 (2.18↓)	81.13 (2.91↓)	87.63 (3.49↓)	90.35 (0.72↓)	88.45 (0.34↓)	64.98 (2.89↓)	80.15 (7.84↓)
				RoBERTa			
0/12	92.20	86.44	91.73	90.48	89.87	68.95	88.48
2/12	93.46 (1.26↑)	86.53 (0.09↑)	91.23 (0.50↓)	91.02 (0.54↑)	90.21 (0.34↑)	71.84 (2.89↑)	89.71 (1.23↑)
4/12	93.00 (0.80↑)	86.20 (0.24↓)	90.57 (1.16↓)	91.12 (0.64↑)	89.77 (0.10↓)	70.40 (1.45↑)	87.50 (0.98↓)
6/12	91.97 (0.23↓)	84.44 (2.00↓)	90.00 (1.73↓)	90.91 (0.43↑)	88.92 (0.95↓)	64.62 (4.33↓)	85.78 (2.70↓)

Compressing BERT: Studying the Effects of Weight Pruning on Transfer Learning, 2020

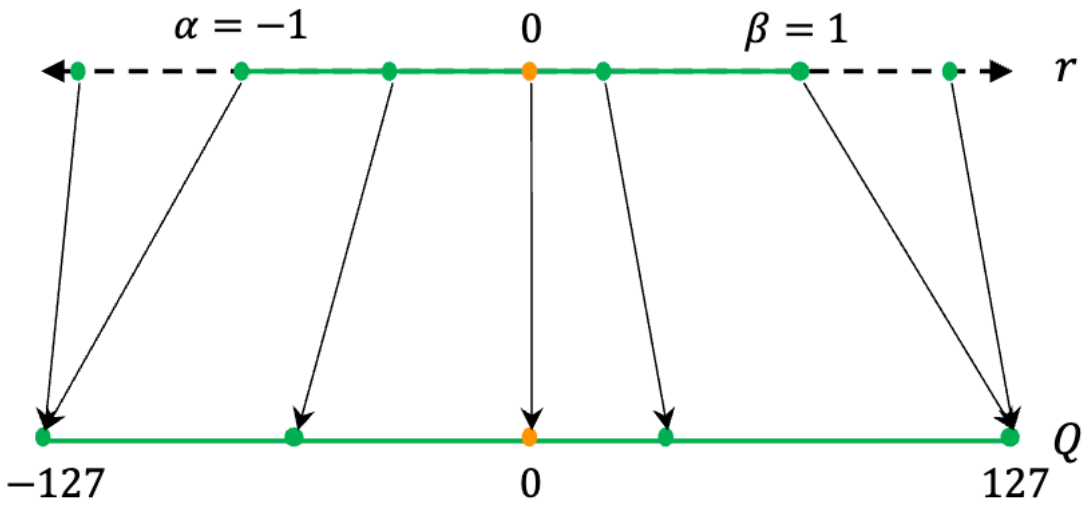
On the Effect of Dropping Layers of Pre-trained Transformer Models, 2022

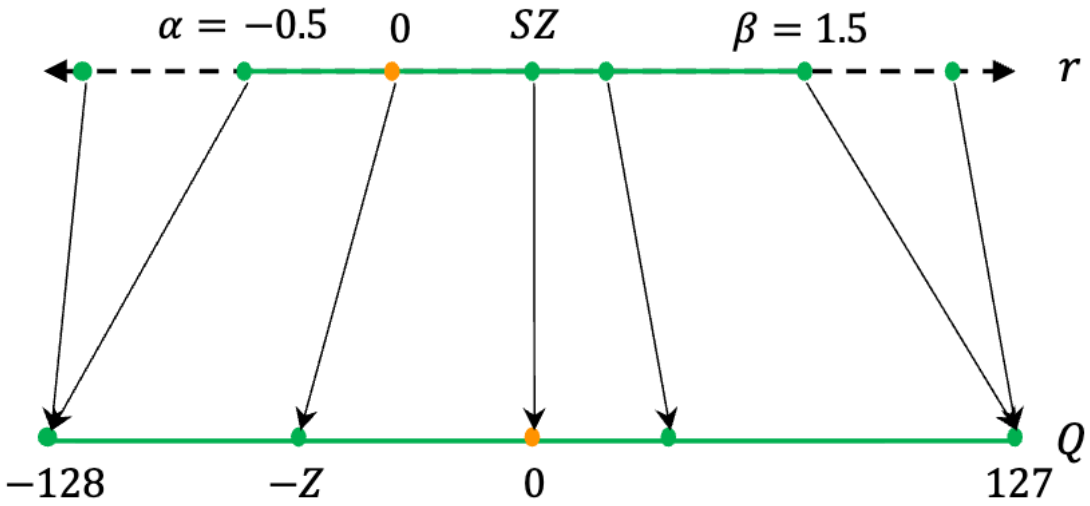
Quantization

- Reduce numerical precision of model weights
- Converts high-precision values (fp32) → lower precision (int8, fp16, etc.)
- Reduces memory and computation requirements
- Model trained with fp32 → Quantization → Inference with int8

Quantization

- 123.45 → 123
- 10.789654 → 10





A Survey of Quantization Methods for Efficient Neural Network Inference, 2021

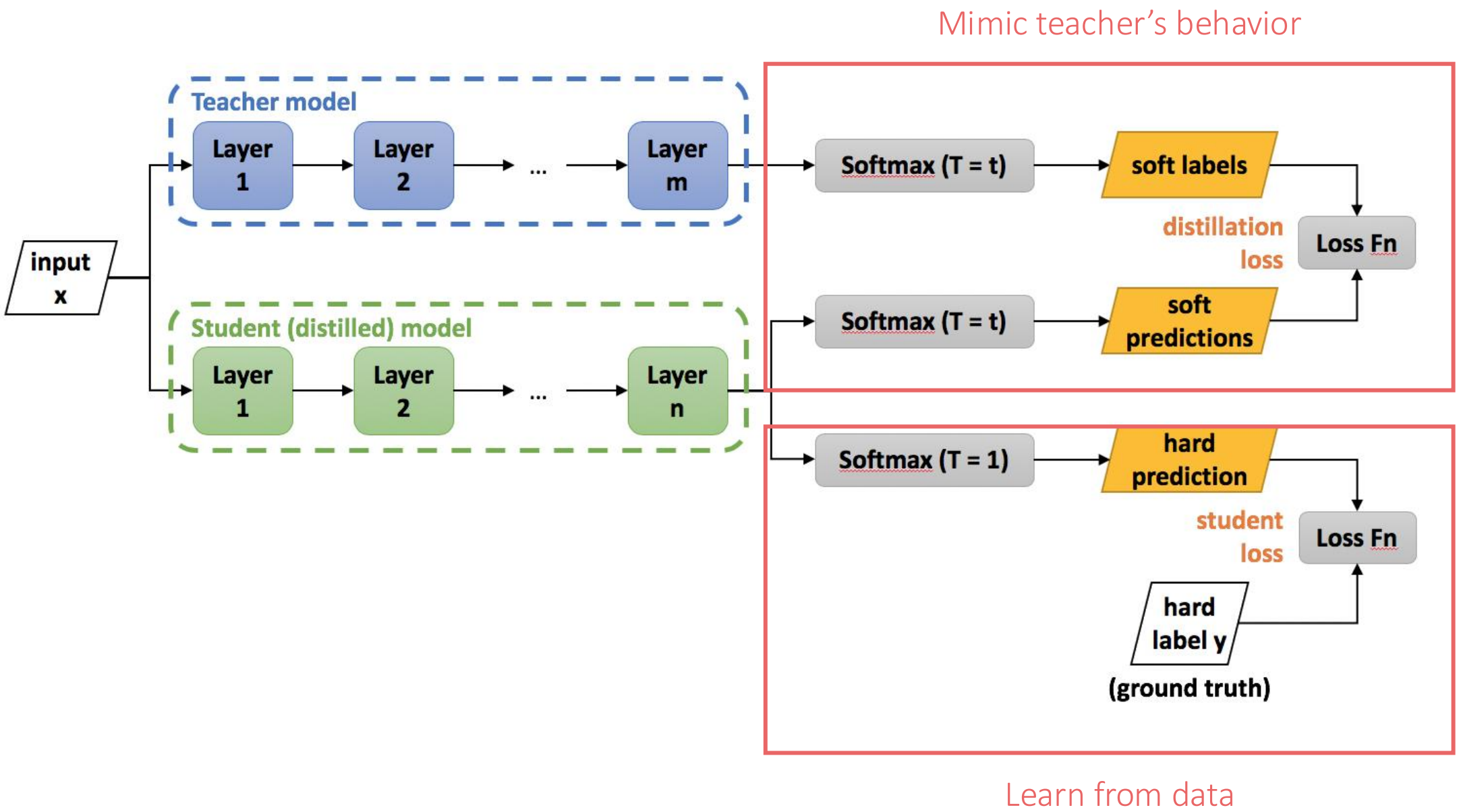
Quantization

Model	Method	W/A	Storage (GB)	OpenLLM Leaderboard-v1 ↑						
				ARC-c (25-shot) acc_norm	GSM8k (5-shot) acc	HellaSwag (10-shot) acc_norm	MMLU (5-shot) acc	TruthfulQA (0-shot) mc2	Winogrande (5-shot) acc	Avg.
Llama-3.1-8B-it	FP16	16 / 16	16	60.24	76.65	80.21	68.10	54.03	76.16	69.23
	FP8	8 / 8	8	61.52 (↑1.28)	74.75 (↓1.90)	80.12 (↓0.09)	68.52 (↑0.42)	53.81 (↓0.22)	77.43 (↑1.27)	69.36 (↑0.13)
	GPTQ*	4 / 16	4	61.43 (↑1.19)	72.33 (↓4.32)	78.36 (↓1.85)	66.85 (↓1.25)	53.60 (↓0.43)	75.22 (↓0.94)	67.97 (↓1.26)
	GPTQ**	4 / 16	4	59.81 (↓0.43)	69.98 (↓6.67)	78.53 (↓1.68)	66.07 (↓2.03)	50.45 (↓3.58)	76.64 (↑0.48)	66.91 (↓2.32)
	GPTQ**	8 / 16	8	61.01 (↑0.77)	75.81 (↓0.84)	80.27 (↓0.06)	68.21 (↑0.11)	54.03 (0.00)	77.19 (↑1.03)	69.42 (↑0.19)
	SmoothQuant	8 / 8	8	60.75 (↑0.51)	76.12 (↓0.53)	80.08 (↓0.13)	68.22 (↑0.12)	53.85 (↓0.18)	77.11 (↑0.95)	69.36 (↑0.13)
	AWQ	4 / 16	4	58.53 (↓1.71)	73.39 (↓3.26)	79.10 (↓1.11)	66.26 (↓1.84)	51.87 (↓2.16)	75.37 (↓0.79)	67.42 (↓1.81)
Llama-3.1-70B-it	FP16	16 / 16	140	69.54	88.70	86.74	82.30	59.85	85.40	78.76
	FP8	8 / 8	70	69.45 (↓0.09)	88.25 (↓0.45)	86.69 (↓0.05)	82.02 (↓0.28)	59.80 (↓0.05)	85.08 (↓0.32)	78.55 (↓0.21)
	GPTQ*	4 / 16	35	69.80 (↑0.26)	89.54 (↑0.84)	86.28 (↓0.46)	81.40 (↓0.90)	59.37 (↓0.48)	84.69 (↓0.71)	78.51 (↓0.25)
	GPTQ**	4 / 16	35	69.97 (↑0.43)	89.76 (↑1.06)	86.26 (↓0.48)	81.97 (↓0.33)	58.74 (↓1.11)	84.53 (↓0.87)	78.54 (↓0.22)
	GPTQ**	8 / 16	70	69.03 (↓0.51)	87.95 (↓0.75)	86.29 (↓0.45)	82.17 (↓0.13)	58.94 (↓0.91)	84.53 (↓0.87)	78.15 (↓0.61)
	SmoothQuant	8 / 8	70	70.05 (↑0.51)	88.55 (↓0.15)	86.56 (↓0.18)	82.10 (↓0.20)	60.39 (↑0.54)	85.24 (↓0.16)	78.82 (↑0.06)
	AWQ	4 / 16	35	69.80 (↑0.26)	90.83 (↑2.13)	86.18 (↓0.56)	81.33 (↓0.97)	59.68 (↓0.17)	84.37 (↓1.03)	78.70 (↓0.06)
Llama-3.1-405B-it	FP16	16 / 16	810	73.72	94.84	88.40	83.98	65.42	85.00	81.89
	FP8	8 / 8	405	73.12 (↓0.60)	95.38 (↑0.54)	88.32 (↓0.08)	85.91 (↑1.93)	64.79 (↓0.63)	85.63 (↑0.63)	82.19 (↑0.30)
	GPTQ**	4 / 16	202.5	72.10 (↓1.62)	94.24 (↓0.60)	88.17 (↓0.23)	85.79 (↑1.81)	64.80 (↓0.62)	85.48 (↑0.48)	81.76 (↓0.13)
	SmoothQuant	8 / 8	405	72.01 (↓1.71)	92.72 (↓2.12)	87.53 (↓0.87)	73.28 (↓10.70)	65.19 (↓0.23)	85.95 (↑0.95)	79.45 (↓2.44)
	AWQ	4 / 16	202.5	73.98 (↑0.26)	94.84 (0.00)	88.04 (↓0.36)	85.71 (↑1.73)	64.25 (↓1.17)	86.35 (↑1.35)	82.20 (↑0.31)

Model Distillation

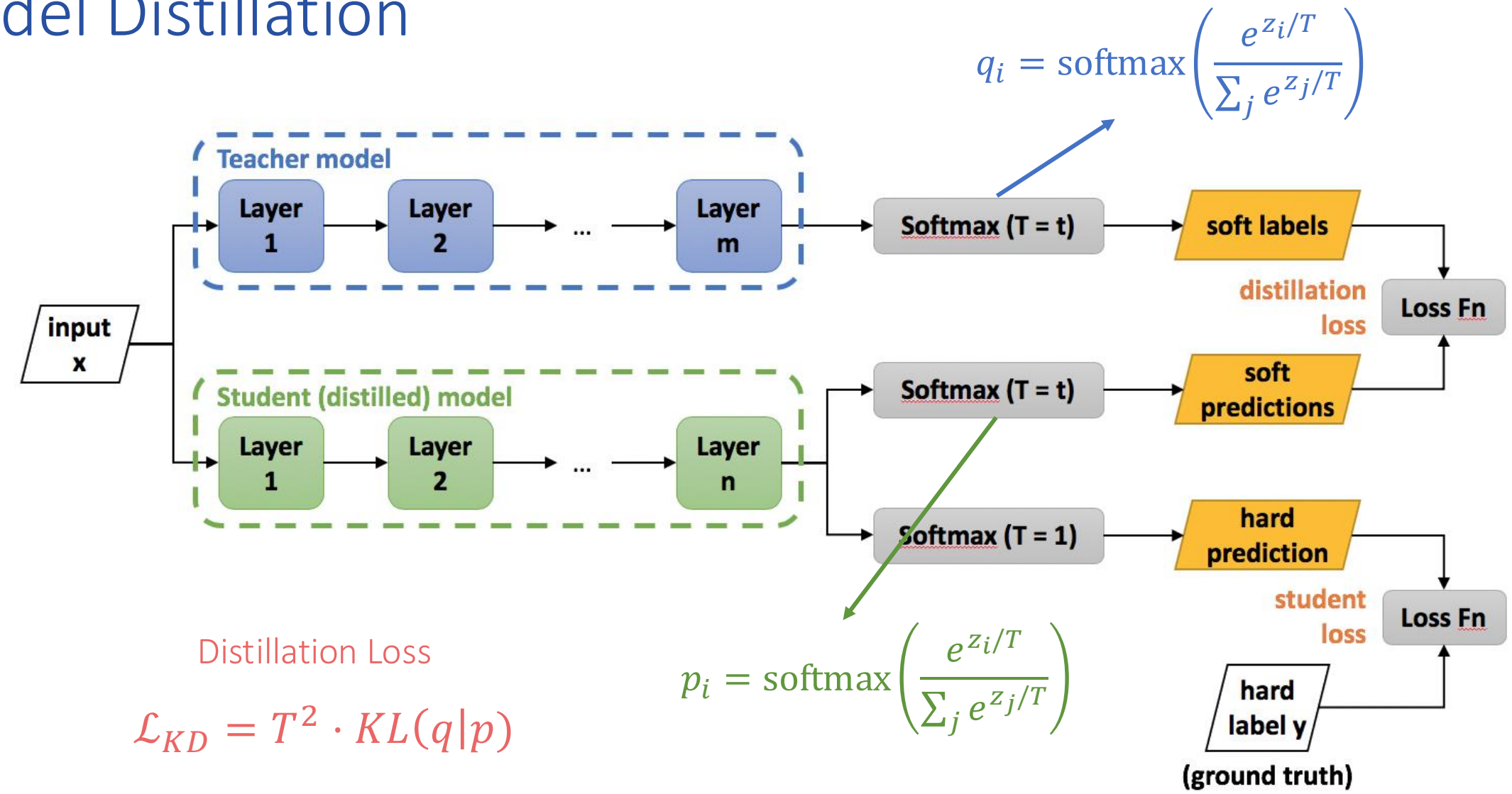
- Distill knowledge from a large model to a small model while maintaining similar capability
  - **Large model:** teacher model
  - **Small model:** student model
- Train a student model to mimic the behavior of the teacher model
- Reduce the number of parameters

Model Distillation



https://onedreame.github.io/2020/12/10/%E6%A8%A1%E5%9E%8B%E8%92%B8%E9%A6%8F%E6%8E%A2%E7%B4%A2(Bert)/

Model Distillation

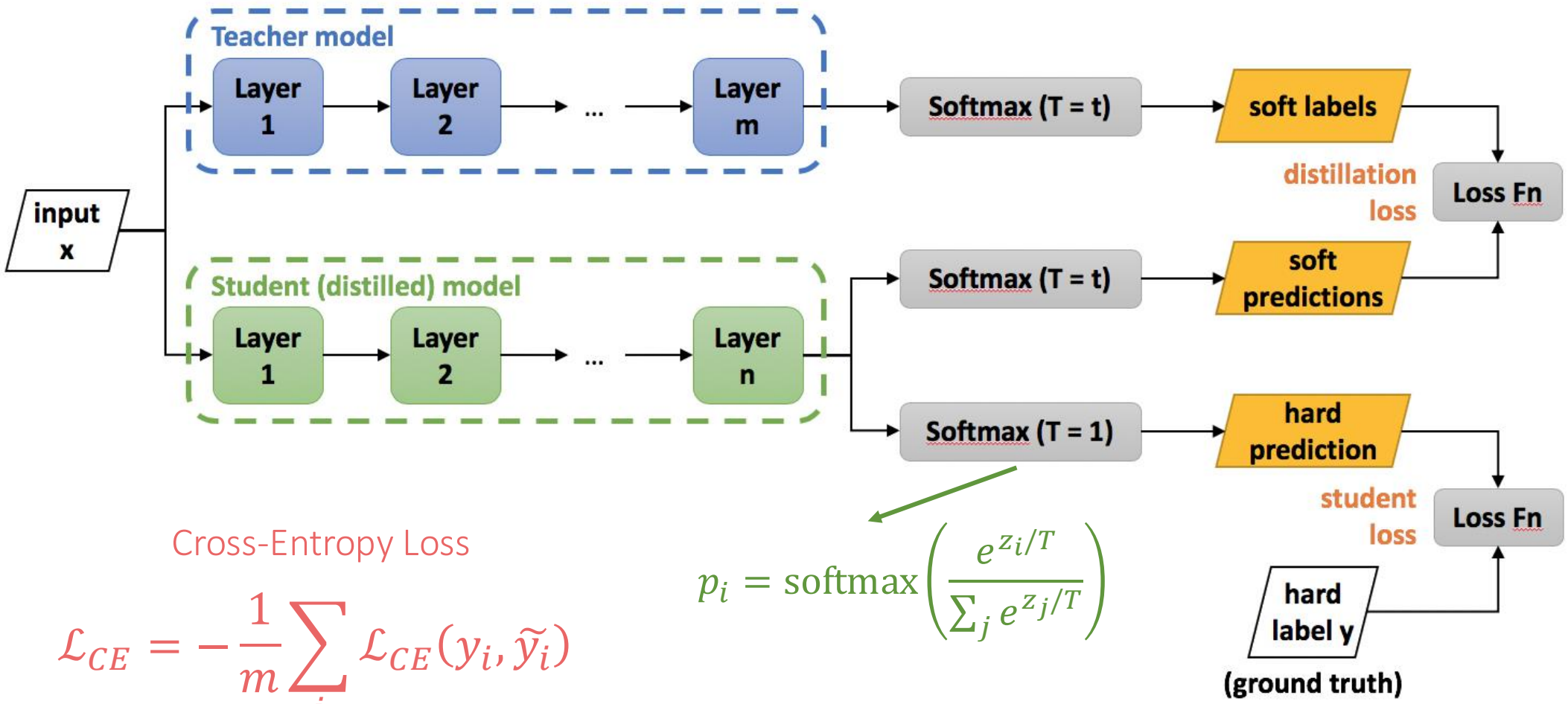


Distillation Loss

$$\mathcal{L}_{KD} = T^2 \cdot KL(q|p)$$

https://onedreame.github.io/2020/12/10/%E6%A8%A1%E5%9E%8B%E8%92%B8%E9%A6%8F%E6%8E%A2%E7%B4%A2(Bert)/

Model Distillation



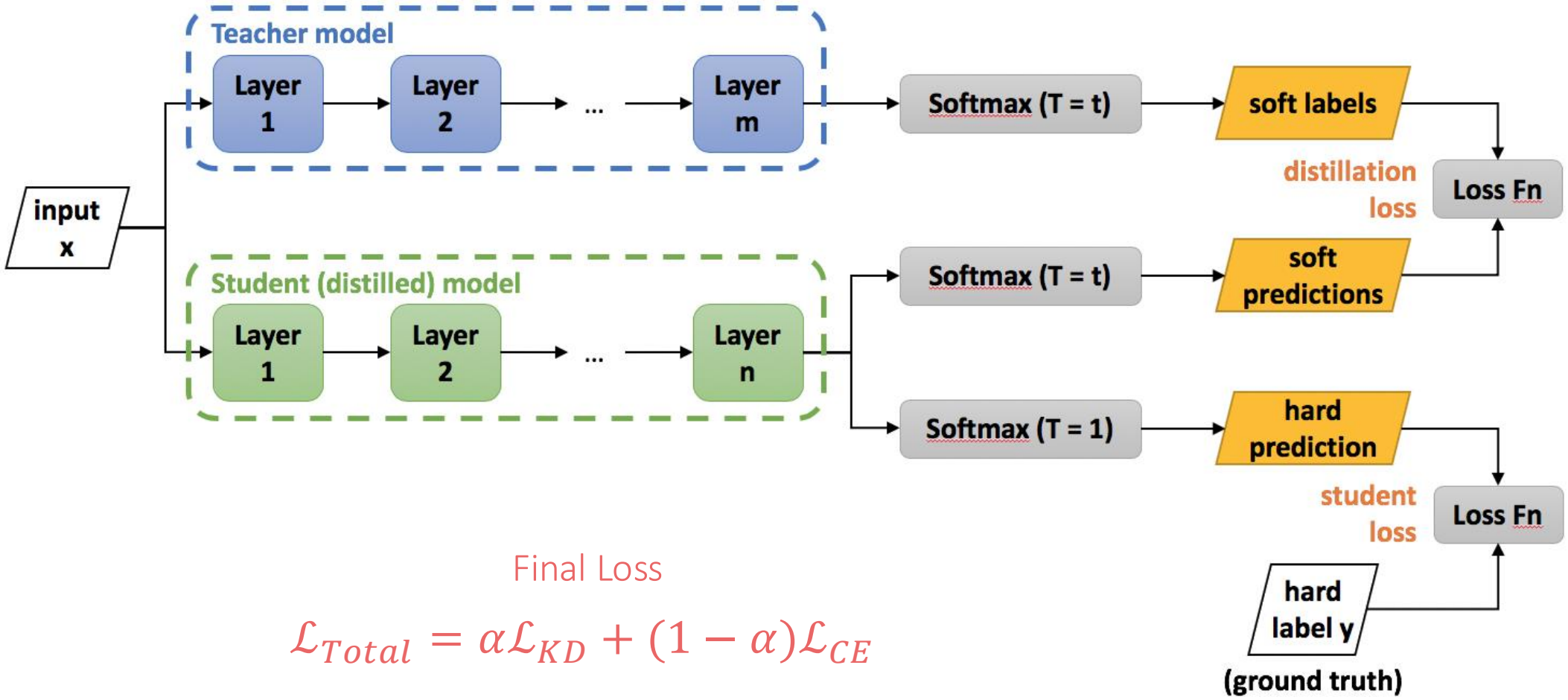
Cross-Entropy Loss

$$\mathcal{L}_{CE} = -\frac{1}{m}\sum_{i}\mathcal{L}_{CE}(y_i, \tilde{y}_i)$$

$$p_i = \text{softmax}\left(\frac{e^{z_i/T}}{\sum_j e^{z_j/T}}\right)$$

https://onedreame.github.io/2020/12/10/%E6%A8%A1%E5%9E%8B%E8%92%B8%E9%A6%8F%E6%8E%A2%E7%B4%A2(Bert)/

Model Distillation



Final Loss

$$\mathcal{L}_{Total} = \alpha \mathcal{L}_{KD} + (1 - \alpha)\mathcal{L}_{CE}$$

https://onedreame.github.io/2020/12/10/%E6%A8%A1%E5%9E%8B%E8%92%B8%E9%A6%8F%E6%8E%A2%E7%B4%A2(Bert)/

DistilBERT

Smaller Size

Model	# param. (Millions)	Inf. time (seconds)
ELMo	180	895
BERT-base	110	668
DistilBERT	66	410

- BERT-base
  - 12 layers, hidden size = 768, 12 attention heads
- DistilBERT
  - 6 layers, hidden size = 768, 12 attention heads

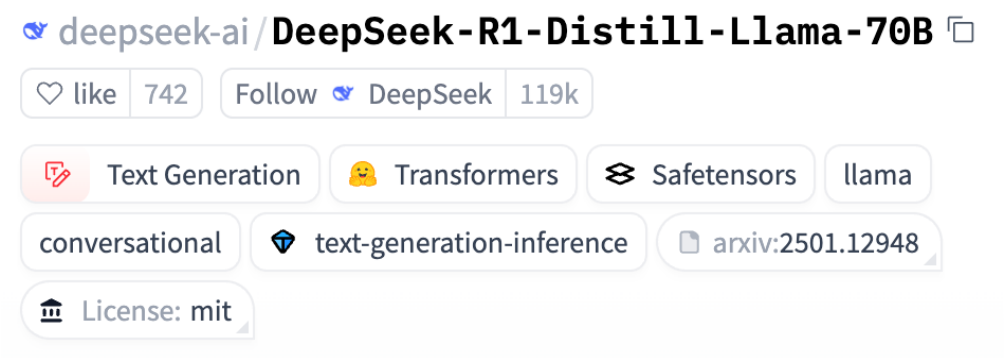
Almost similar performance

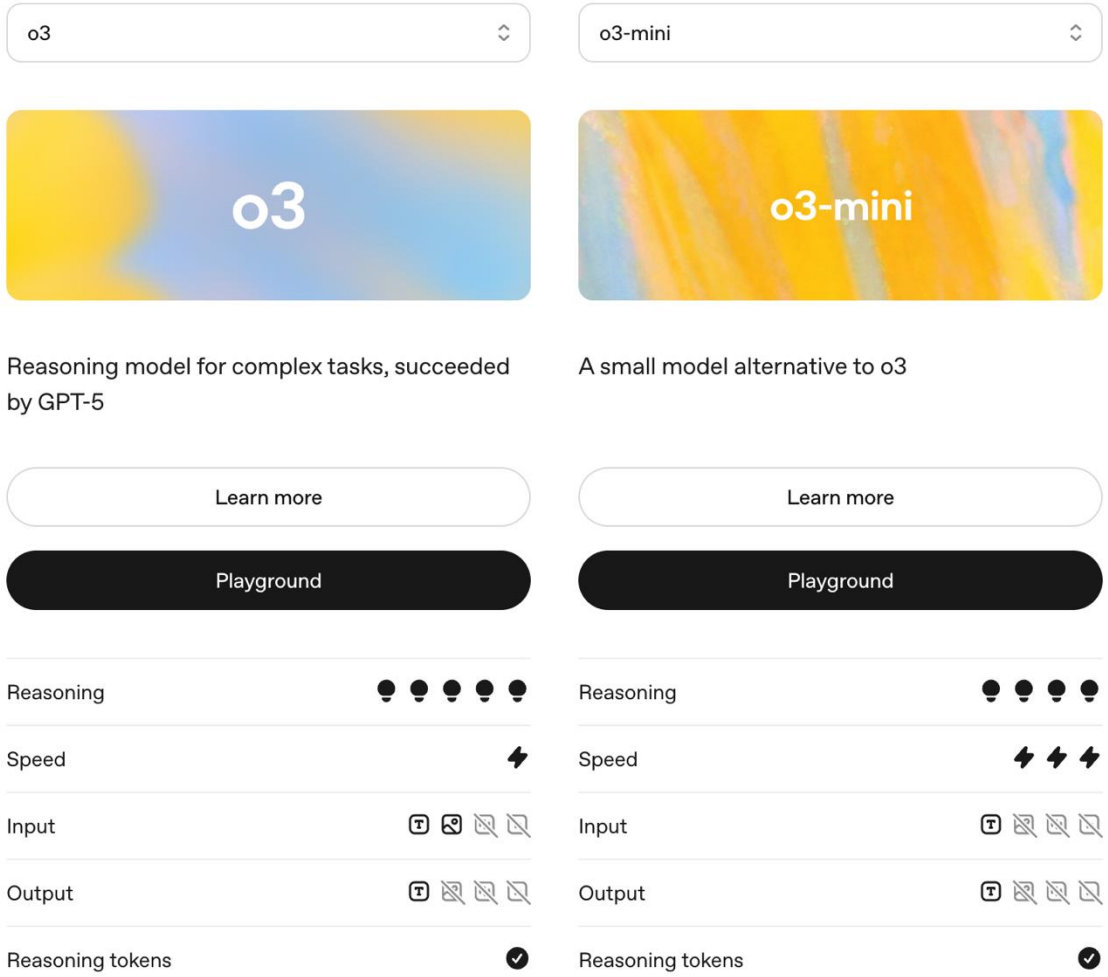
Model	**Score**	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	WNLI
ELMo	68.7	44.1	68.6	76.6	71.1	86.2	53.4	91.5	70.4	56.3
BERT-base	79.5	56.3	86.7	88.6	91.8	89.6	69.3	92.7	89.0	53.5
DistilBERT	77.0	51.3	82.2	87.5	89.2	88.5	59.9	91.3	86.9	56.3

DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, 2020

LLM Distillation

- Smaller and faster
  - Sacrifice a little bit performance
- Distillation → Copy training data



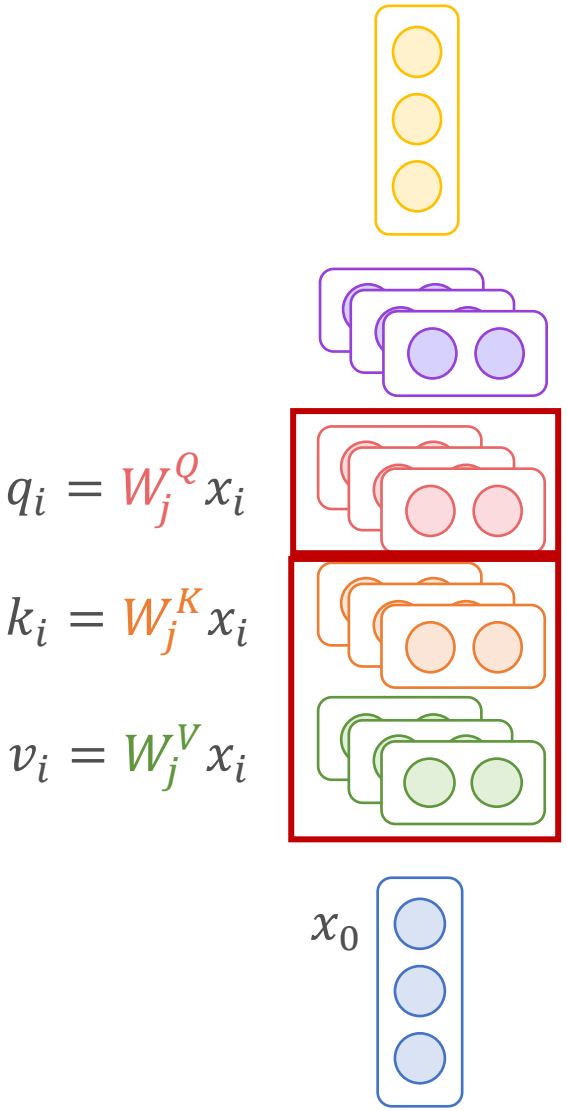


https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Llama-70B
https://developers.openai.com/api/docs/models/compare

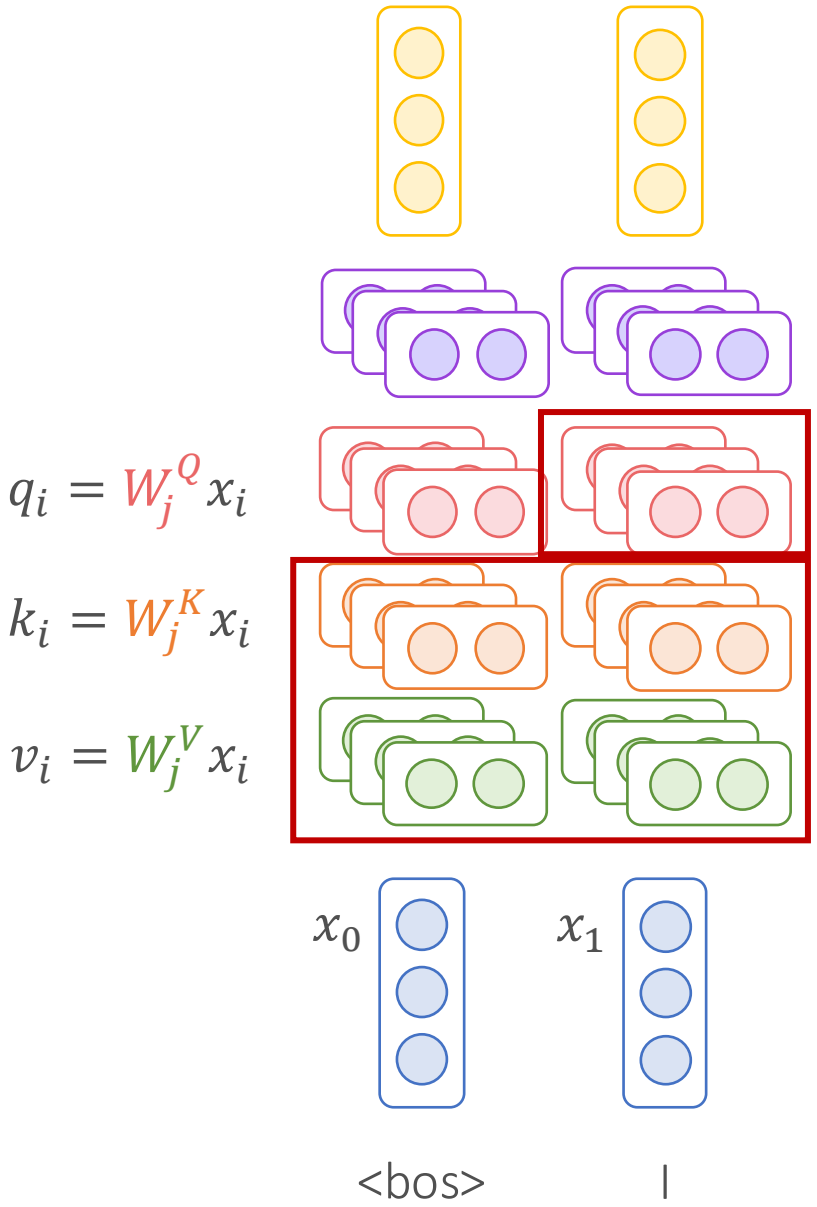
Lecture Plan

- Parameter-Efficient Fine-Tuning
  - Prompt Tuning, Prefix Tuning, Adapter
  - Low-Rank Adaptation (LoRA)
- Efficient Architecture
  - Mixture of Experts (MoE)
- Model Compression
  - Pruning, Quantization
  - Distillation
- Inference
  - KV Cache

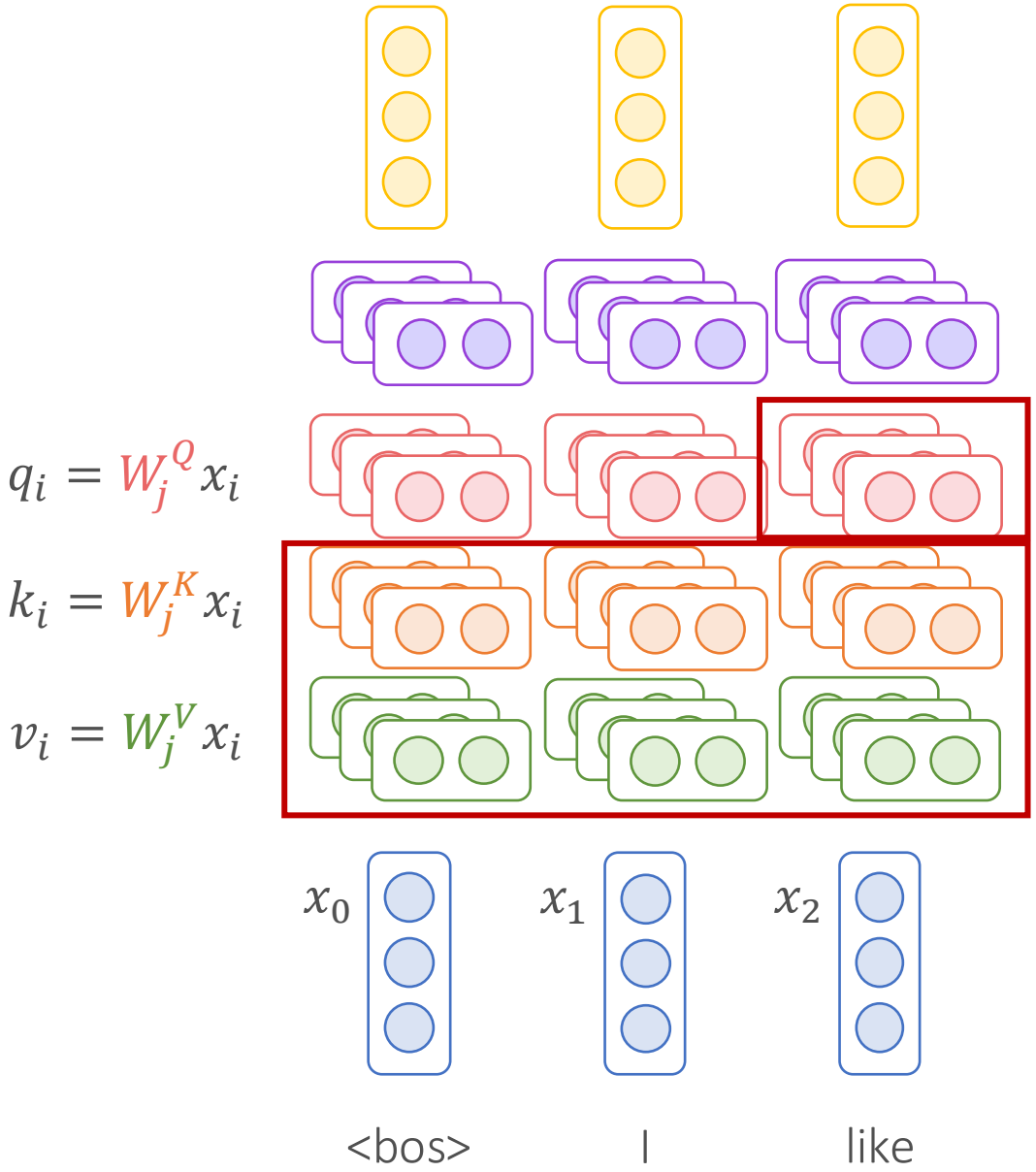
LLM Inference



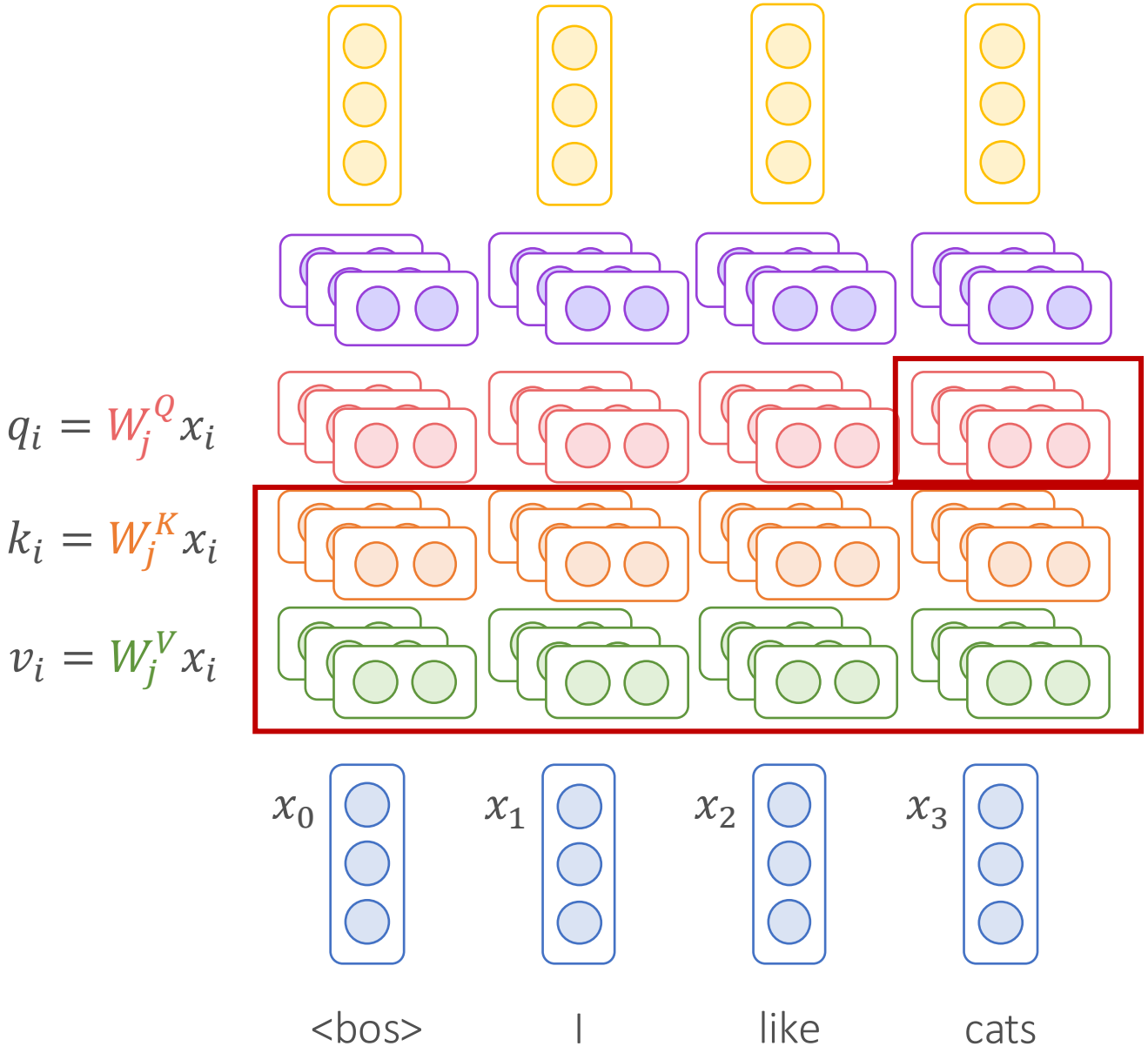
LLM Inference



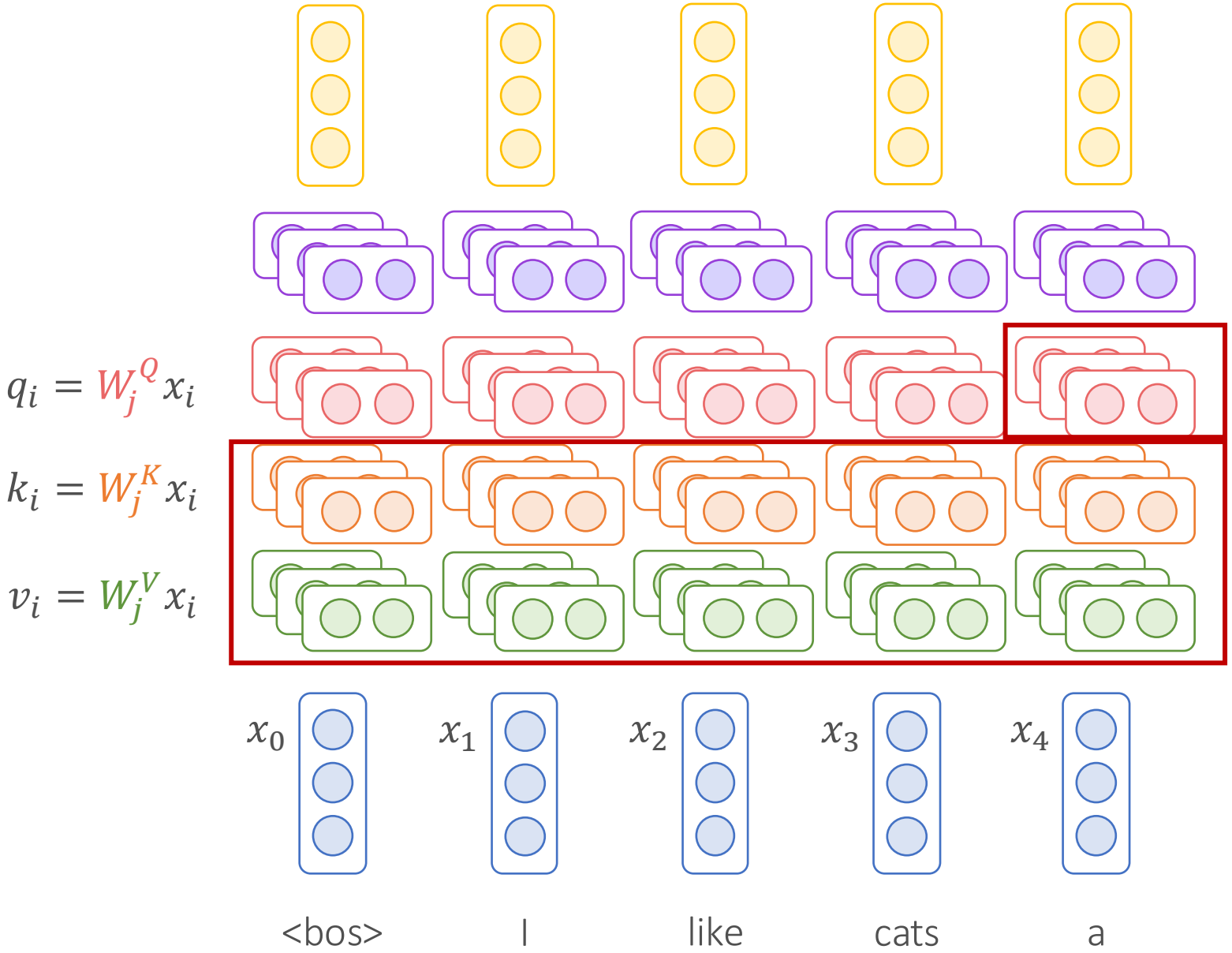
LLM Inference



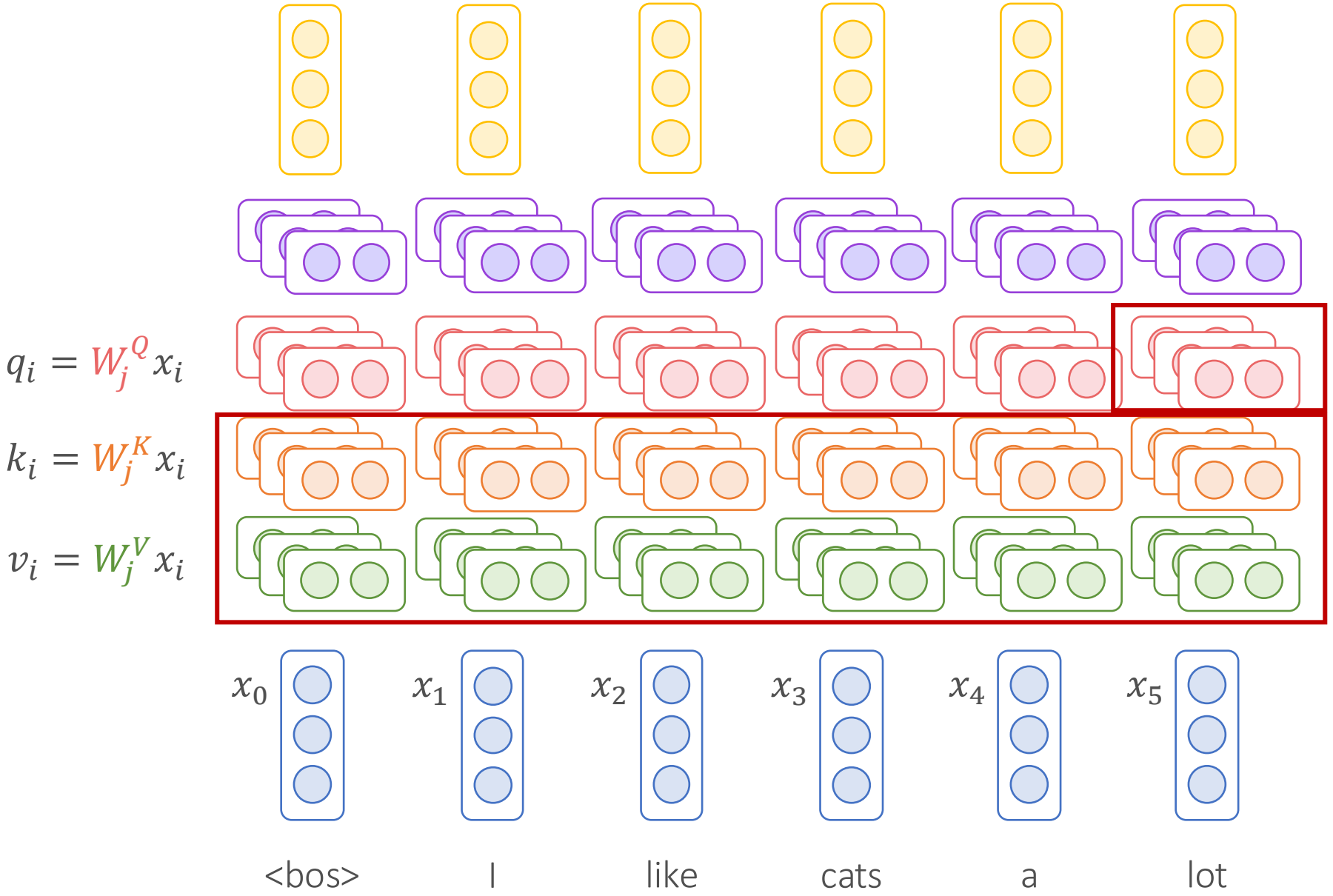
LLM Inference



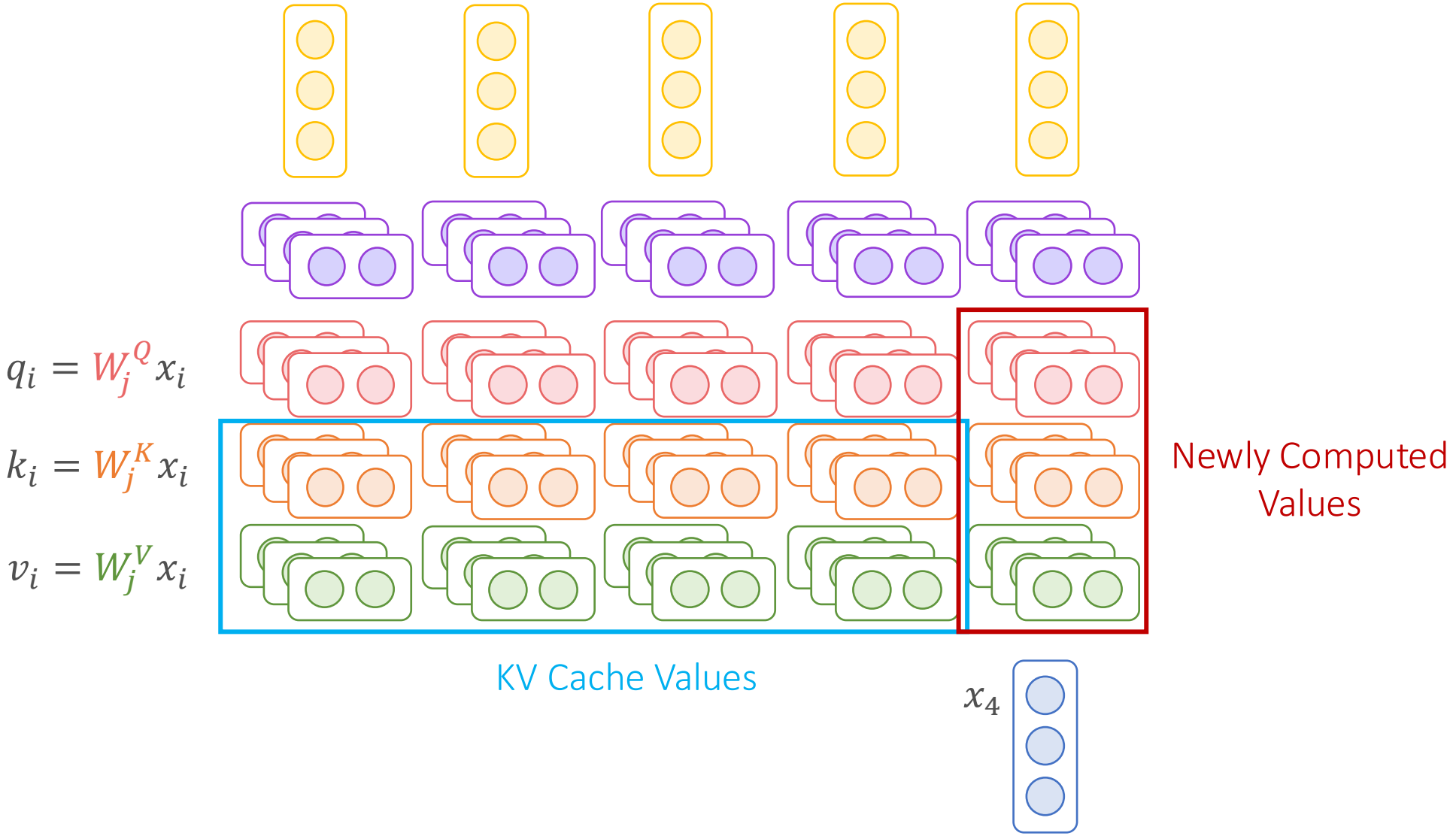
LLM Inference



LLM Inference



KV Cache

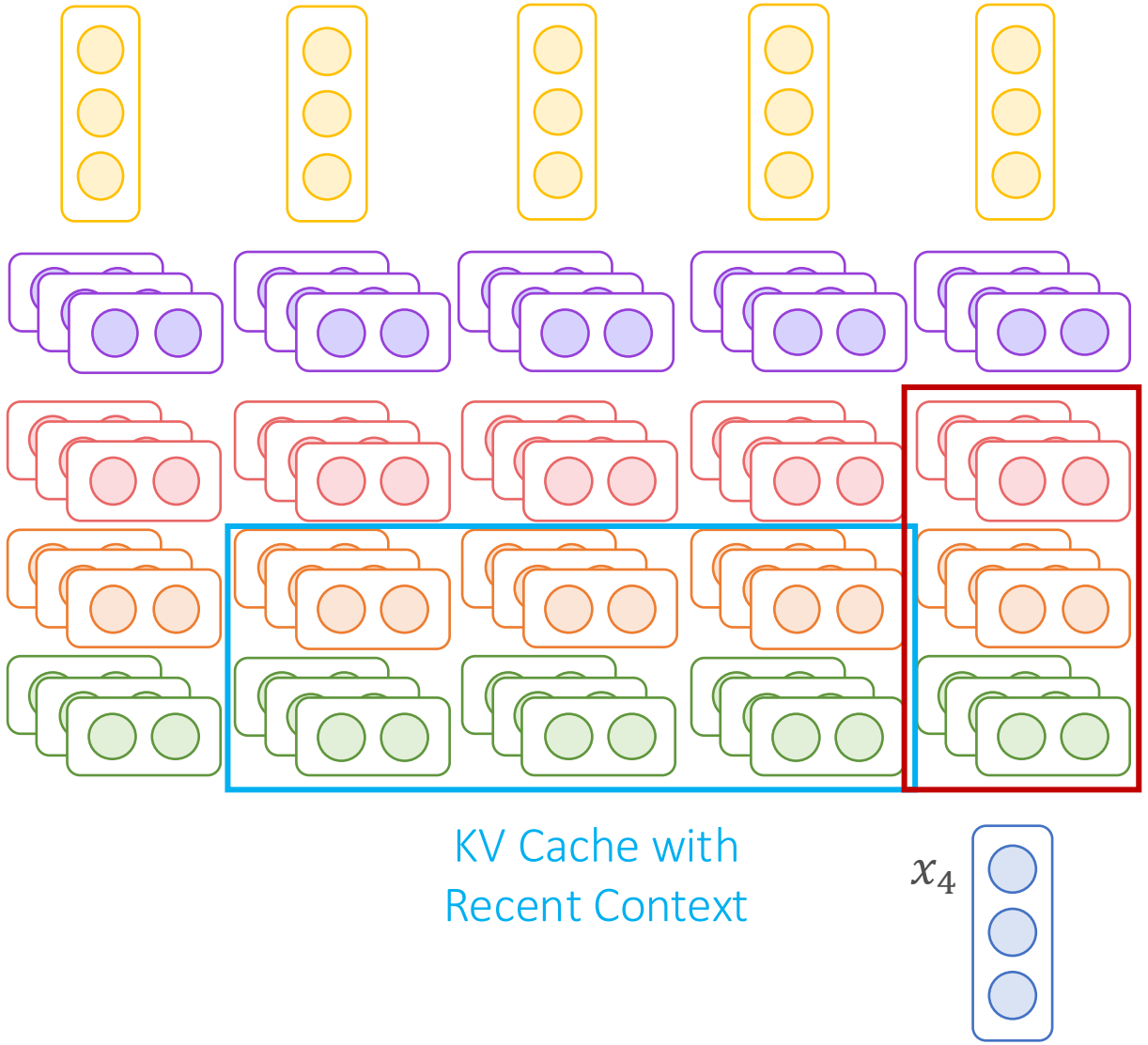

$q_i = W_j^Q x_i$
$k_i = W_j^K x_i$
$v_i = W_j^V x_i$
Newly Computed Values
KV Cache Values
x_4
a

Selective KV Cache

$q_i = W_j^Q x_i$

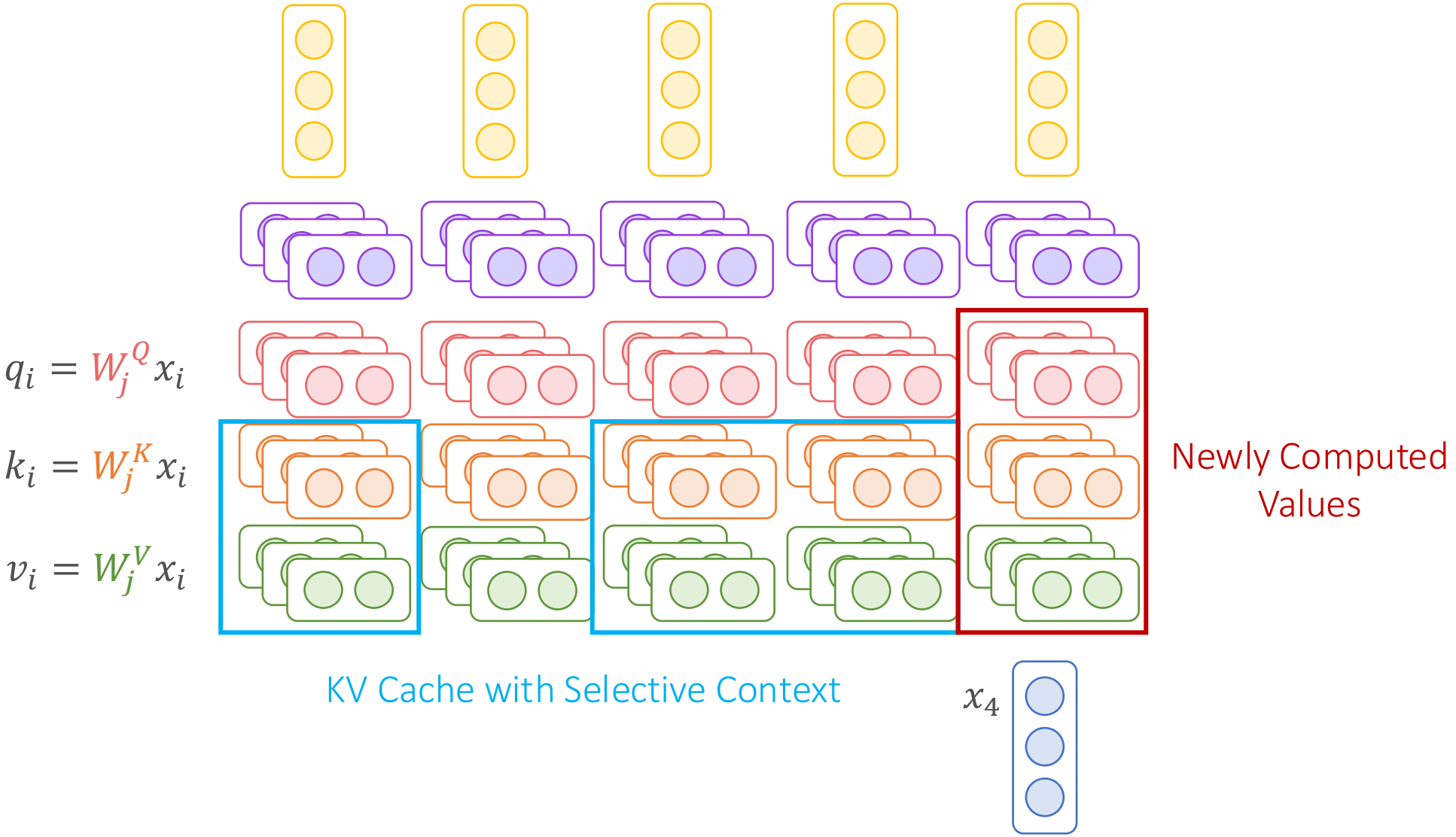
$k_i = W_j^K x_i$

$v_i = W_j^V x_i$



a

Selective KV Cache



a

Lecture Plan

- Parameter-Efficient Fine-Tuning
  - Prompt Tuning, Prefix Tuning, Adapter
  - Low-Rank Adaptation (LoRA)
- Efficient Architecture
  - Mixture of Experts (MoE)
- Model Compression
  - Pruning, Quantization
  - Distillation
- Inference
  - KV Cache