LLaMA-Adapter

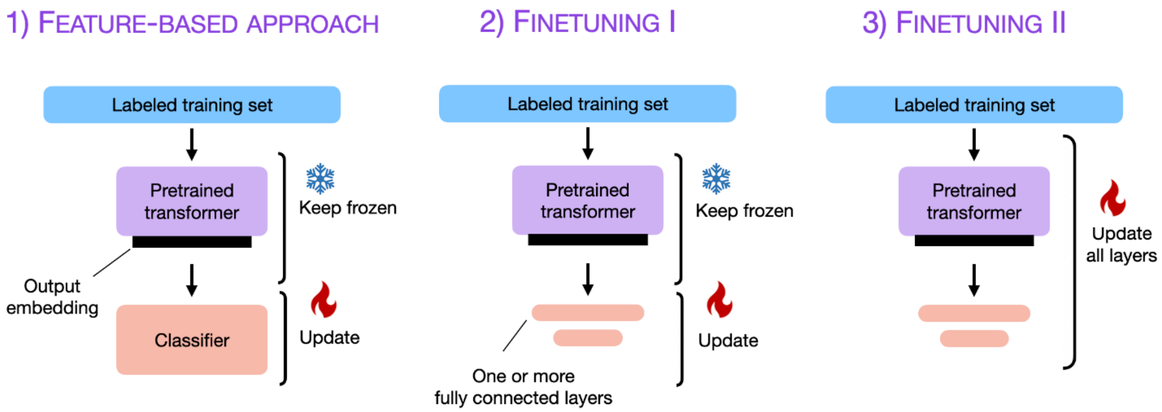
: Efficient Fine-tuning of Language Models with Zero-init Attention (2023.03)

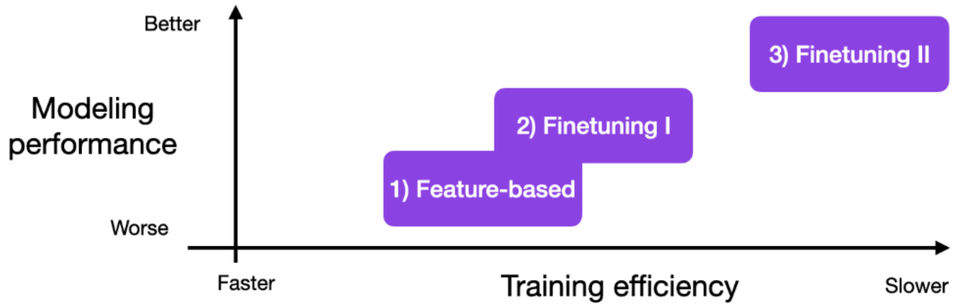
2023.09.13
AI2-1팀 이현경

Background & Introduction

Fine-tuning

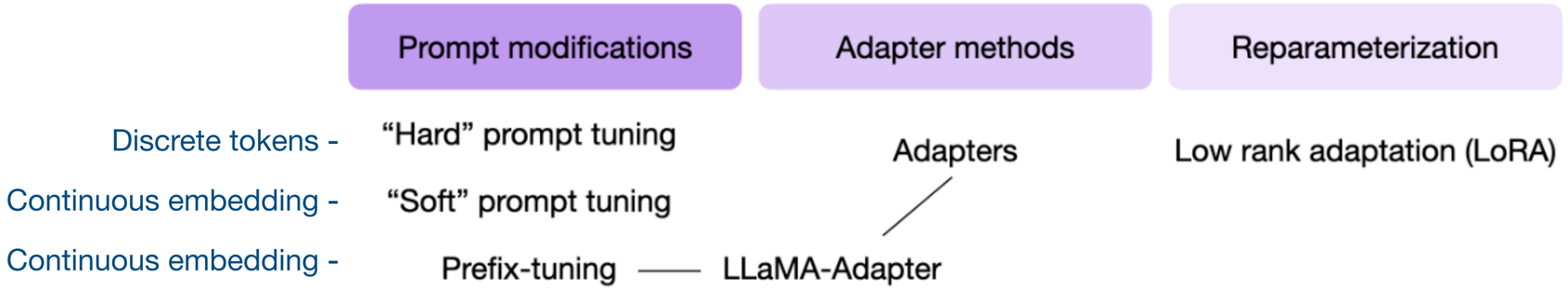
pre-trained LM을 fine-tuning하는 방식





Parameter-Efficient Finetuning

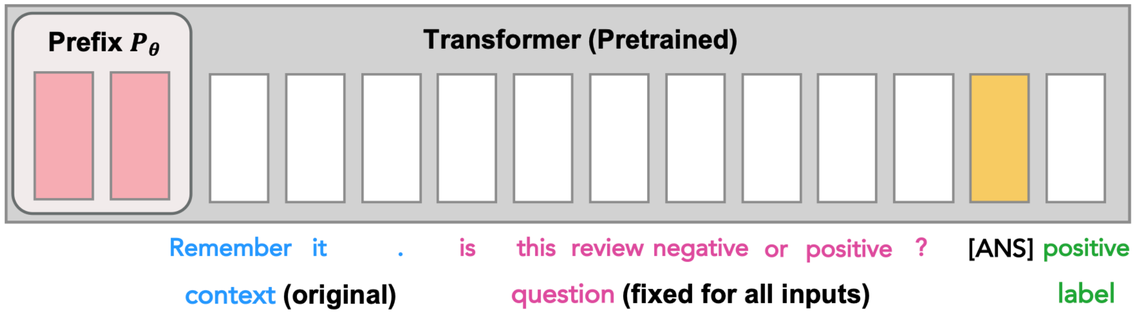
pre-trained LM을 fine-tuning하는 방법

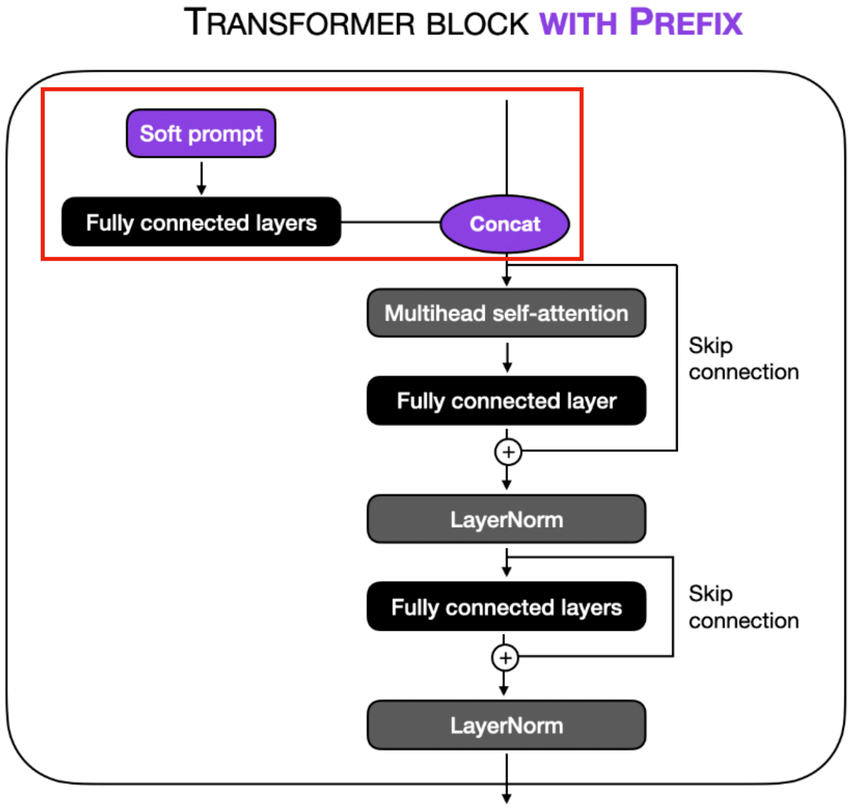


- P-tuning (2021, Tsinghua University,MIT)
- Prompt-tuning (2021, Google Research)
- Prefix-tuning (2021, Stanford)
- Adapter-tuning (2019, Nelil Houslby et al.)
- LoRA (2021, Microsoft)

Parameter-Efficient Finetuning

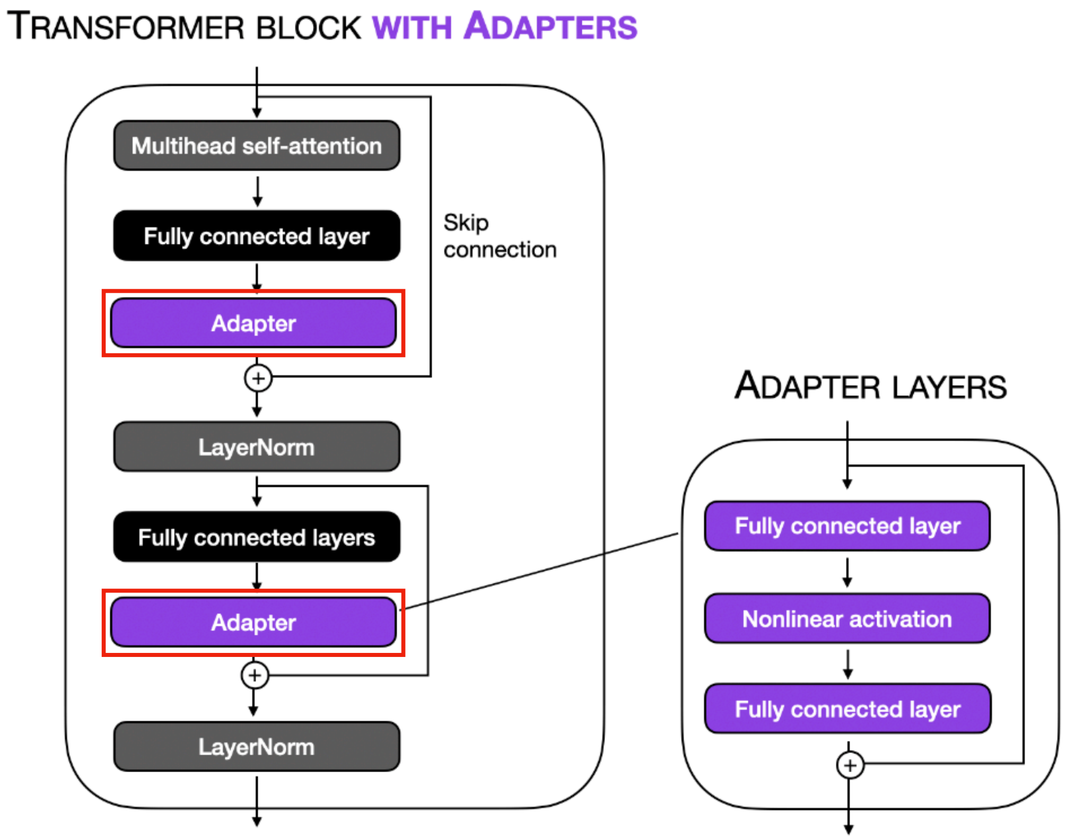
Prefix tuning





Parameter-Efficient Finetuning

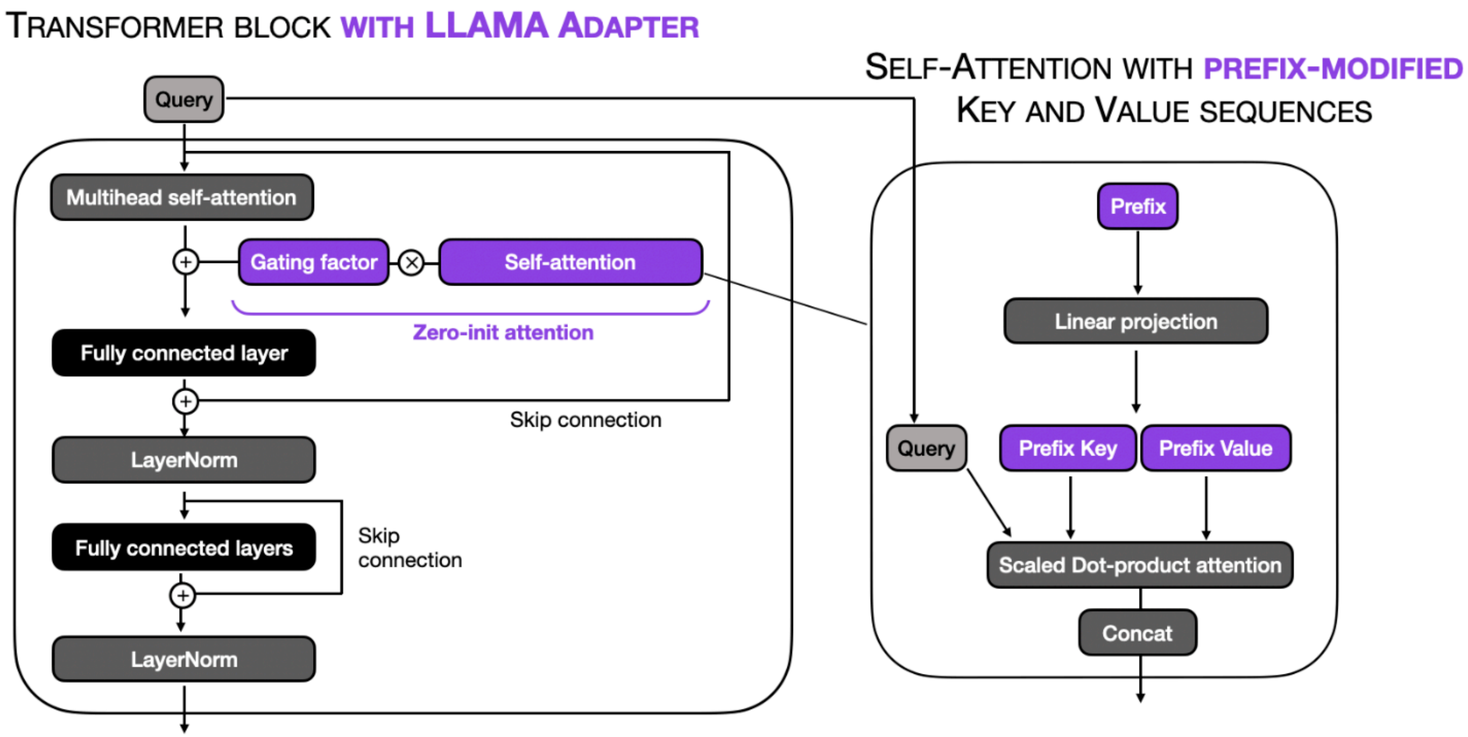
Adapters



Parameter-Efficient Finetuning

LLaMA Adapter

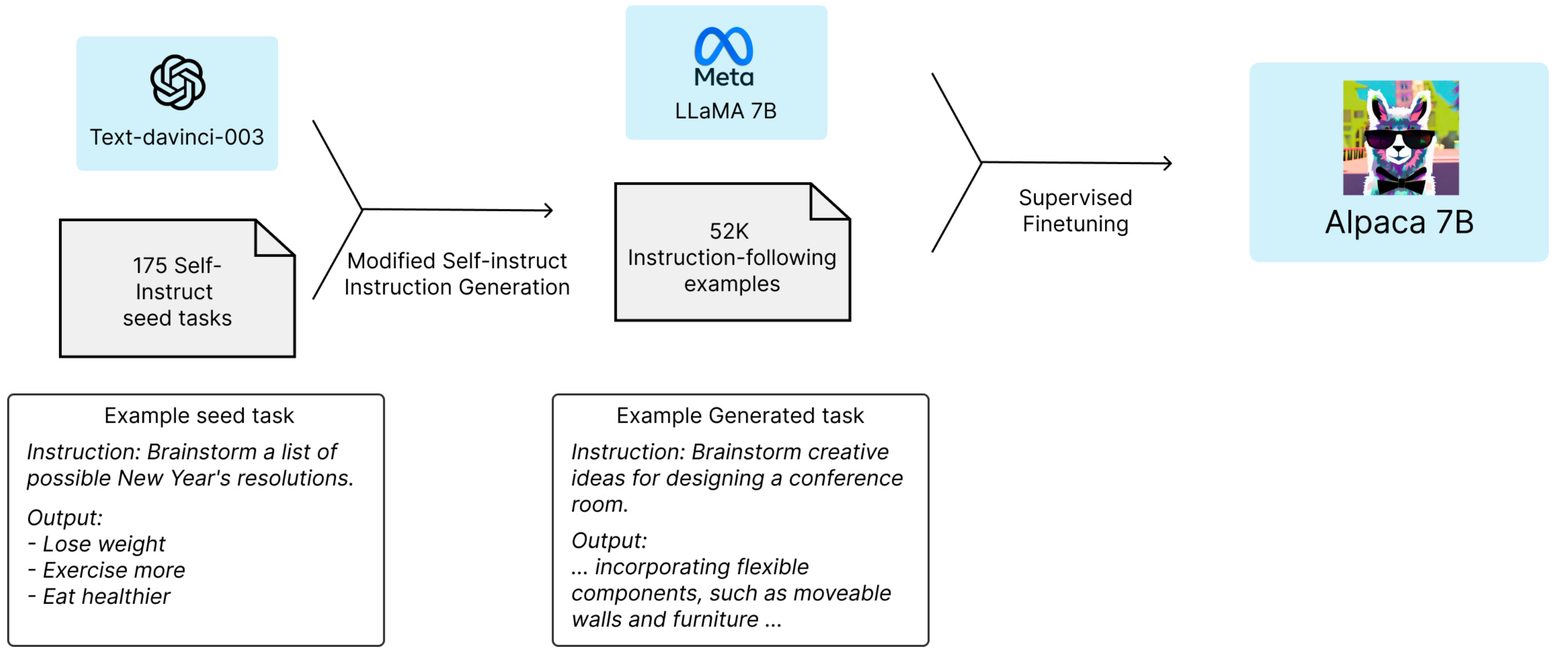
- 상위 L개 레이어에 adaption prompt삽입



Instruction-following model

LLM을 fine-tuning

- Alpaca (Standford)
  - LLaMA 7B fully fine-tune, self-instruction 방식 사용함

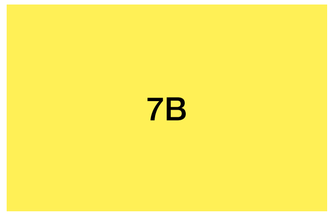


Instruction-following model

모델 간 파라미터 및 성능 비교

GPT-3.5
(text-davinci-003)

<=



Alpaca

Fully
80GB A100 8개로 3시간

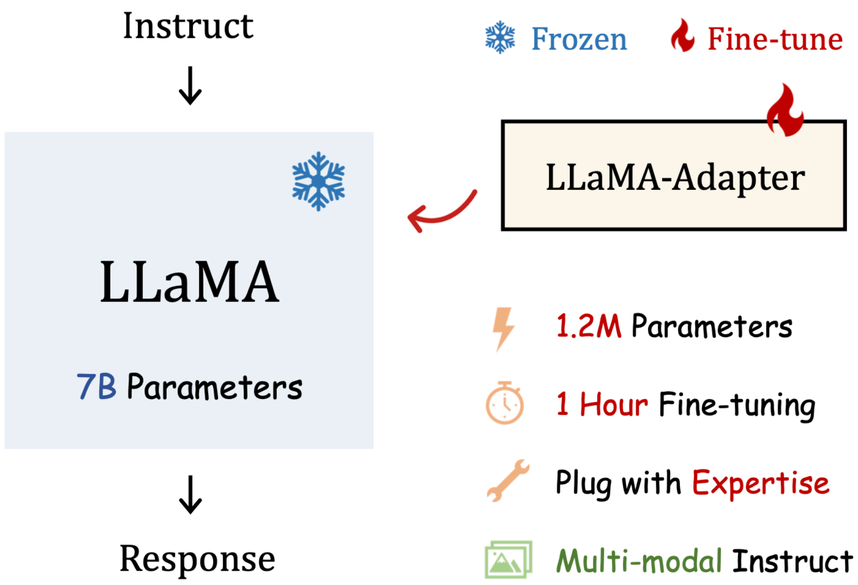
<

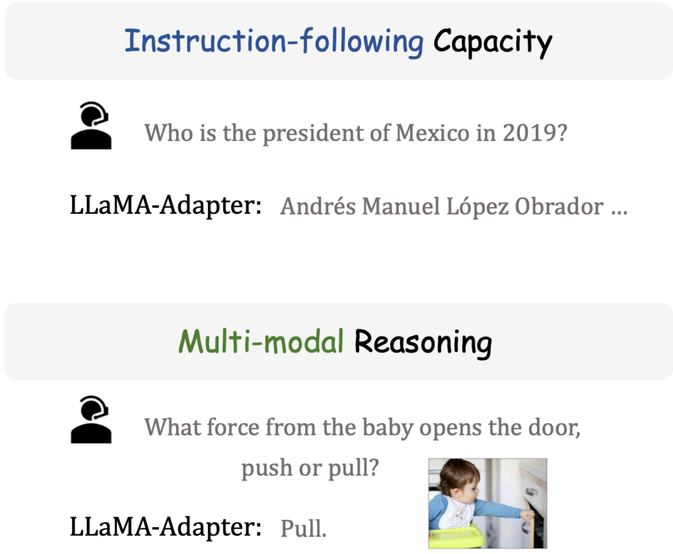
LLaMA
-Adapter

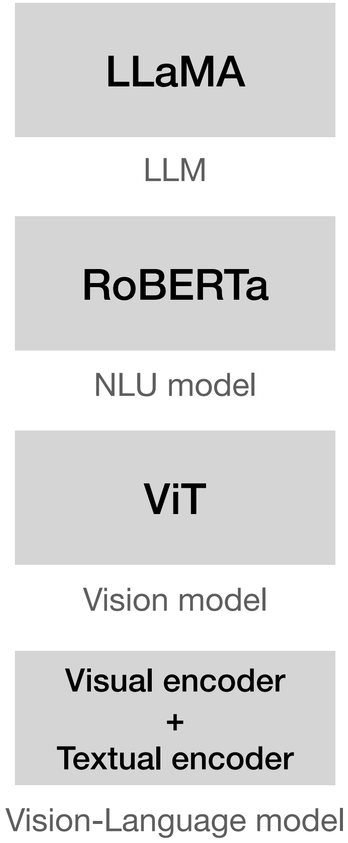
PEFT
80GB A100 8개로 1시간

LLaMA Adapter

Contributions



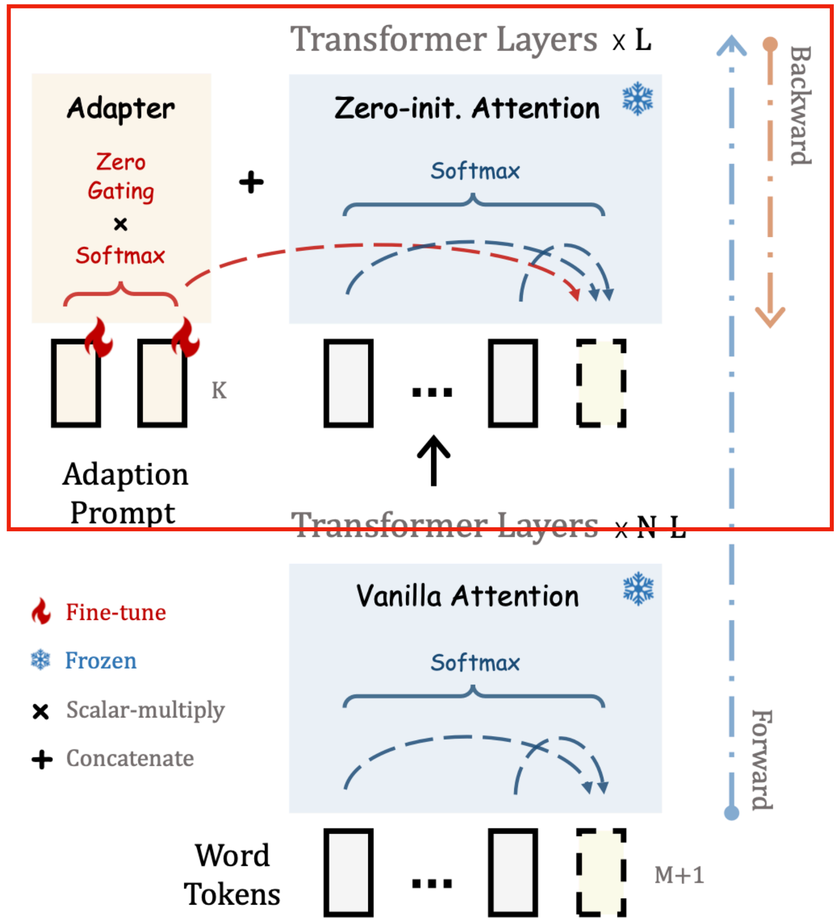




Method

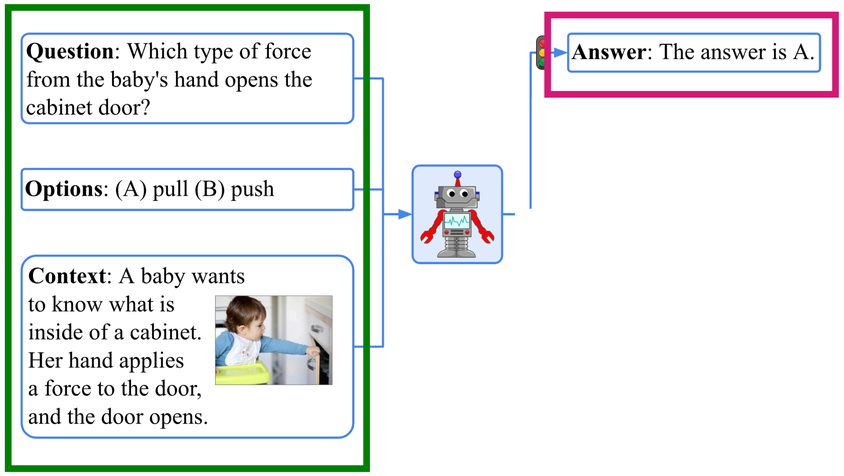
LLaMA-Adapter

: adaption prompts with zero-initialized attention

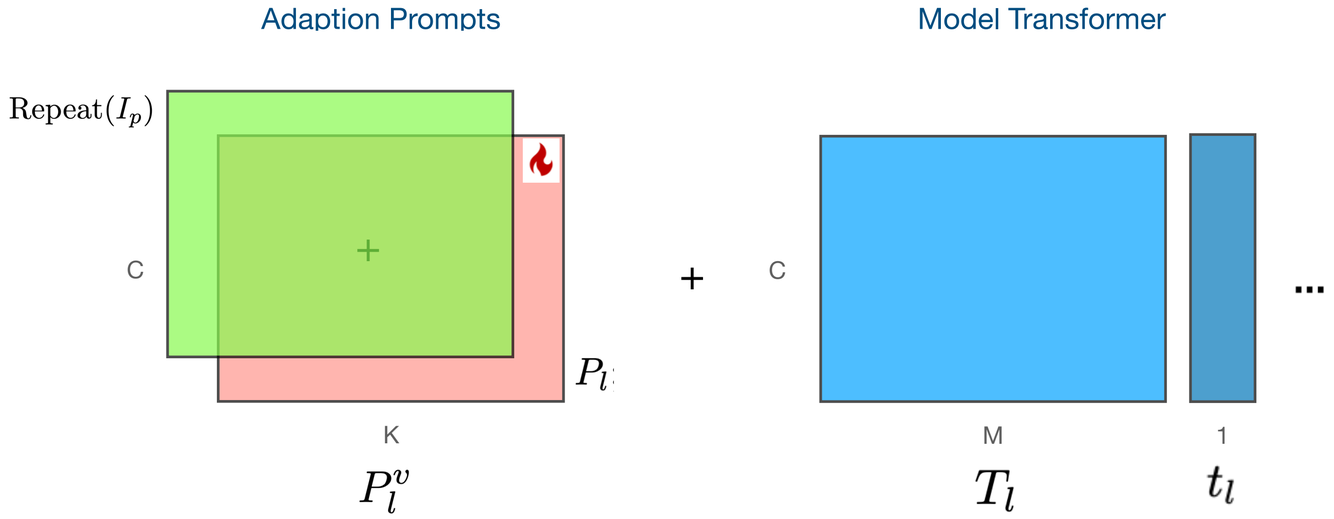


(예시) pre-trained LLaMA로 Multi-modal task 파인튜닝

K: prompt length for each layer
C: feature dimension of model's transformer
M: length for word tokens(input instruction and the already generated response)



<ScienceQA 데이터셋>



프롬프트

디코더

Inputs : + “ 이 이미지에 대한 캡션을 생성해주세요. ###질문: 아기 손에서 어떤 힘이 캐비닛 문을 열까요? ###선택지: (A) 당기는 힘 (B) 미는 힘 ”

<Textual instruction> <Input sequence>

Labels : + “ 이 이미지에 대한 캡션을 생성해주세요. ###질문: 아기 손에서 어떤 힘이 캐비닛 문을 열까요? ###선택지: (A) 당기는 힘 (B) 미는 힘 ### 답변: 정답은 A입니다. ”

<Answer>

K=10, LLaMA의 max length는 보통 1024로 설정함

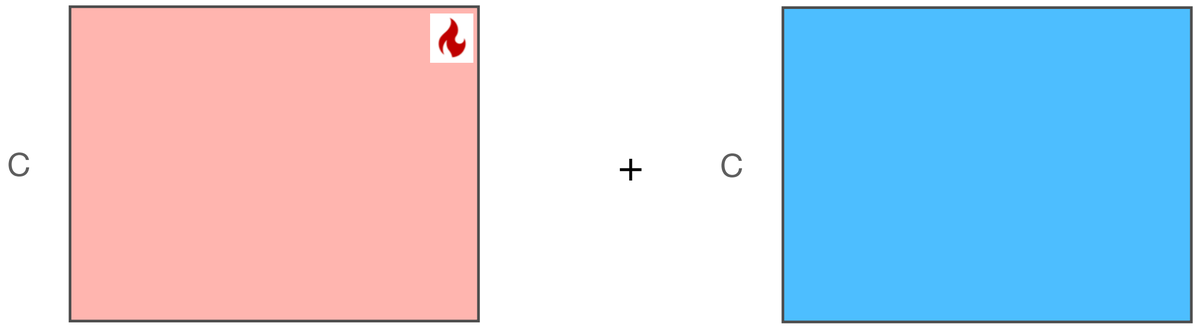
1. Learnable Adaption Prompts

K: prompt length for each layer
C: feature dimension of LLaMA's transformer
M: length for word tokens(input instruction and the already generated response)

$$[P_l;\ T_l] \in \mathbb{R}^{(K+M)\times C}$$

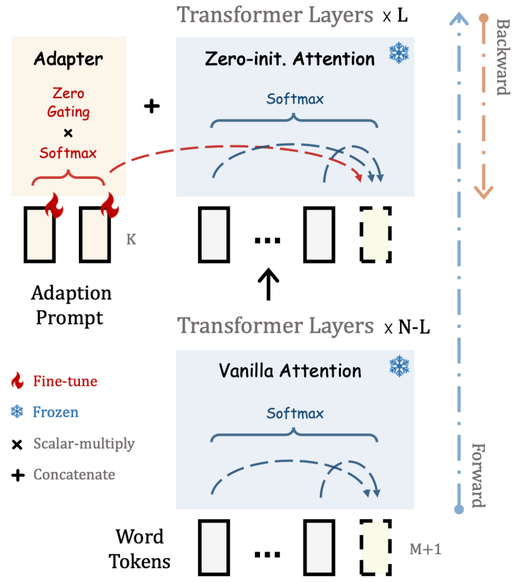
instruction
knowledge 학습

subsequent contextual
response 생성

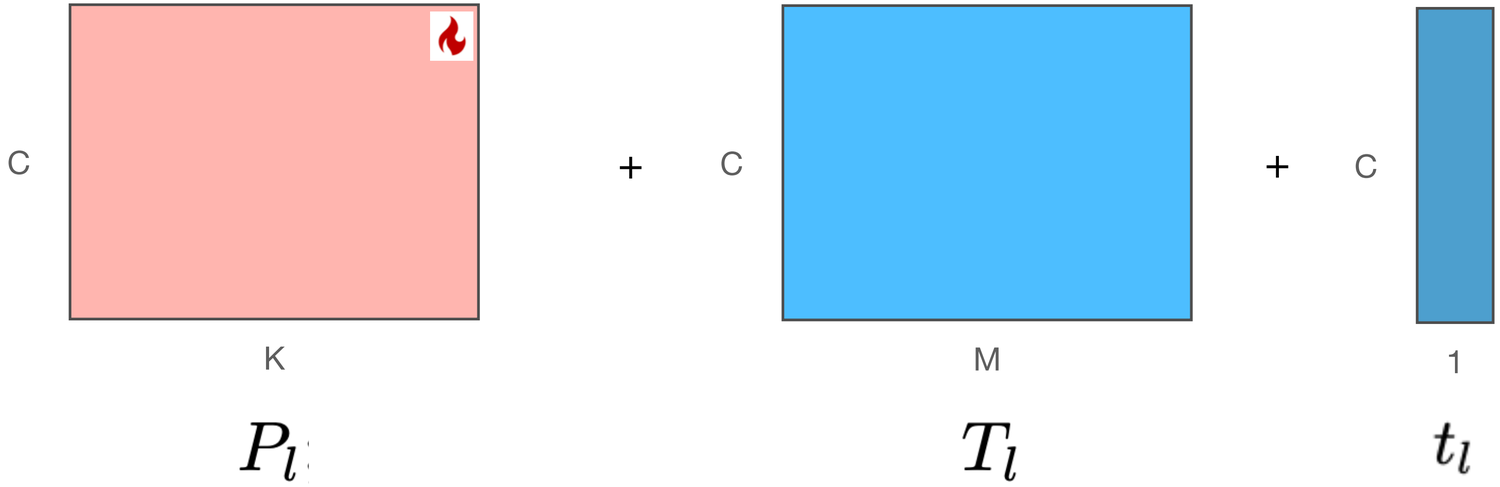


P_l T_l

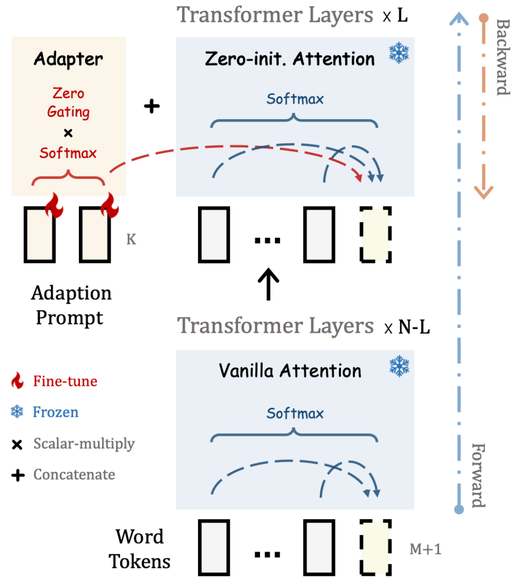
2. Zero-initialized Attention



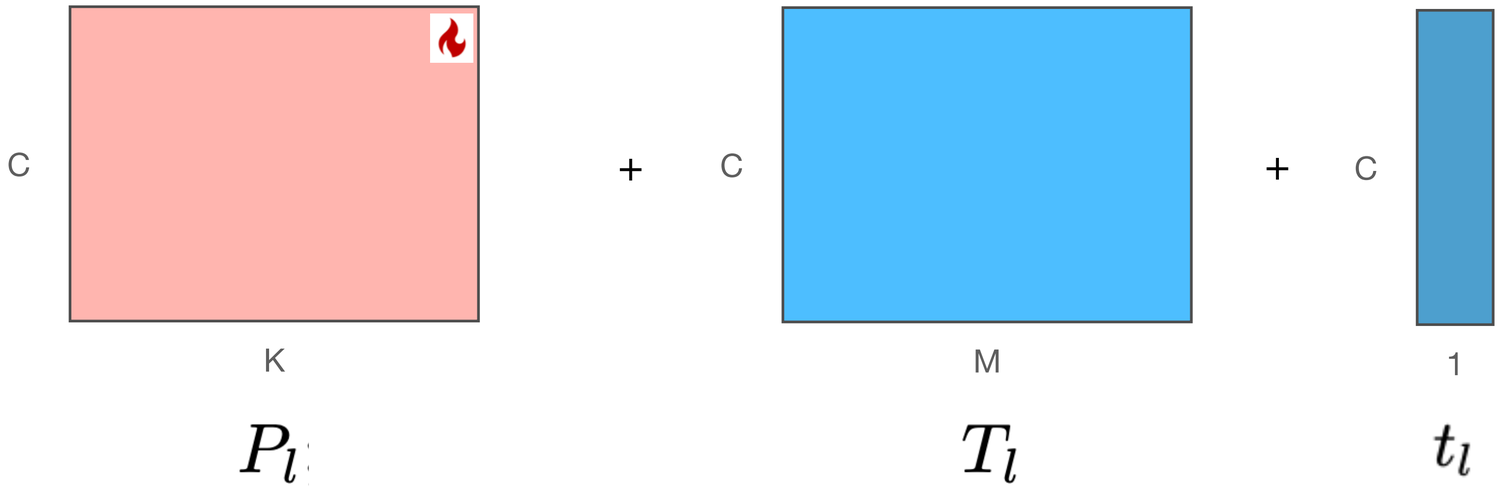
$$
\begin{aligned}
Q_l &= \text{Linear}_\text{q}(\ t_l\ );\\
K_l &= \text{Linear}_\text{k}(\ [P_l;\ T_l;\ t_l]\ );\\
V_l &= \text{Linear}_\text{v}(\ [P_l;\ T_l;\ t_l]\ ).
\end{aligned}
$$



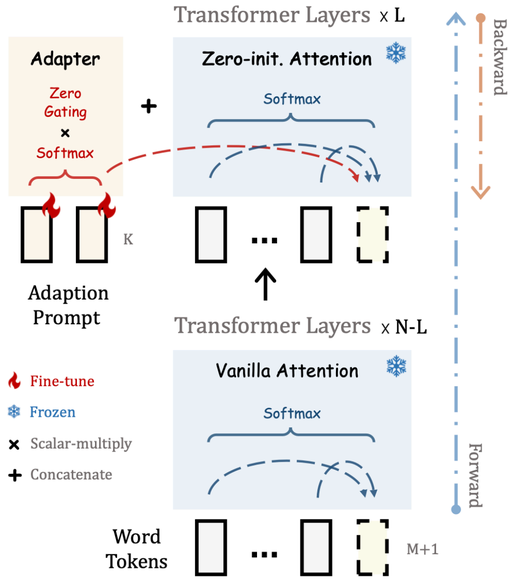
2. Zero-initialized Attention



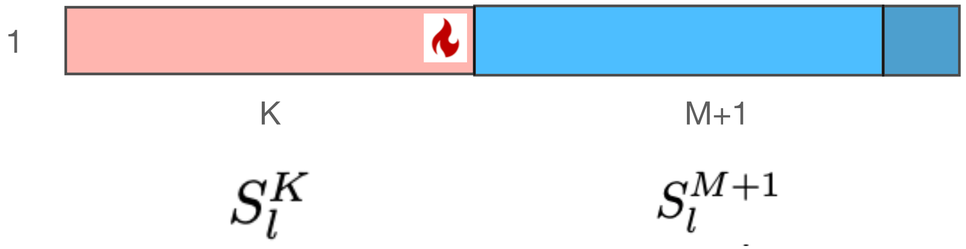
$$S_l = Q_l K_l^T / \sqrt{C} \in \mathbb{R}^{1\times(K+M+1)}$$



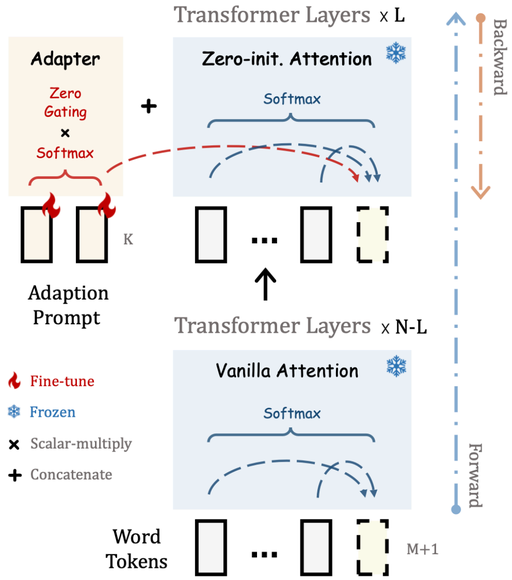
2. Zero-initialized Attention



$$S_l = Q_l K_l^T / \sqrt{C} \in \mathbb{R}^{1\times(K+M+1)}$$

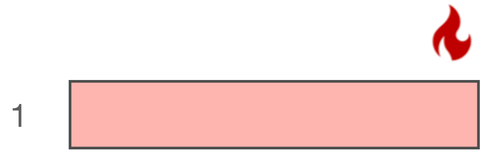


2. Zero-initialized Attention



$$S_l = [S_l^K;\ S_l^{M+1}]^T$$

Adaption prompt



S_l^K

Word tokens

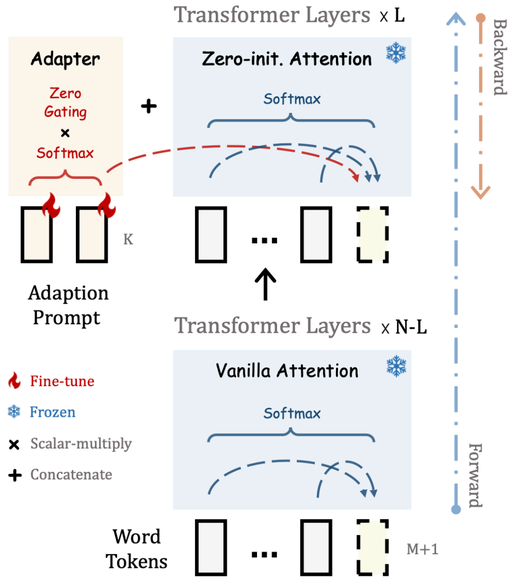
S_l^{M+1}

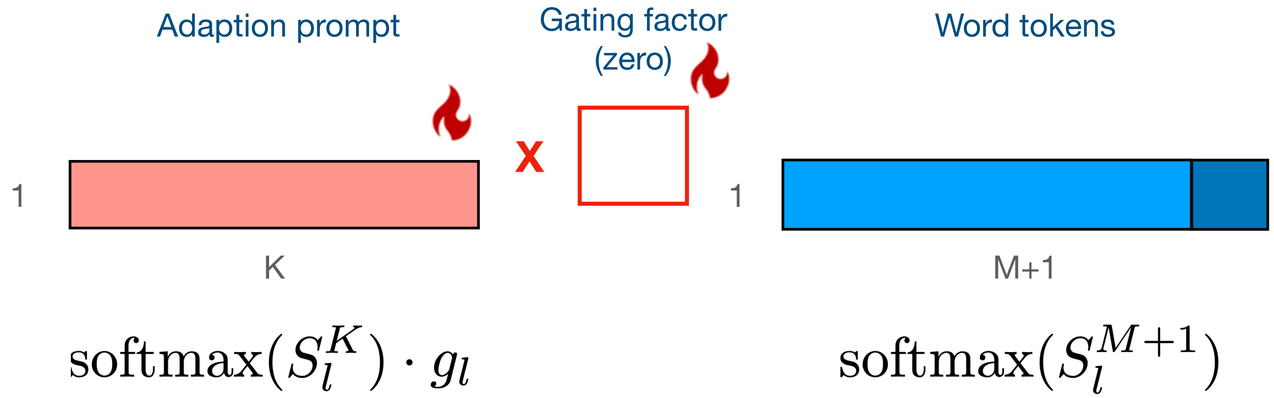
2. Zero-initialized Attention

Gating factor

- Learnable, zero-initialized

$$S_l^g = [\text{softmax}(S_l^K) \cdot g_l;\ \ \text{softmax}(S_l^{M+1})]^T$$



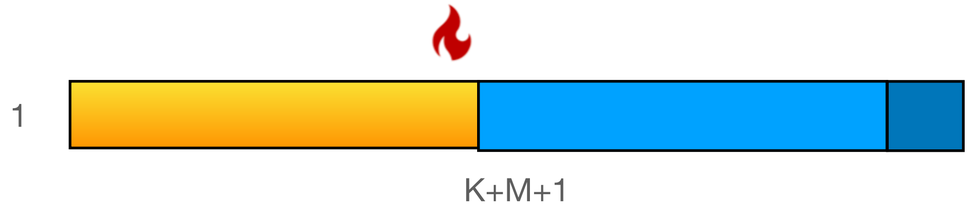


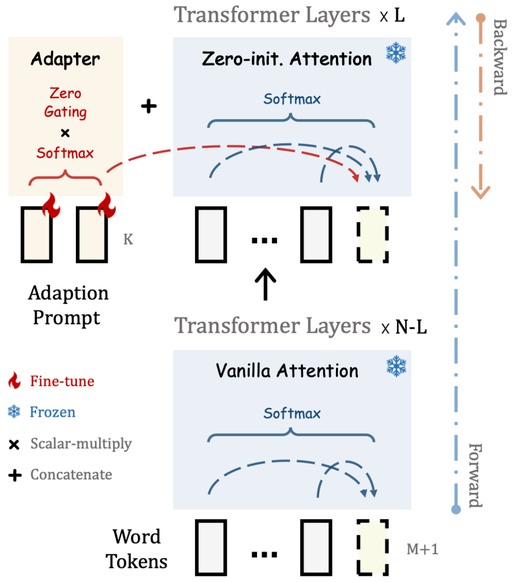
2. Zero-initialized Attention

Gating factor

- Learnable, zero-initialized

$$S_l^g = [\text{softmax}(S_l^K) \cdot g_l;\ \ \text{softmax}(S_l^{M+1})]^T$$

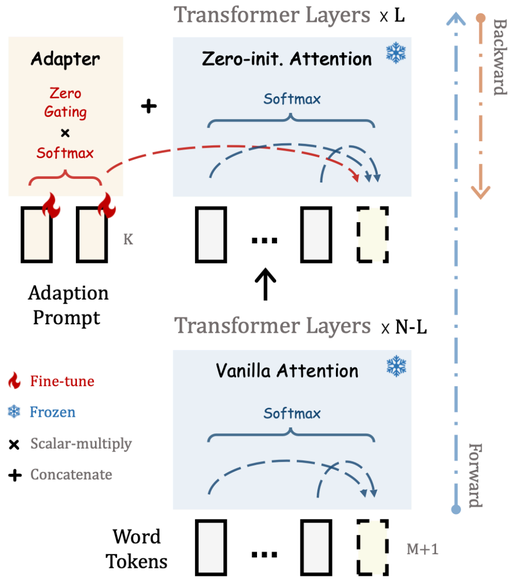




2. Zero-initialized Attention

- Output of the l-th attention layer

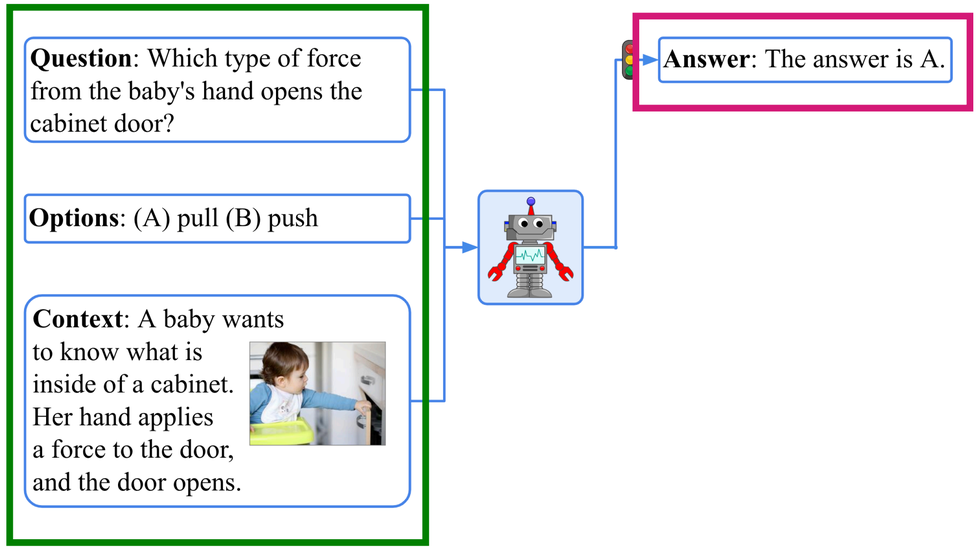
$$t_l^o = \text{Linear}_o(S_l^g V_l) \in \mathbb{R}^{1 \times C}$$



3. Multi-modal Reasoning

멀티 모델 이해가 필요한 생성 tasks

- Ex) ScienceQA 벤치마크, COCO Caption 벤치마크
  - Given visual and textual contexts, along with the corresponding question and options, the model is required to conduct multi-modal understanding to give the correct answer.



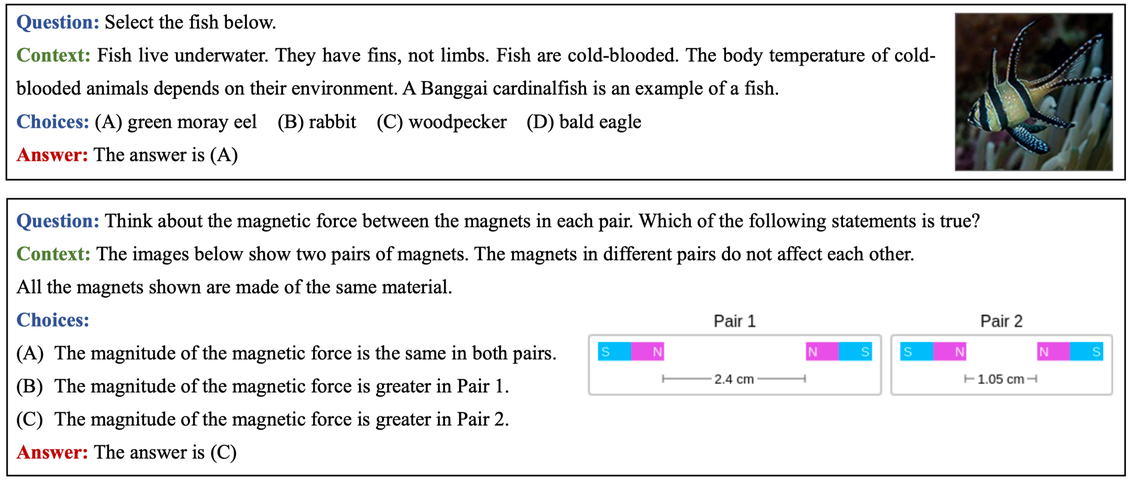
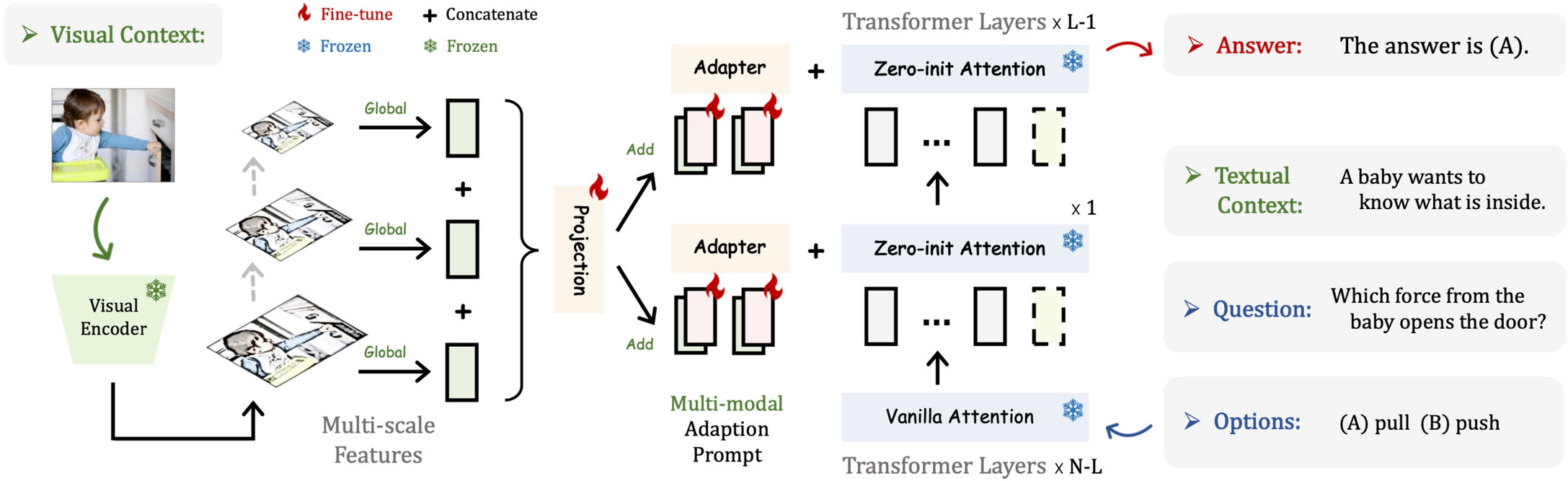


Figure 5: **Multi-modal Reasoning on ScienceQA [41]** dataset by LLaMA-Adapter.

3. Multi-modal Reasoning

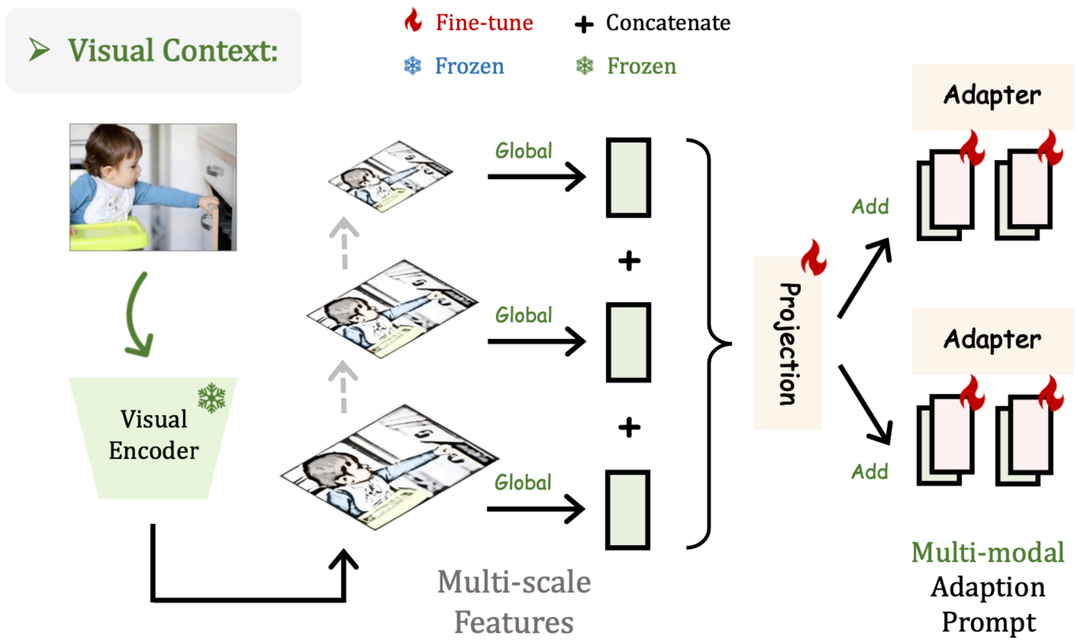
멀티 모델 이해가 필요한 생성 tasks

- LLaMA is fine-tuned to generate responses conditioned on vision-language inputs

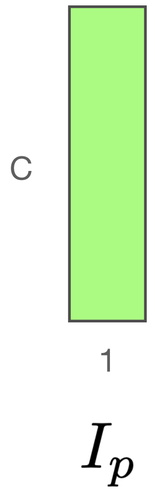


3. Multi-modal Reasoning

P_l^v 구성

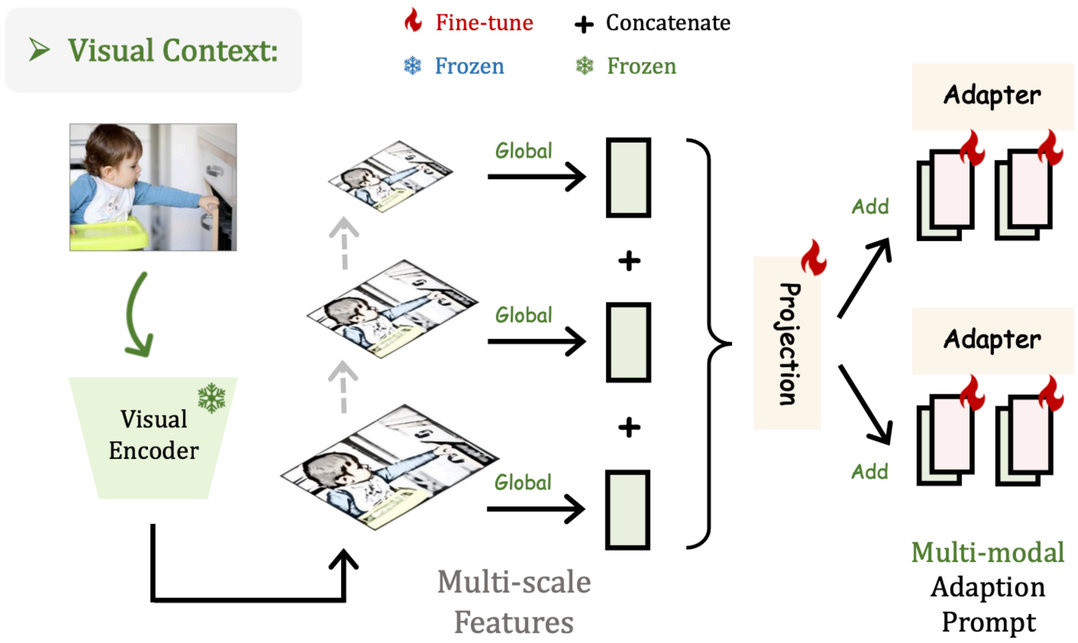


$$I_p = \text{Projection}\left(\text{Concat}\left(\{I_m\}_{m=1}^{M}\right)\right)$$

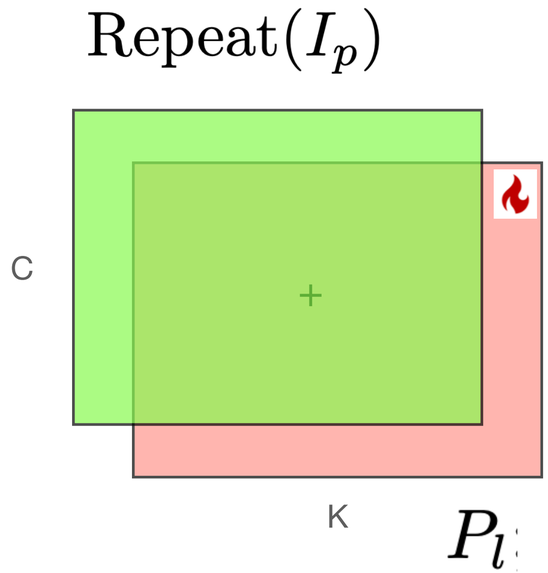


3. Multi-modal Reasoning

P_l^v 구성

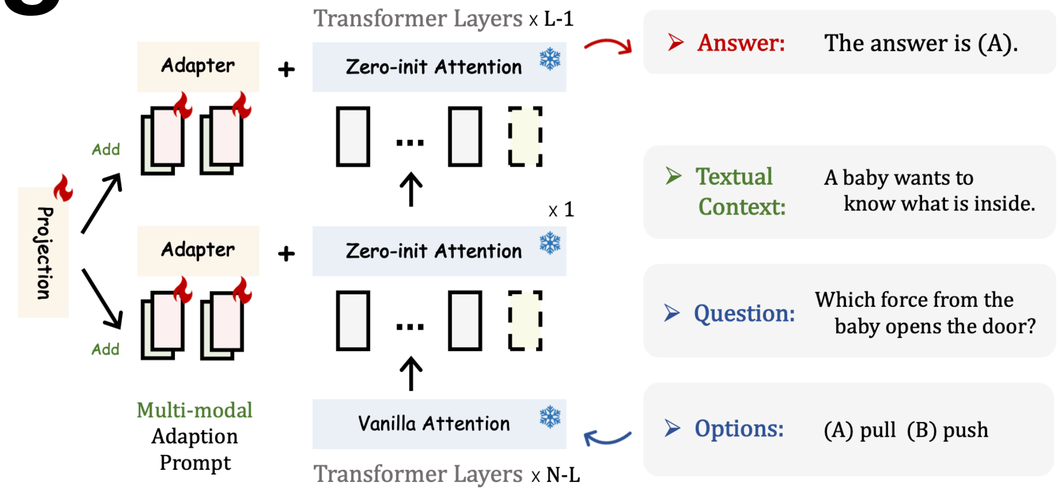


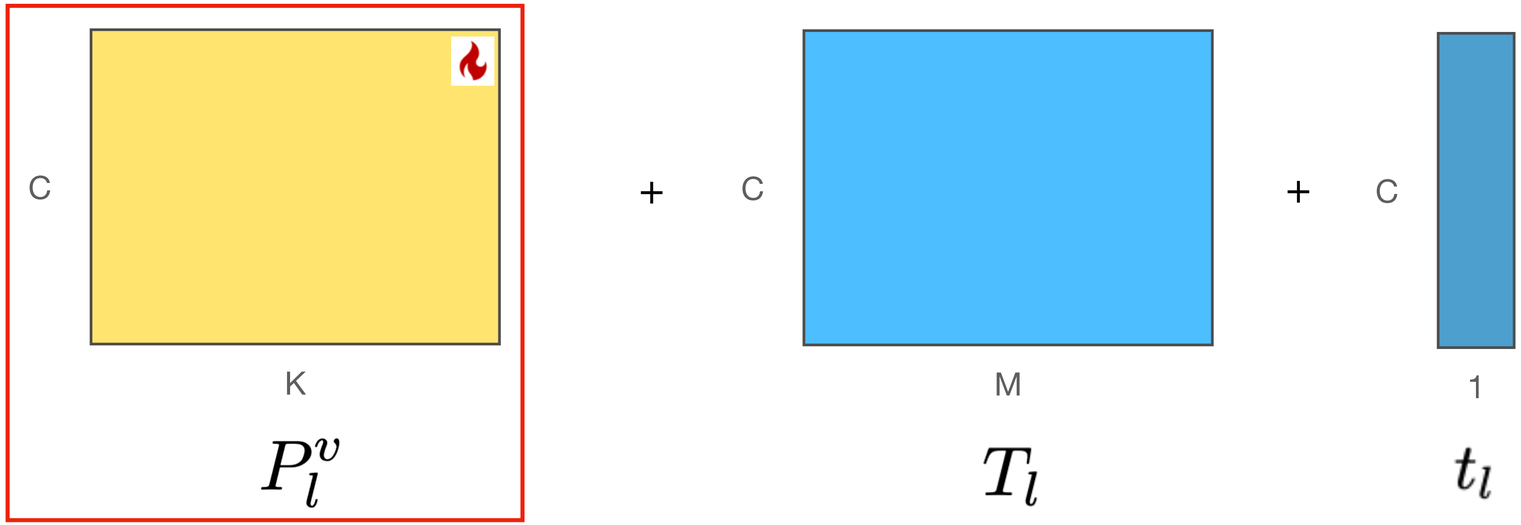
$$P_l^v = P_l + \text{Repeat}(I_p) \in \mathbb{R}^{K \times C}$$



3. Multi-modal Reasoning

Zero-init Attention 수행





Experiment

1. Instruction-following Evaluation

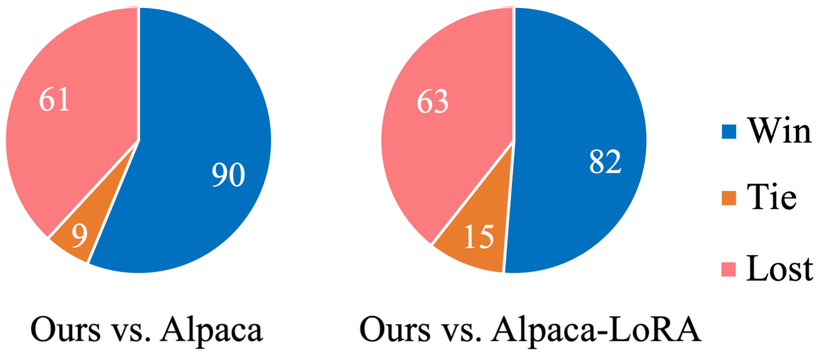
Alpaca 및 Alpaca-LoRA와 비교

Settings

- Alpaca 52K instruction-following 데이터로 5 epoch 파인튜닝 (K=10, L=30)
- 생성 시 temperature=0.1

Performance

- 80개 질문에 대한 responses, GPT-4로 정량평가



Efficiency

- 8 A100 GPUs

Model	Tuned Params	Storage Space	Training Time
Alpaca [60]	7B	13G	3 hours
Alpaca-LoRA [1]	4.2M	16.8M	1.5 hours
LLaMA-Adapter	**1.2M**	**4.7M**	**1 hour**

2. Multi-modal Evaluation

Traditional VQA method와 비교

Settings

- CLIP의 Visual encoder, MLPs로 learnable projection network 구성
- Textual instruction: **“Generate caption for this image”**
- Input sequence: **question + textual context + multiple options + answer**

Performance

- ScienceQA test set

Model	Tuned Params	Avg	NAT	SOC	LAN	TXT	IMG	NO	G1-6	G7-12
Random Choice [41]	-	39.83	40.28	46.13	29.25	47.45	40.08	33.66	39.35	40.67
Human [41]	-	88.40	90.23	84.97	87.48	89.60	87.50	88.10	91.59	82.42
MCAN [65]	95M	54.54	56.08	46.23	58.09	59.43	51.17	55.40	51.65	59.72
VisualBERT [33, 34]	111M	61.87	59.33	69.18	61.18	62.71	62.17	58.54	62.96	59.92
UnifiedQA [27]	223M	70.12	68.16	69.18	74.91	63.78	61.38	77.84	72.98	65.00
UnifiedQA$_{CoT}$	223M	74.11	71.00	76.04	78.91	66.42	66.53	81.81	77.06	68.82
GPT-3 [4]	0M	74.04	75.04	66.59	78.00	74.24	65.74	79.58	76.36	69.87
GPT-3$_{CoT}$	0M	75.17	75.44	70.87	78.09	74.68	67.43	79.93	78.23	69.68
ChatGPT$_{CoT}$ [2]	0M	78.31	78.82	70.98	83.18	77.37	67.92	86.13	80.72	74.03
GPT-4$_{CoT}$ [45]	0M	83.99	85.48	72.44	90.27	82.65	71.49	92.89	86.66	79.04
MM-COT$_T$ [74]	223M	70.53	71.09	70.75	69.18	71.16	65.84	71.57	71.00	69.68
MM-COT	223M	84.91	87.52	77.17	85.82	87.88	82.90	86.83	84.65	85.37
LLaMA-Adapter$_T$	**1.2M**	**78.31**	**79.00**	**73.79**	**80.55**	**78.30**	**70.35**	**83.14**	**79.77**	**75.68**
LLaMA-Adapter	**1.8M**	**85.19**	**84.37**	**88.30**	**84.36**	**83.72**	**80.32**	**86.90**	**85.83**	**84.05**

- COCO Caption validation set

Model	Data Scale		COCO Caption	
	PT	FT	B@4	CIDEr
BLIP [32]	14M	0.6M	40.4	136.7
BLIP-2 [31]	129M	0.6M	43.7	145.3
ClipCap [43]	0	0.6M	33.5	113.1
LLaMA-Adapter	**0**	**0.6M**	**36.2**	**122.2**

3. Ablation Study

L의 개수, 어텐션 매커니즘, 과적합

Inserted Layers

- ScienceQA validation set

Layers	Params	Val Acc (%)
10	0.97	55.95
20	1.37	73.36
30	1.79	**83.85**

Zero-initialized Attention

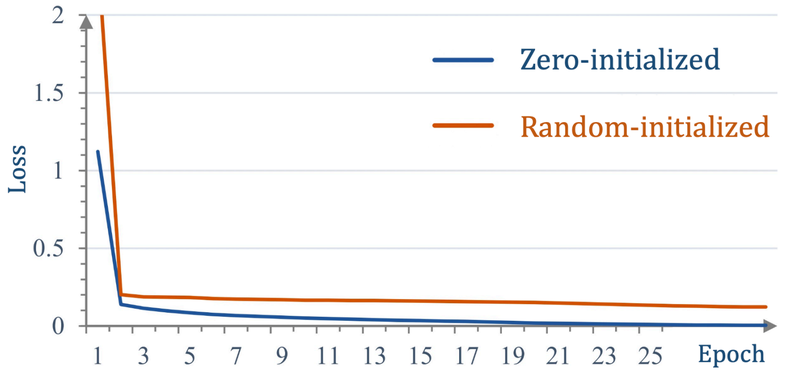
- ScienceQA validation set

Setting	Val Acc (%)
Rand-Init Attention	40.77
Zero-Init Attention	**83.85**
Gain	+43.08

Robustness to Over-fitting

- ScienceQA

Epoch	Train Loss	Val Loss	Val Acc (%)
15	0.022	0.136	82.08
30	0.004	0.241	83.85
60	0.001	0.282	**83.94**



4. Zero-initialized Attention for other Large Models

Traditional Vision model에 적용

Settings

- 사전학습된 ViT-B/16 모델
- Image classification Task
  - VTAB-1k 데이터셋으로 파인튜닝

Performance

- Image classification

Method	Natural	Specialized	Structured
Full	75.88	83.36	47.64
Bias [66]	73.30	78.25	44.09
Adapter [22]	70.39	77.11	33.43
Sidetune [68]	58.21	68.12	23.41
VPT [24]	78.48	82.43	54.98
Zero-init.	**81.74**	**84.43**	**56.75**

(Average accuracy)

4. Zero-initialized Attention for other Large Models

Traditional Language model에 적용

Settings

- 사전학습된 RoBERTa large 모델
- Extractive question answering Task
  - SQuAD v1.1과 v2.0 벤치마크로 파인튜닝
  - P-tuning v2(PT2) + zero-initialized attention

Performance

- Extractive QA

Method	SQuAD 1.1 dev		SQuAD 2.0 dev	
	EM	F1	EM	F1
Full	88.9	94.6	86.5	89.4
PT [30]	1.2	12.0	50.2	50.2
PT2 [38]	88.5	94.4	82.1	85.5
PT2*	88.1	94.2	81.3	84.7
Zero-init.	**88.8**	**94.6**	**83.9**	**87.2**

(Exact Match(EM) & F1 scores)

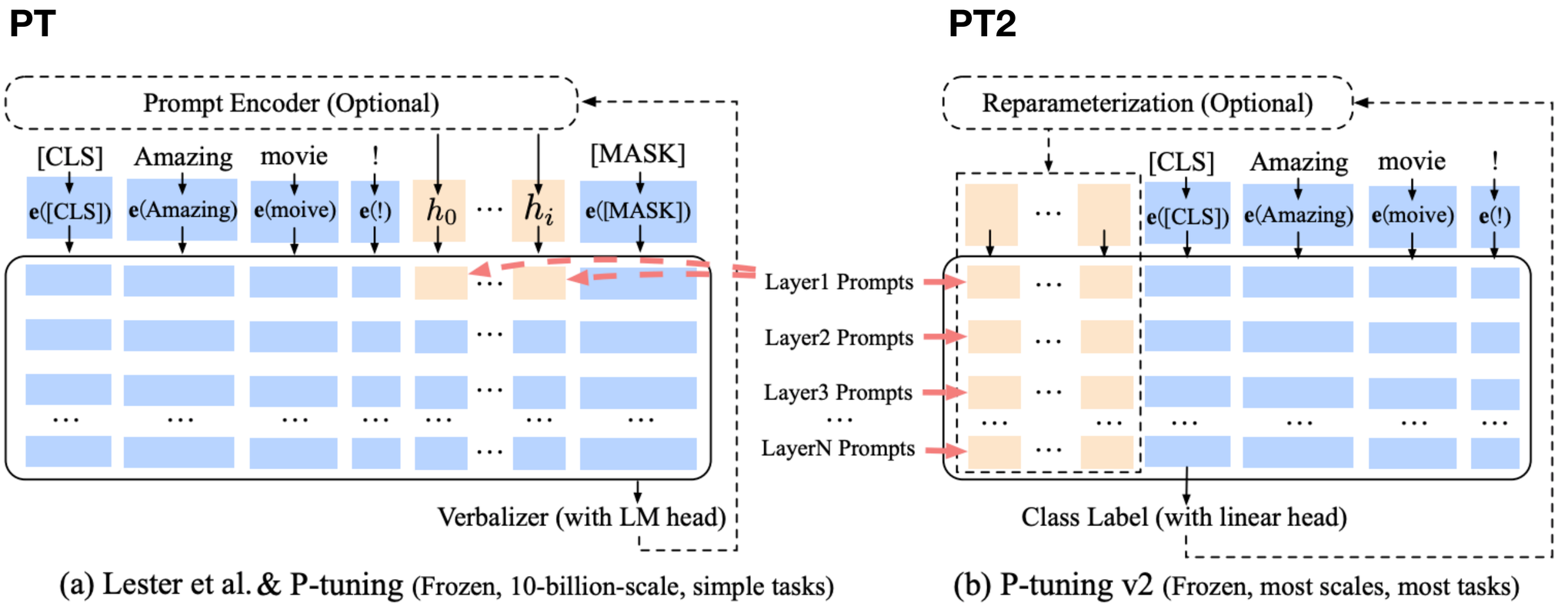
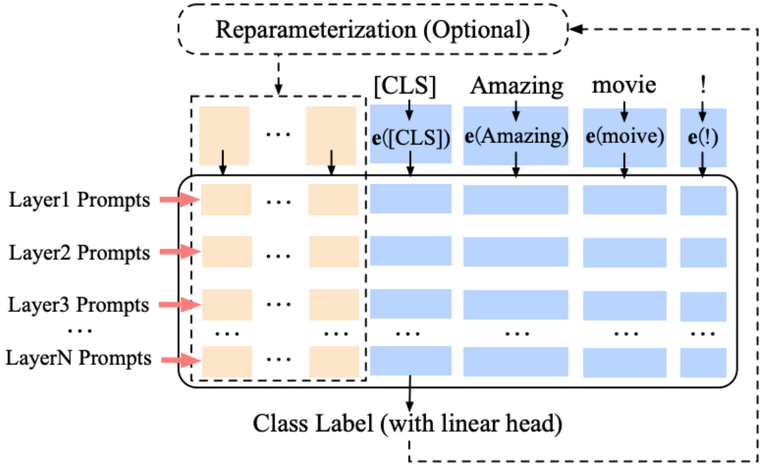
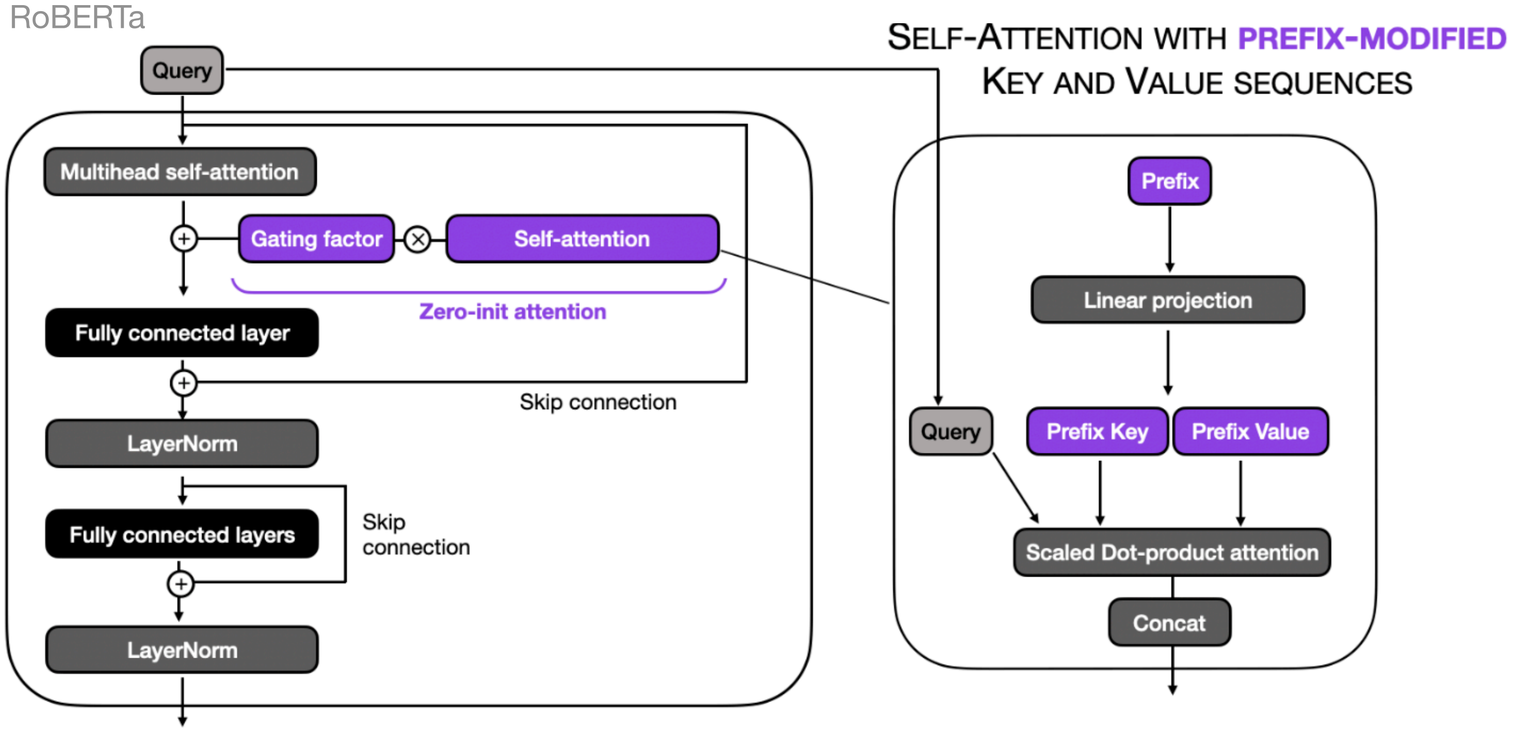


Figure 2: From Lester et al. (2021) & P-tuning to P-tuning v2. Orange tokens (include h_0, h_i) refer to prompt embeddings we add; blue tokens are embeddings stored or computed by frozen pre-trained language models. Compared to Lester et al. (2021), P-tuning v2 adds trainable continuous prompts to inputs of every transformer layer independently (as prefix-tuning (Li and Liang, 2021) does). Additionally, P-tuning v2 removes verbalizers with LM head and returns to the traditional class labels with ordinary linear head to allow its task-universality.

Zero-init


Reparameterization (Optional)
[CLS]
Amazing
movie
!
e([CLS])
e(Amazing)
e(moive)
e(!)
Layer1 Prompts
Layer2 Prompts
Layer3 Prompts
LayerN Prompts
Class Label (with linear head)


RoBERTa
Query
Multihead self-attention
Gating factor
Self-attention
Zero-init attention
Fully connected layer
Skip connection
LayerNorm
Fully connected layers
Skip connection
LayerNorm
SELF-ATTENTION WITH PREFIX-MODIFIED KEY AND VALUE SEQUENCES
Prefix
Linear projection
Query
Prefix Key
Prefix Value
Scaled Dot-product attention
Concat

4. Zero-initialized Attention for other Large Models

Traditional Language model에 적용

Settings

- 사전학습된 RoBERTa large 모델
- named entity recognition (NER) or semantic role labeling (SRL) Task
  - CoNLL03, CoNLL04, CoNLL12, CoNLL05Brown, CoNLL05WSJ로 파인튜닝
  - P-tuning v2(PT2)로 파인튜닝

Performance

- NER, Relation extraction, Coreference, SRL

Method	CoNLL03 [56]	CoNLL04 [5]	CoNLL12 [49]	CoNLL05$_{Brown}$ [6]	CoNLL05$_{WSJ}$ [6]
Full	92.6	88.8	86.5	85.6	90.2
PT [30]	86.1	76.2	67.2	70.7	76.8
PT2 [38]	92.8	88.4	84.6	84.3	89.2
PT2*	91.8	88.4	84.7	83.9	89.4
Zero-init.	**92.4**	**88.8**	**85.2**	**84.7**	**89.6**

(Micro-f1 score)

B.3 Fine-tuning Vision-Language Models

Vision-Language model에 적용

Settings

- 사전학습된 CLIP 모델
- CLIP with a ViT-B/16으로 Visual encoder, 12-layer transformers로 Textual encoder 구성
- Image Classification Task
  - Base-to-novel generalization 벤치마크로 파인튜닝
    - 훈련은 Base로만 few-shot
    - 평가는 Base, Novel 둘다 (HM: Harmonic Mean)

Performance

- Image classification

Method	ImageNet [14]			Caltech101 [17]			Flowers102 [44]			Average		
	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM
CLIP [51]	72.43	68.14	70.22	96.84	94.00	95.40	72.08	77.80	74.83	80.45	79.98	80.15
CoOp [76]	76.47	67.88	71.92	98.00	89.81	93.73	97.60	59.67	74.06	90.69	72.45	79.90
CoCoOp [75]	75.98	70.43	73.10	97.96	93.81	95.84	94.87	71.75	81.71	89.60	78.66	83.55
MaPLe [28]	76.66	70.54	73.47	97.74	94.36	96.02	95.92	72.46	82.56	90.11	79.12	84.02
Zero-init.	**76.70**	**71.00**	**73.74**	**98.10**	**94.53**	**96.28**	**96.00**	**74.67**	**84.00**	**90.27**	**80.07**	**84.67**

(Classification accuracy & HM: Harmonic Mean)

느낀점

- P-LLM의 학습된 정보는 최대한 활용, Instructions를 잘 따르도록 align
- 멀티 모달의 경우에도 이미지는 캡셔닝만, 추론 능력은 언어 쪽에서 가지고 있음
    - 이미지는 Prompt 벡터에 붙어 구성됨
- 효율적인 멀티 모달 가능
    - GPT-4 VS LLaMA-Adapter
    - 추정치? VS 1.2M
    - Open-domain VS Task-specific
- 파인튜닝 시간
    - LLaMA PEFT(LoRA)으로 1 에폭 96시간 → 5 에폭 20일?
    - Colab T4 GPU

주의 요함 | llama-adapter.opengvlab.com

LLaMA-Adapter V2

For Bilingual Multi-modality Instruction Model, please try ImageBind-LLM

Multi-Modal Interaction

Input

Output

The painter of this image is Johannes Vermeer, a Dutch painter who was active in the Dutch Golden Age during the 17th century. He is known for his exquisite and detailed paintings, often featuring women in domestic settings.

Prompt

Introduce the painter of this image?

Max length 512

Advanced options

Temperature 0.1

Top p 0.75

Run

GPT-4 visual input example, Extreme Ironing:	
User	What is unusual about this image?
	 Source: https://www.barnorama.com/wp-content/uploads/2016/12/03-Confusing-Pictures.jpg
GPT-4	The unusual thing about this image is that a man is ironing clothes on an ironing board attached to the roof of a moving taxi.

Table 16. Example prompt demonstrating GPT-4's visual input capability. The prompt requires image understanding.

GPT-4 visual input example, Chicken Nugget Map:	
User	Can you explain this meme?
	
GPT-4	This meme is a joke that combines two unrelated things: pictures of the earth from space and chicken nuggets. The text of the meme suggests that the image below is a beautiful picture of the earth from space. However, the image is actually of chicken nuggets arranged to vaguely resemble a map of the world. The humor in this meme comes from the unexpected juxtaposition of the text and the image. The text sets up an expectation of a majestic image of the earth, but the image is actually something mundane and silly.

Table 18: Example prompt demonstrating GPT-4's visual input capability.

Q & A

Appendix

Instruction: Tell me about alpacas.

Alpaca: Alpacas are small, fluffy animals related to camels and llamas. They are native to Peru and Bolivia, and were first domesticated around 5,000 years ago. They are kept mainly for their fine, soft fleece, which is used to make knitwear and other garments. Alpacas are herd animals and live in small family groups. They are highly social animals and can form strong bonds with their owners.

LLaMA-Adapter: Alpacas are members of the camelid family and are native to the Andes Mountains of South America. They are typically found in herds of 10-20 animals and are known for their gentle nature and their ability to thrive in harsh climates. Alpacas are typically white or grey in color and have long, silky coats. They are also known for their ability to produce high-quality wool, which is used in the production of clothing and other products.

Instruction: Translate the sentence 'I have no mouth but I must scream' into Spanish.

Alpaca: No tengo boca, pero debo gritar.

LLaMA-Adapter: No tengo boca pero tengo que gritar.

Instruction: Write a Python program that prints the first 10 Fibonacci numbers.

Alpaca:

```
numbers = [0, 1]
for i in range(2, 11):
    numbers.append(numbers[i-2] + numbers[i-1])
print(numbers)
```

LLaMA-Adapter:

```
def fibonacci():
    a, b = 0, 1
    for i in range(10):
        a, b = b, a + b
        print(a, b)
    print("Fibonacci numbers: ", a, b)
fibonacci()
```

```
def transformer_block_with_llama_adapter(x, gating_factor, soft_prompt):
    residual = x
    y = zero_init_attention(soft_prompt, x)  # llama-adapter: prepend prefix
    x = self_attention(x)
    x = x + gating_factor * y  # llama-adapter: apply zero_init_attention
    x = LayerNorm(x + residual)
    residual = x
    x = FullyConnectedLayers(x)
    x = LayerNorm(x + residual)
    return x
```