Memory-Space Visual Prompting for Efficient Vision-Language Fine-Tuning

Journal Club 2024/06/01

中村圭佑

目次

概要

背景

- 従来のVL(Vision and Language)モデルにおけるPEFT(parameter-efficient fine-tuning)の手法では、Visual tokenとText tokenを結合して言語モデルに与えていたため、計算量が増加していた。

提案手法(MemVP)

- 計算量を抑えるために、Visual tokenをFFN層に加える手法が提案された。
- FFN層はkey-valueの関係性があるため、この方法が有効。

結果

- Visual promptを言語モデルの重みに結合することで訓練と推論の効率を改善。
- 様々なVLタスクにおいて、ファインチューニングされたモデルの訓練時間と推論遅延を大幅に削減。
- 従来のPEFT手法および事前学習から訓練したVLモデル(LLaVA等)の性能を上回った。

関連研究

Vision and Languageの概要

Vision and Languageとは？

- Computer Vision(CV)とNatural Language Processing(NLP)を組み合わせた研究分野。

Vision and Languageのアプローチ

- Vision and Languageのタスクは全く異なる形式のデータを効果的に結合する必要がある。
- そのため、異なるモーダル間の情報を統合するために共有の特徴空間にマッピングする。

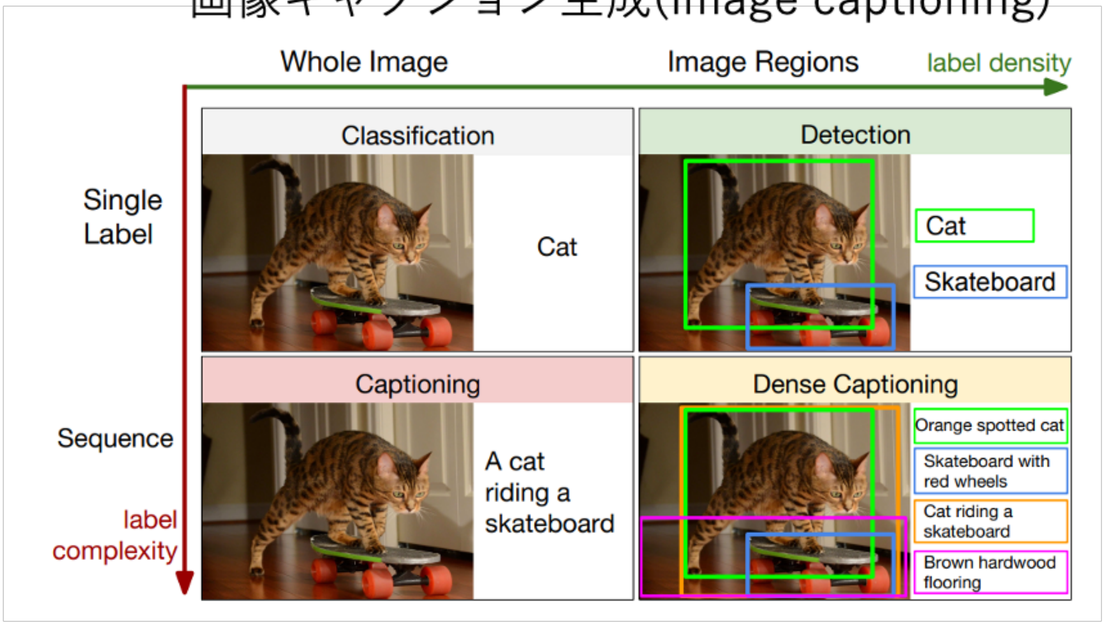
関連研究

Vision and Languageの概要(タスクの一例)

応用タスク

- Image Captioning, Visual Question Answering, Image Retrievalなど

画像キャプション生成(Image captioning)



Densecap: Fully convolutional localization networks for dense captioning.

視覚的質問対応(VQA: Visual Question Answering)

Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering.

関連研究

Vision and Languageの概要(共通空間のマッピング手法)

CLIP(Contrastive Language-Image Pre-training)

- 目的：画像とテキストを関連付ける。
- 方法：大量の画像とテキストペアに対して対照学習を行う。

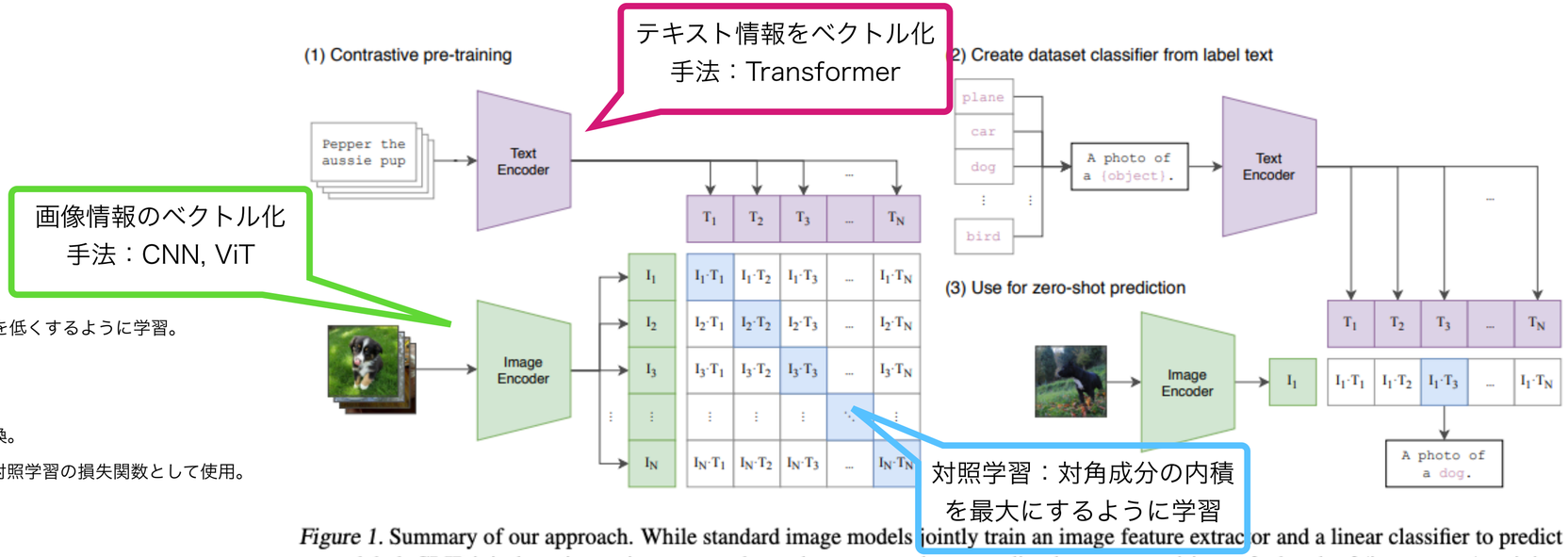


Figure 1. Summary of our approach. While standard image models jointly train an image feature extractor and a linear classifier to predict some label, CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes.

引用元：Learning Transferable Visual Models From Natural Language Supervision

詳細 via ChatGPT

対照学習

画像とテキストのペアを使用し、同じペアの関連性を高く、異なるペアの関連性を低くするように学習。

アーキテクチャ

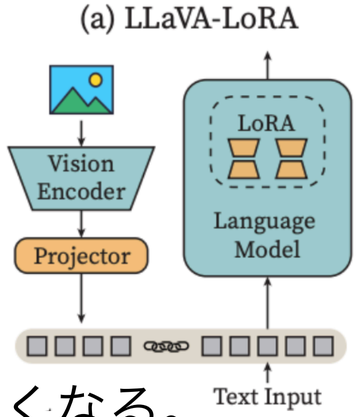
- 視覚エンコーダ：画像をエンコードし、固定次元の特徴ベクトルに変換。
- テキストエンコーダ：テキストをエンコードし、固定次元の特徴ベクトルに変換。
- 一致させる方法：視覚ベクトルとテキストベクトルの内積を計算し、その値を対照学習の損失関数として使用。

応用例

- 画像検索：テキストクエリに対して関連する画像を検索。
- 画像キャプショニング：画像に対して適切なキャプションを生成。
- ゼロショット学習：見たことのないカテゴリーの画像に対してもテキストで説明する能力。

関連研究

PEFTの手法について



MemVP 図1

- Vision and Languageモデルを事前学習から行うと学習コストがとても大きくなる。
- そこで、多くの研究では事前学習されたVisual Encoderと言語モデルを用いて効率的にVLを構築する。
- その手法の一つとしてPEFT(parameter-efficient fine-tuning)がある。
  - AdapterやLoRAなど
- PEFTの従来手法ではVisual tokenとText tokenを結合して言語モデルに与える。
  - Visual tokenがText tokenと比べて大きい場合に、計算量が大きくなるという問題がある。
  - 従来モデルは、Visual promptはパラメータ効率化できているが訓練や推論の計算効率が良くない。

関連研究

Visual promptが及ぼす影響

- LLaMA-7Bの入力（トークンシーケンス）の長さが増加することによる訓練/推論時間の影響を計算している
- テキストの入力長を64, 画像情報の入力長を256とするとテキストのみと画像を加えた場合を比較して訓練で2.6倍、推論で4.8倍の時間がかかる。

テキスト(64)＋画像(256)

テキストのみ(64)

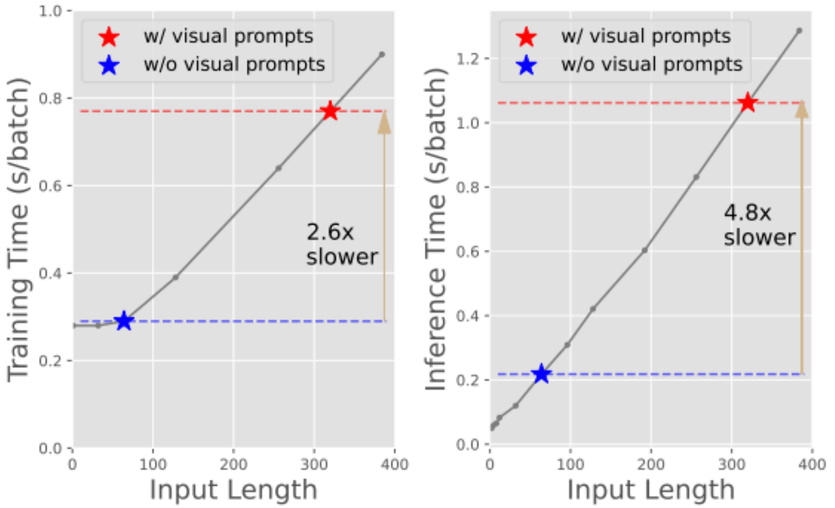


Figure 2. **Training and inference time of LLaMA-7B on a single V100.** The training process adopts PEFT in which we only tune LoRA modules. The training batch size and inference batch size are 4 and 16, respectively, to maximize utilization of GPU memory. We also highlight the position when the text token length is 64 w/ and w/o input-space visual prompts. The length of visual prompts is 256 as in LLaVA. We fix the output length to 1.

Memory-Space Visual Prompting for Efficient Vision-Language Fine-Tuning

関連研究

FFN層のKey-Valueの関係性(Transformer)

- TransformerのアーキテクチャではMHSA(Multi-Head Self Attention)層とFFN(Feed-Forward Networks)層を組み合わせている。
  - Self-Attention層のAttentionの計算は以下の変換が行われている。

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

  - FFN層は以下の変換が行われている。

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (2)$$

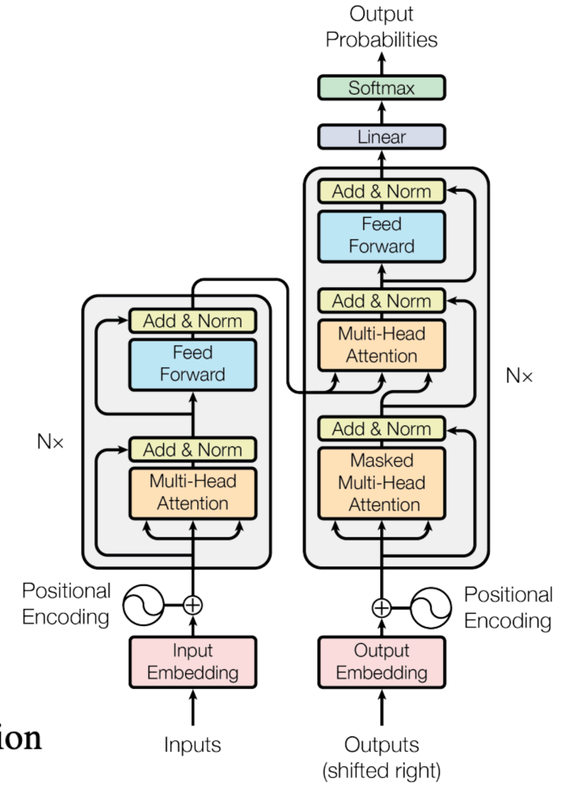
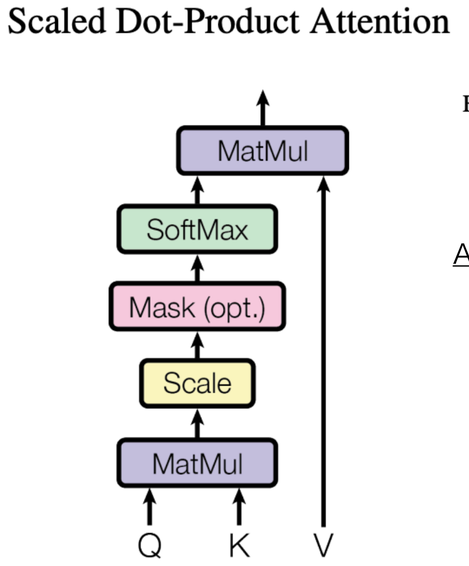


Figure 1: The Transformer - model architecture.



Attention Is All You Needの図を参照

関連研究

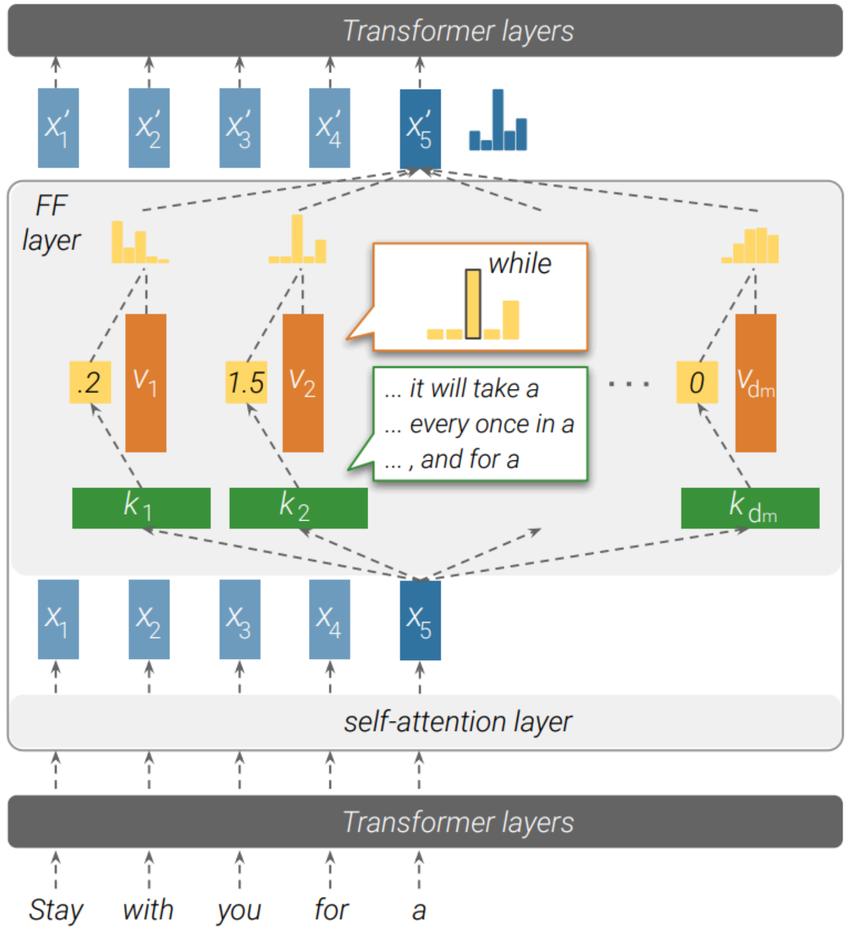
FFN層のKey-Valueの関係性(Transformer)

- Transformer Feed-Forward Layers Are Key-Value MemoriesではTransformerのFFN層のパラメータがKeyとValueの役割をすることが示されている。
- 各Keyが訓練例のテキストパターンに対応し、各Valueが出力語彙分布に対応している。

$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

↓ パラメータをKey, Valueとする

$$\mathrm{FF}(\mathbf{x}) = f(\mathbf{x} \cdot K^\top) \cdot V \qquad (1)$$



Transformer Feed-Forward Layers Are Key-Value Memoriesの図を参照

提案手法：MemVP

新規性

- 従来手法のPEFTでは、Visual TokenとText tokenを結合して言語モデルに与えていた。
- 計算量の効率化のために、FFN層のパラメータにVisual tokenを結合するMemVPを提案している。
  - MemVP: Memory-space visual prompting

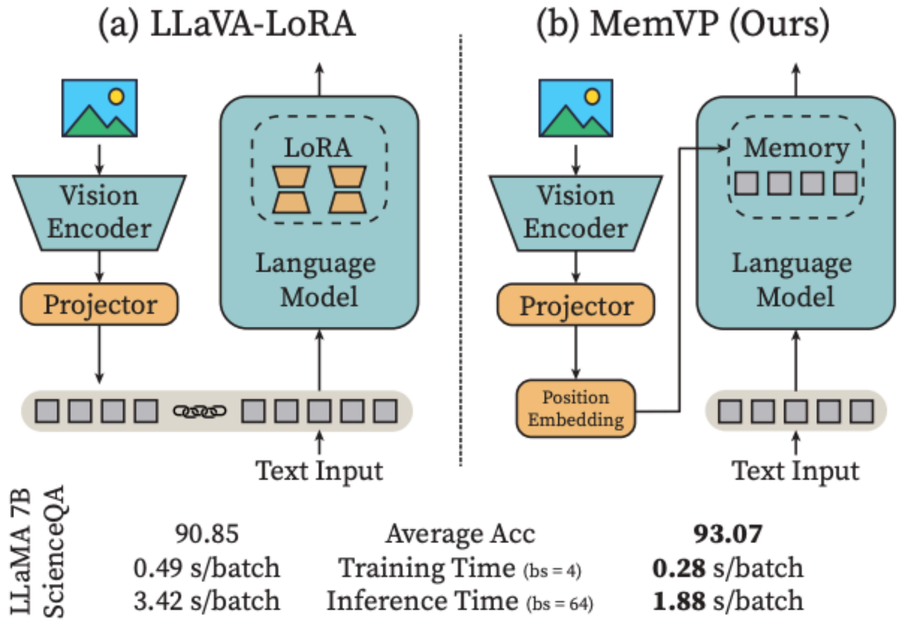


Figure 1. Illustration of PEFT methods using (a) the conventional input-space visual prompting and (b) our memory-space visual prompting. MemVP outperforms previous paradigms in terms of performance, training speed, and inference speed.

Memory-Space Visual Prompting for Efficient Vision-Language Fine-Tuningの図1

提案手法：MemVP

アーキテクチャ①：イメージ

画像情報の追加で入力次元が増える

Input-space visual prompting と比較すると計算量の増加は無視できる程度

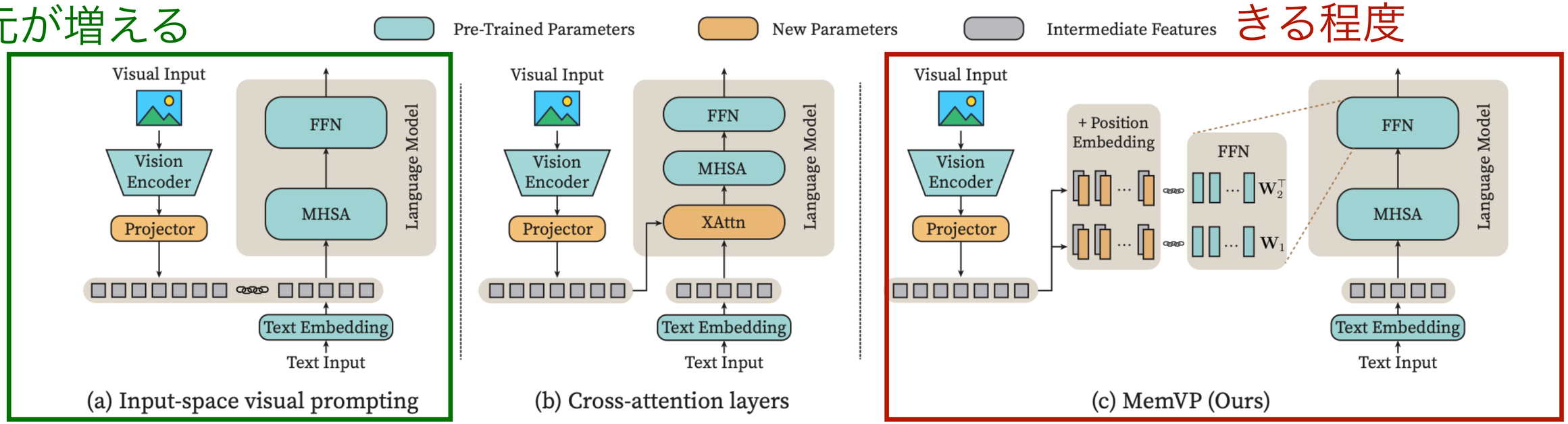
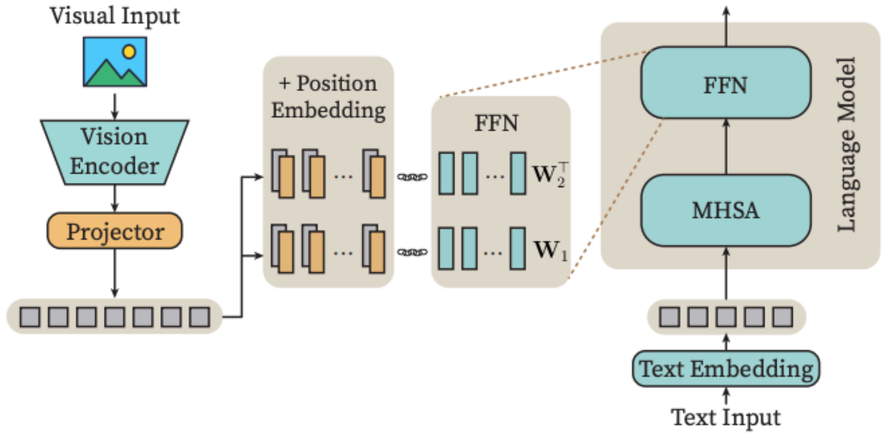


Figure 3. **Overview of the mainstream paradigms to concatenate vision encoder and language model. (a)** Concatenating visual prompts with the text tokens as inputs of the language model is not computation-efficient, *e.g.*, LLaVA, VL-Adapter, VL-PET. **(b)** Using cross-attention layers to incorporate the visual information from visual tokens is not parameter-efficient, *e.g.*, Flamingo, BLIP. **(c)** Our MemVP injects visual prompts into the FFN blocks of language models, achieving both parameter and computation efficiency.

Memory-Space Visual Prompting for Efficient Vision-Language Fine-Tuning

提案手法：MemVP

アーキテクチャ②：FFN層の拡張



(c) MemVP (Ours)

Memory-Space Visual Prompting for Efficient Vision-Language Fine-Tuning

FFN層

一般的なFFN層は$\mathrm{FFN}(x) = \phi(xW_1)W_2$である。ここで、それぞれのパラメータを

$W_1 = (k_1, k_2, \ldots, k_D), W_2 = (v_1, v_2, \ldots, v_D)^\top$とすると、$\mathrm{FFN}(x) = \sum_{i=1}^{D} \phi(\langle x, k_i \rangle) \cdot v_i$となる。

Visual Prompting

Visual featuresを$Z = (z_1, z_2, \ldots, z_n)^\top$として、$z_i$のKey, Valueを$K(z_i), V(z_i)$とすると

$\mathrm{Retrieval}(x) = \sum_{i=1}^{n} \phi(\langle x, K(z_i) \rangle) \cdot V(z_i)$と表せる。ただし、$K(z_i) = \lambda f(z_i) + p_k^i, V(z_i) = \lambda f(z_i) + p_v^i$とする。

*fは画像情報の投影層、λはハイパーパラメータ、p_i^k, p_i^vはVisual promptの位置埋め込み

Memory-SpaceへのVisual Promptの拡張

$$\mathrm{FFN}(x) = \underbrace{\sum_{i=1}^{D} \phi(\langle x, k_i \rangle) \cdot v_i}_{\text{テキスト情報}} + \underbrace{\sum_{i=1}^{n} \phi(\langle x, K(z_i) \rangle) \cdot V(z_i)}_{\text{画像情報}}$$

事前学習済みのパラメータを利用する　Visual Promptの位置埋め込みを学習

提案手法：MemVP

計算量：LM

- MHSA(Multi-Head Self-Attention)ブロックとFFN(Feed-Forward)ブロックだけで構成される言語モデルの層の計算量(FLOPs)を考える。
- 前提条件：バイアス項と正規化層は除外している。また、ここでは”Attention Is All You Need”のアーキテクチャを採用している。
- 変数の定義
  - L：トークンシーケンスの長さ
  - d：トークンの次元
  - n：Visual Promptの長さ

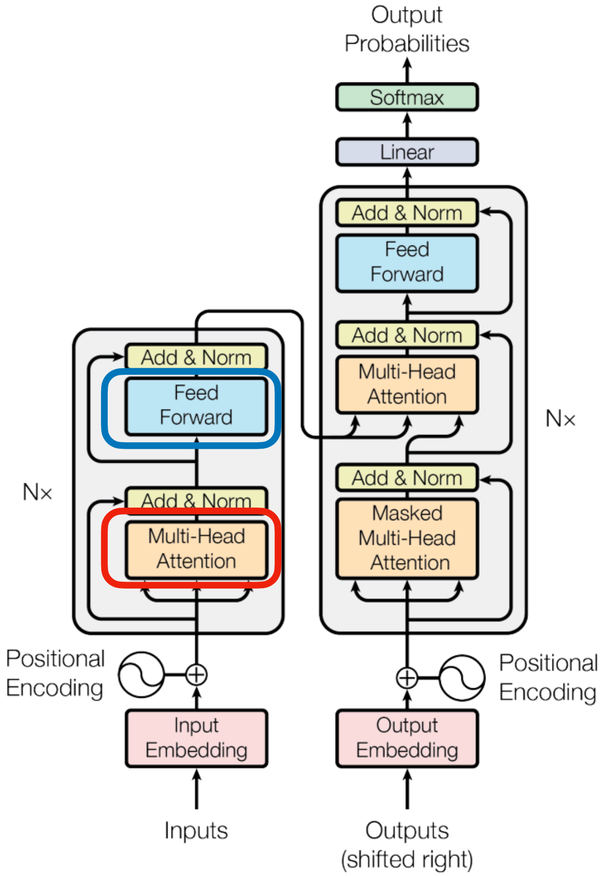


Figure 1: The Transformer - model architecture.

Attention Is All You Needの図を参照

提案手法

計算量：従来手法とMemVPの計算量の比較

変数の定義
L：トークンシーケンスの長さ
d：トークンの次元
n：Visual Promptの長さ

- LMの計算量

$$FLOPs_{LM} = 4Ld(6d + L)$$

- Input-Space Visual Promptingの画像情報を加えることによって増加する計算量

$$FLOPs_{VP} - FLOPsLM = 4nd(6d + n + 2L)$$

- MemVPの画像情報を加えることによって増加する計算量

$$FLOPs_{MemVP} - FLOPs_{LM} = 4ndL$$

→現在のVLモデルは基本的にd>>nを満たし、ほとんどのタスクにおいてn>Lである。そのため、MemVPはVLタスクの計算効率が高い。

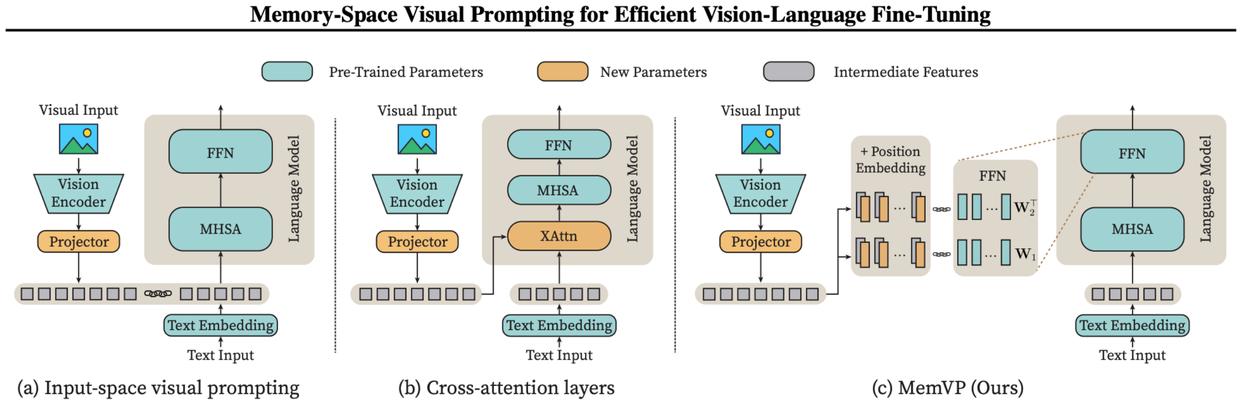


Figure 3. **Overview of the mainstream paradigms to concatenate vision encoder and language model. (a)** Concatenating visual prompts with the text tokens as inputs of the language model is not computation-efficient, *e.g.*, LLaVA, VL-Adapter, VL-PET. **(b)** Using cross-attention layers to incorporate the visual information from visual tokens is not parameter-efficient, *e.g.*, Flamingo, BLIP. **(c)** Our MemVP injects visual prompts into the FFN blocks of language models, achieving both parameter and computation efficiency.

Memory-Space Visual Prompting for Efficient Vision-Language Fine-Tuning

*LLaMA-7Bでトークン次元は4096

実験

実験１：Encoder Decoder(Bart base, T5 base)モデル

- データセット
  - Visual Question Answering(VQAv2, GQA), Image Captioning(COCO Captioning)
- ベースライン
  - BERTとT5をbaseとして以下でtuning
    - Input-Space Visual Promptingを使用したPEFT手法：VL-Adapter, VL-PET
    - 言語モデル向けに設計された代表的なPEFT手法：Compactor, LoRA
    - Input-Space Visual Promptingを使用したFully Fine-Turning
  - Visual featuresの取得
    - CLIPで事前学習されたVisual Encoder(ResNet101)のGAPの前のgrid featuresを単一のFC層を通じてVisual Promptを投影する

実験

実験１：Encoder Decoder(Bart base, T5 base)モデル

- 結果
  - 最新のPEFT手法であるVL-PETよりも平均パフォーマンスで優れている。(上図)
  - MemVPのFLOPsは他のベースラインの23~44%に抑えられている。(上図)
  - 訓練、推論時間が早く、訓練のメモリの使用量も抑えられている。(左下図)
  - MemVPでは、GAPの前の情報を用いているため、細かい局所情報を維持できている。(右下図)

Table 1. Results on VQAv2, GQA, and COCO Captions. "FLOPs" denotes the average FLOPs *in language models* on test set. We report average performance over three runs on Karpathy test split for VQAv2 and COCO Captions, and on test-dev split for GQA. All the baseline results are reproduced using the official code of VL-PET (Hu et al., 2023).

Method	#Trainable Params (M/task)	VQAv2 VQA Score	VQAv2 FLOPs (G)	GQA VQA Score	GQA FLOPs (G)	COCO Captions CIDEr	COCO Captions FLOPs (G)	Average Score
BART-base								
Full Fine-Tuning	141.16	65.4	4.8	53.1	5.3	110.6	6.4	76.4
Compacter	3.87	64.2	4.9	52.3	5.4	115.3	6.5	77.3
LoRA	3.92	64.8	4.8	52.2	5.3	115.1	6.4	77.4
VL-Adapter	3.87	**65.5**	4.9	53.7	5.4	114.3	6.5	77.8
VL-PET	3.84	65.3	5.0	53.9	5.5	**120.3**	6.6	79.8
MemVP (Ours)	**3.78**	65.2	**1.2**	**55.1**	**1.8**	120.2	**2.8**	**80.2**
T5-base								
Full Fine-Tuning	224.54	64.3	9.4	52.0	10.8	112.6	12.9	76.3
Compacter	6.11	65.5	9.6	53.6	11.0	113.4	13.2	77.5
LoRA	6.05	63.3	9.4	50.8	10.8	113.9	12.9	76.0
VL-Adapter	6.10	65.6	9.6	54.4	11.0	113.4	13.2	77.8
VL-PET	6.07	65.4	9.8	54.6	11.3	**121.2**	13.4	80.4
MemVP (Ours)	**6.00**	**65.7**	**2.3**	**56.0**	**3.8**	120.8	**5.8**	**80.8**

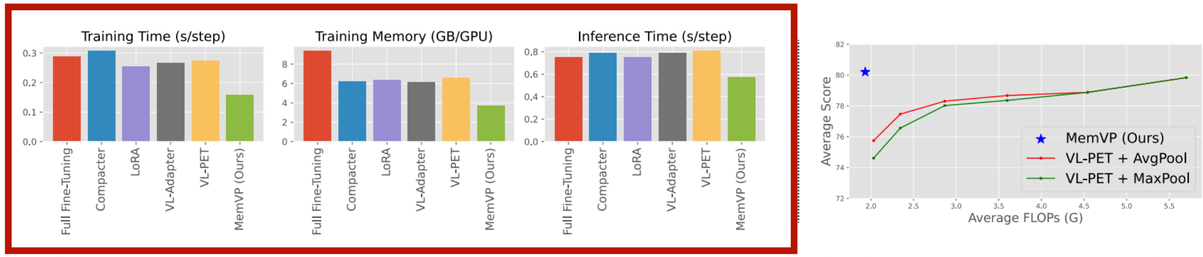


Figure 4. ***Left*: Training time, training memory, and inference time of T5-base on VQAv2.** The per-GPU batch sizes for training and inference are 64 and 512, respectively. Measured on V100 GPUs. ***Right*: Average score *vs*. FLOPs of BART-base on the three datasets.** The visual prompts of VL-PET are downsampled to reduce the input length.

実験

実験２：Decoder(LLaMA base)モデル

- データセット
  - VQA(ScienceQA：多様な知識分野から収集された科学質問応用データセット)
- ベースライン
  - LLaMAをbaseとしてtuning
    - LLaVA（LoRAありも比較する）
    - LLaMA-Adapter
    - LaVIN
  - Visual featuresの取得
    - CLIPで事前学習されたVisual Encoder(ViT-L/14)の最終層前のパッチ特徴量を非線形活性化関数を挟んだ２つのFC層を通じてVisual Promptに投影する

実験

実験２：Decoder(LLaMA base)モデル

- 結果
  - MemVPはLLaMA-7B, 13Bの両方でベースラインのPEFT手法を大きく上回っている(表2)
  - LLaVAのフルファインチューニングモデルを7/8のサブセットでうわまっている(表2)
  - Memory-Space Visual Promptingは計算効率を高めている。(表3)

Table 3. **Training and inference time.** Measured on 8×A800 GPUs without memory-saving or speed-up techniques (*e.g.,* flash attention). The per-GPU batch size is 4 for training and 64 for inference.

Method	Length of Visual Prompt	#Trainable Params	Training Time (s/batch)	Inference Time (s/batch)
LLaVA-LoRA 7B	256	4.4M	0.49	3.42
LaVIN 7B	6	3.8M	0.39	2.06
MemVP 7B	256	3.9M	**0.28**	**1.88**
MemVP 13B	256	5.5M	0.46	3.07

Table 2. **Accuracy on ScienceQA test set.** Question categories: NAT = natural science, SOC = social science, LAN = language science, TXT = w/ text context, IMG = w/ image context, NO = no context, G1-6 = grades 1-6, G7-12 = grades 7-12. † denotes our reproduced results. Other results are quoted from their original papers.

Method	#Trainable Params	Language Model	VL Pre-Train	Subject			Context Modality			Grade		Average
				NAT	SOC	LAN	TXT	IMG	NO	G1-6	G7-12	
Human	-	-	-	90.23	84.97	87.48	89.60	87.50	88.10	91.59	82.42	88.40
GPT-4 (0-shot)	-	GPT-4	-	84.06	73.45	87.36	81.87	70.75	90.73	84.69	79.10	82.69
LLaVA	7B	Vicuna-7B	√	-	-	-	-	-	-	-	-	89.84
LLaVA	13B	Vicuna-13B	×	-	-	-	-	-	-	-	-	85.81
LLaVA	13B	Vicuna-13B	√	90.36	**95.95**	88.00	89.49	88.00	90.66	90.93	**90.90**	90.92
PEFT methods												
LLaMA-Adapter	1.8M	LLaMA-7B	×	84.37	88.30	84.36	83.72	80.32	86.90	85.83	84.05	85.19
LLaVA-LoRA†	4.4M	LLaMA-7B	×	91.70	94.60	86.09	91.25	90.28	88.64	91.52	89.65	90.85
LaVIN	3.8M	LLaMA-7B	×	89.25	94.94	85.24	88.51	87.46	88.08	90.16	88.07	89.41
MemVP (Ours)	3.9M	LLaMA-7B	×	94.45	95.05	88.64	93.99	92.36	90.94	93.10	93.01	93.07
LaVIN	5.4M	LLaMA-13B	×	90.32	94.38	87.73	89.44	87.65	90.31	91.19	89.26	90.50
MemVP (Ours)	5.5M	LLaMA-13B	×	**95.07**	**95.15**	**90.00**	**94.43**	**92.86**	**92.47**	**93.61**	**94.07**	**93.78**

LLaVAのフルチューニングよりも良い

Memory-Space Visual Prompting for Efficient Vision-Language Fine-Tuning

実験

実験２：Decoder(LLaMA base)モデル

アブレーション実験：各要素の有用性を示す。

- Visual Promptを追加せずに位置埋め込みのみ
  - IMGのサブセットで大幅に性能低下
- LaVINのようにグローバル特徴量を使用
  - ローカル情報の喪失による性能低下
- 位置埋め込みを追加する代わりに結合する
  - 性能の低下（位置情報が重要）
- Key, Valueの一方のみ
  - 両方のケースで性能低下

Table 4. **Ablation experiments on ScienceQA.** "Average" and "IMG" denote the accuracy on the whole test set and on the IMG subset, respectively.

Settings	Average	IMG	#Trainable Params (M)
MemVP 7B	**93.07**	**92.36**	3.9
w/o visual prompts	85.33	76.05	3.3
visual features: local → global	89.01	84.18	3.9
position embedding: add → concat	89.79	86.07	3.9
insert visual prompts in keys only	91.94	90.23	3.9
insert visual prompts in values only	92.78	**92.36**	3.9

Memory-Space Visual Prompting for Efficient Vision-Language Fine-Tuning

実験

実験３：FFN層の視覚情報の認識

Visual Informationが言語モデルのメモリに外部知識として挿入できることを以下で検証する(BARTbaseのモデル)。

- テキスト入力に関連するVisual knowledgeが取得されるか。
  - 方法：QueryとKeyの類似性$\phi(\langle x, \lambda f(z_i)\rangle)$を可視化する。
  - 結果：Text tokenが関連するVisual knowledge入力のキーと高い類似性を持つことがわかり、対応される値が取得されていることを示唆している。
- 手動でQueryとKeyの類似性を操作した場合、モデルが誤った内容を出力するか。
  - 方法：応答性の高い２つのエントリの取得を手動でブロックして$\phi(\langle x, \lambda f(z_i)\rangle) = 0$と設定する。
  - 結果：誤った結果を出力した。

→視覚情報が実際にメモリに挿入され、言語モデルの出力を直接指示することを確認できる。

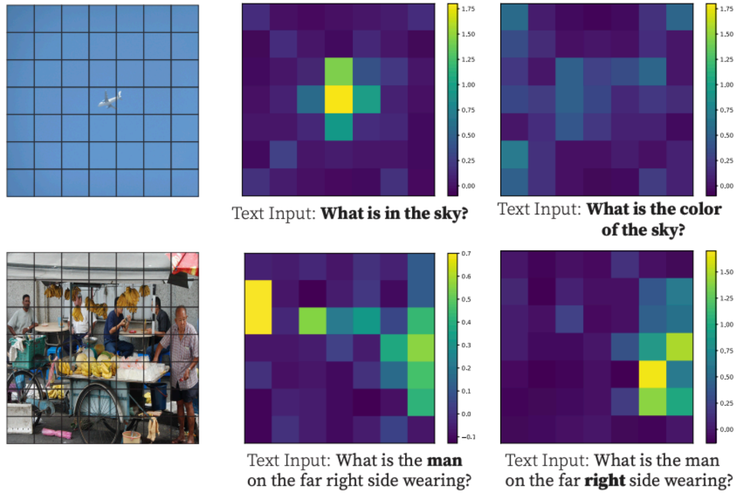


Figure 5. **Visual knowledge locating.** The similarity values between **blod** text tokens and keys of visual knowledge are averaged over all layers.

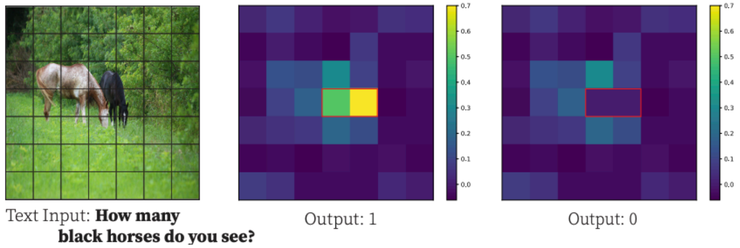


Figure 6. **Visual knowledge distortion.** ***Left:*** Inputs of model; ***Middle:*** Original similarity between text tokens and keys of visual knowledge; ***Right:*** Distorted similarity. The values in the red rectangle are set to 0.

Memory-Space Visual Prompting for Efficient Vision-Language Fine-Tuning

結論

- 従来のVisual PromptingとPEFTはパラメータの面では効率的であるものの計算効率は高くない。
- そこで、FFN層のMemory-SpaceにVisual Promptingを挿入する手法が提案された。
- その結果、従来手法に匹敵する性能と高速な学習、推論、低メモリーの学習を実現した。
- 制約として長文生成の場合は、推論速度の利点は減少する。

参考文献

- 本JCの題材：Memory-Space Visual Prompting for Efficient Vision-Language Fine-Tuning
- CLIP原論文：Learning Transferable Visual Models From Natural Language Supervision
- Transformerの原論文：Attention is All You Need
- Vision and Languageの応用タスク：DenseCap: Fully Convolutional Localization Networks for Dense Captioning, Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering
- FFN層のKey-Valueメモリー：Transformer Feed-Forward Layers Are Key-Value Memories