Visual Prompt Tuning

2026.09.04

Data Mining & Quality Analytics Lab.

김윤아

발표자 소개

❖ 김윤아 (Yoona Kim)

- 고려대학교 산업경영공학과 석사과정 (2026.09 ~ Present)
- Data Mining & Quality Analysis Lab. (김성범 교수님)

❖ Research Interest

- Vision-Language Models

❖ Contact

- 2022130818@korea.ac.kr

Introduction

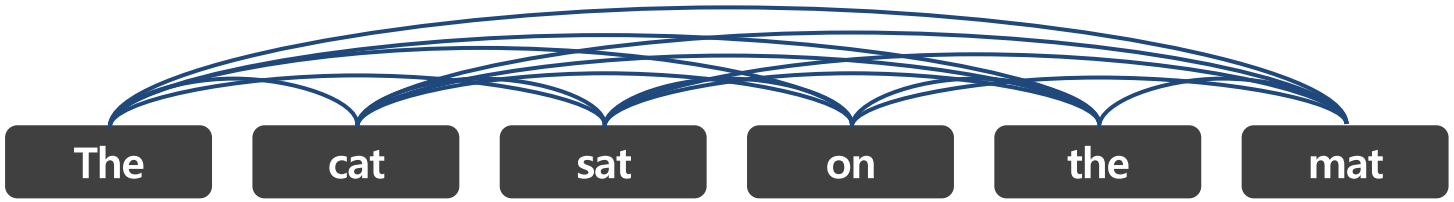
Transformer

❖ RNN 구조란?

- 문장 내 단어들을 순차적으로 처리 → 먼 단어 관계 포착에 불리

❖ Transformer 구조란?

- Self-Attention을 통해 문장 내 모든 단어 간의 상호 관계를 한 번에 계산 → 거리와 무관하게 문맥 포착



Provided proper attribution is provided, Google hereby grants permission to reproduce the tables and figures in this paper solely for use in journalistic or scholarly works.

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

*Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and efficient inference and visualizations. Lukasz and Aidan spent countless long days designing various parts of and implementing tensor2tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research.

†Work performed while at Google Brain.

‡Work performed while at Google Research.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

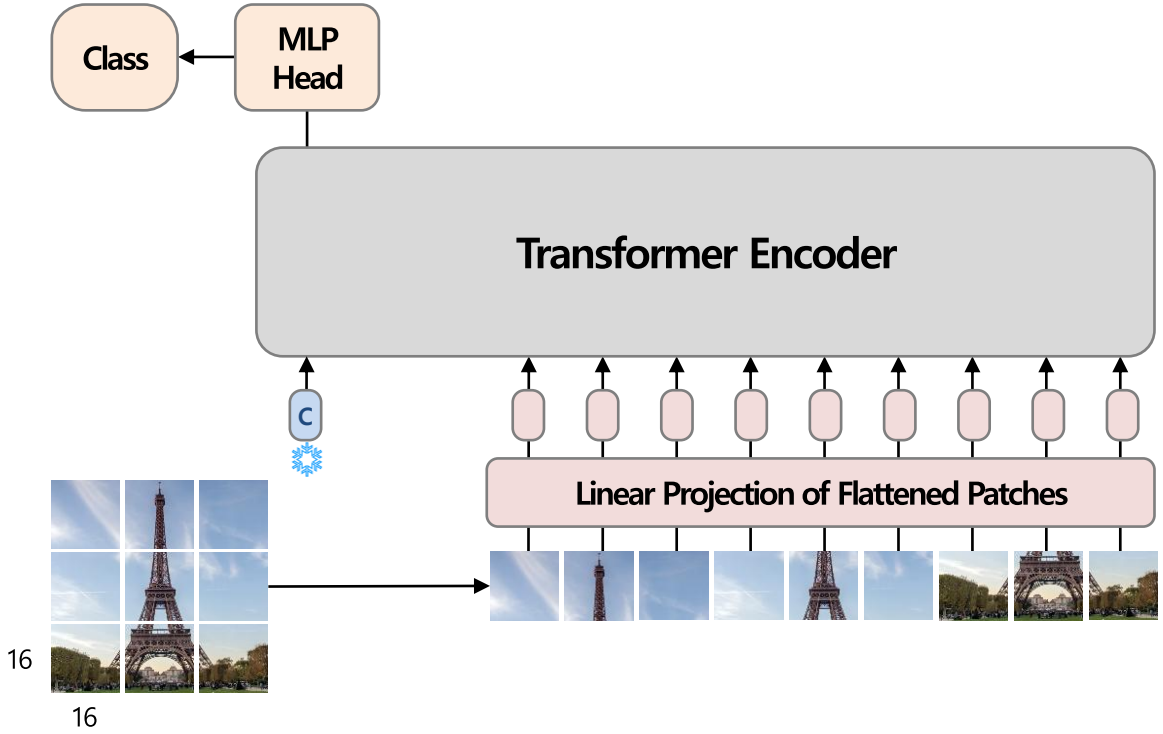
Attention is All You Need
(Vaswani et al., NeurIPS 2017)

Introduction

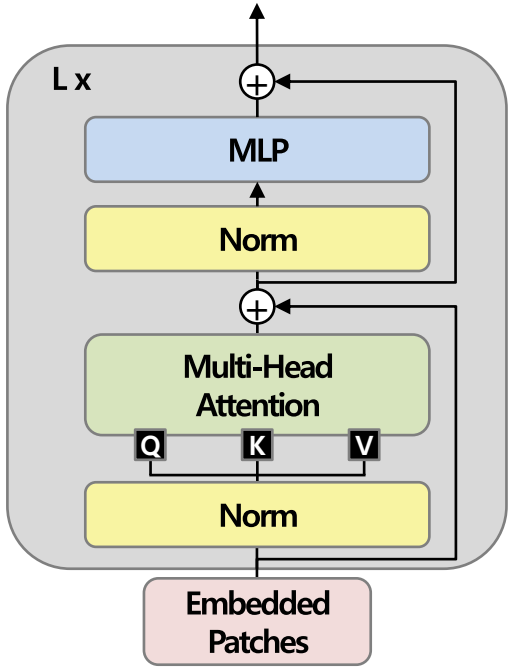
Vision Transformer (ViT)

❖ Vision Transformer (ViT) 구조란?

- 이미지를 16×16 크기의 patch로 나눈 뒤 Transformer Encoder를 통해 patch 간 관계를 학습하고, [CLS] token을 이용해 최종 이미지를 분류



Vision Transformer (ViT)



Transformer Encoder

Published as a conference paper at ICLR 2021

AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

Alexey Dosovitskiy*,†, Lucas Beyer*, Alexander Kolesnikov*, Dirk Weissenborn*, Xiaohua Zhai*, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby*,†

*equal technical contribution, †equal advising
Google Research, Brain Team
{adosovitskiy, neilhoulsby}@google.com

ABSTRACT

While the Transformer architecture has become the de-facto standard for natural language processing tasks, its applications to computer vision remain limited. In vision, attention is either applied in conjunction with convolutional networks, or used to replace certain components of convolutional networks while keeping their overall structure in place. We show that this reliance on CNNs is not necessary and a pure transformer applied directly to sequences of image patches can perform very well on image classification tasks. When pre-trained on large amounts of data and transferred to multiple mid-sized or small image recognition benchmarks (ImageNet, CIFAR-100, VTAB, etc.), Vision Transformer (ViT) attains excellent results compared to state-of-the-art convolutional networks while requiring substantially fewer computational resources to train.[1]

1 INTRODUCTION

Self-attention-based architectures, in particular Transformers (Vaswani et al., 2017), have become the model of choice in natural language processing (NLP). The dominant approach is to pre-train on a large text corpus and then fine-tune on a smaller task-specific dataset (Devlin et al., 2019). Thanks to Transformers' computational efficiency and scalability, it has become possible to train models of unprecedented size, with over 100B parameters (Brown et al., 2020; Lepikhin et al., 2020). With the models and datasets growing, there is still no sign of saturating performance.

In computer vision, however, convolutional architectures remain dominant (LeCun et al., 1989; Krizhevsky et al., 2012; He et al., 2016). Inspired by NLP successes, multiple works try combining CNN-like architectures with self-attention (Wang et al., 2018; Carion et al., 2020), some replacing the convolutions entirely (Ramachandran et al., 2019; Wang et al., 2020a). The latter models, while theoretically efficient, have not yet been scaled effectively on modern hardware accelerators due to the use of specialized attention patterns. Therefore, in large-scale image recognition, classic ResNet-like architectures are still state of the art (Mahajan et al., 2018; Xie et al., 2020; Kolesnikov et al., 2020).

Inspired by the Transformer scaling successes in NLP, we experiment with applying a standard Transformer directly to images, with the fewest possible modifications. To do so, we split an image into patches and provide the sequence of linear embeddings of these patches as an input to a Transformer. Image patches are treated the same way as tokens (words) in an NLP application. We train the model on image classification in supervised fashion.

When trained on mid-sized datasets such as ImageNet without strong regularization, these models yield modest accuracies of a few percentage points below ResNets of comparable size. This seemingly discouraging outcome may be expected: Transformers lack some of the inductive biases

[1]Fine-tuning code and pre-trained models are available at https://github.com/google-research/vision_transformer

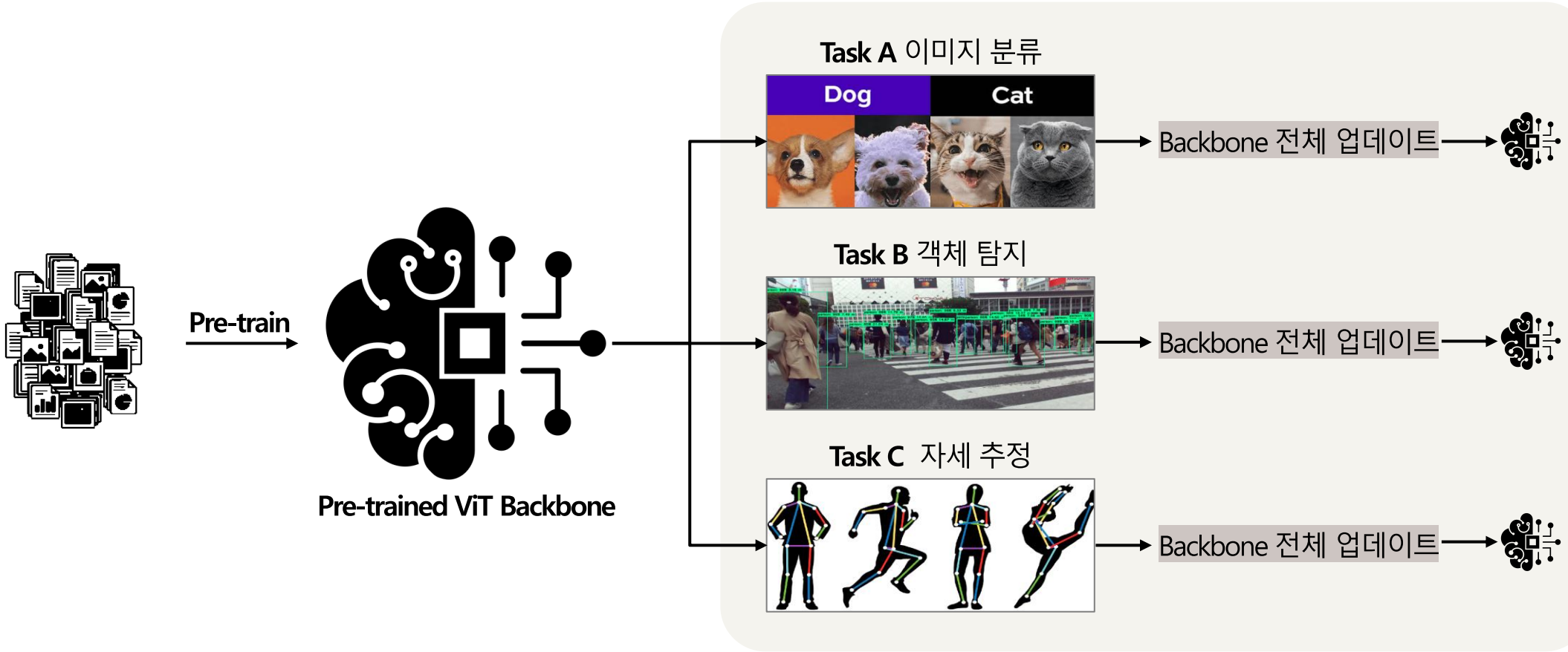
1

arXiv:2010.11929v2 [cs.CV] 3 Jun 2021

An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale
(Dosovitskiy et al., ICLR 2021)

Introduction

Pretrain-then-Finetune Paradigm



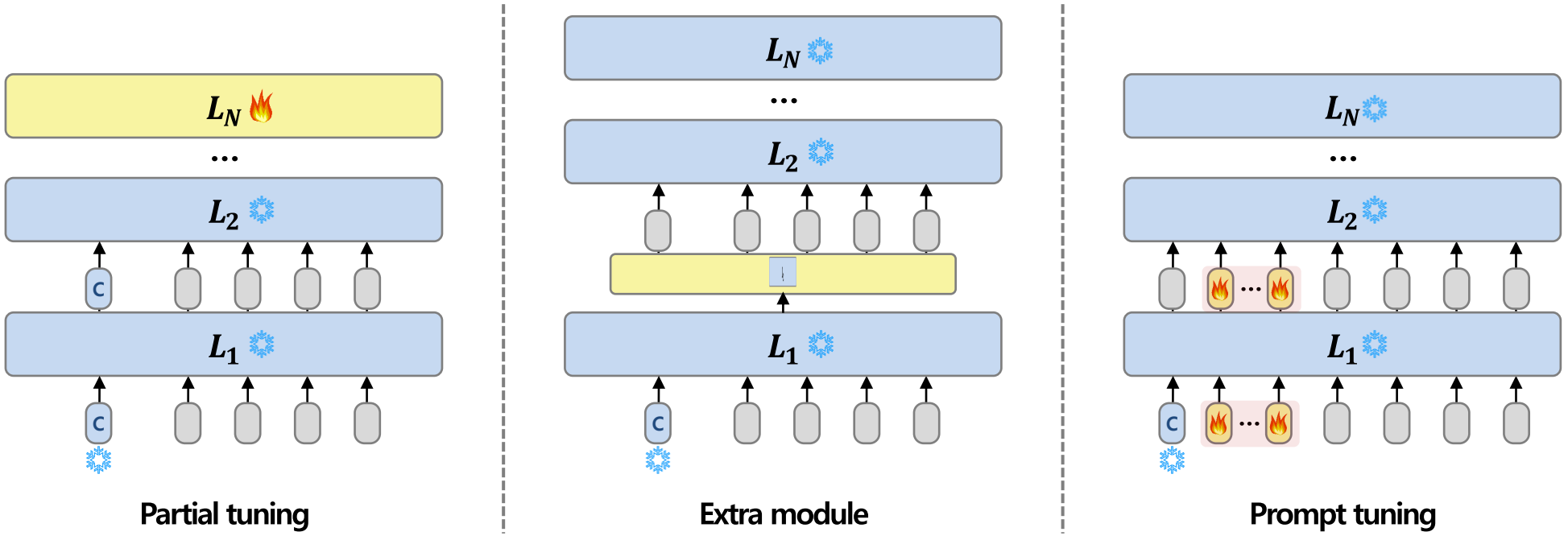
Full Fine-Tuning: Task마다 전체 backbone을 업데이트·저장 → 학습 및 저장 비용 증가
→ Pre-trained backbone은 공유하고 일부 parameter만 학습하는 **Parameter-Efficient Fine-Tuning (PEFT)** 필요

Introduction

Parameter-Efficient Fine-Tuning (PEFT)

❖ **PEFT의 세 가지 접근 방식: 어디를 학습할 것인가?**

- **Partial tuning**: 백본의 일부 파라미터만 업데이트 → 백본 가중치가 변형되므로 다중 태스크 환경에서 공유 · 재사용이 어려움
- **Extra module**: 백본은 동결, 작은 task-specific module만 학습 → 구조 의존성과 추가 연산량 발생
- **Prompt tuning**: 백본은 동결, 입력에 learnable prompt token 추가 → 구조 수정 없이 적은 파라미터만으로 효율적인 task adaptation 가능

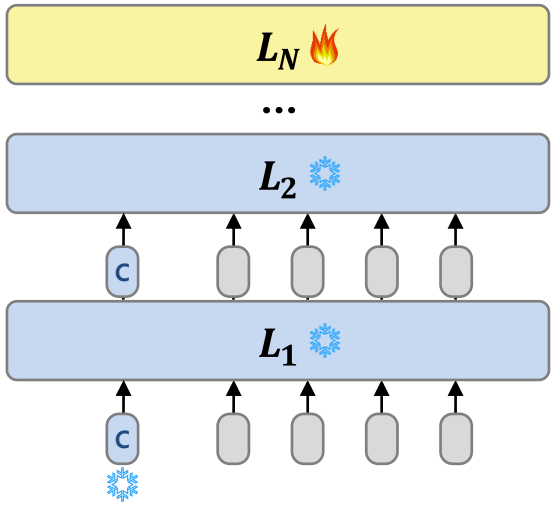


Introduction

Parameter-Efficient Fine-Tuning (PEFT)

❖ **PEFT의 세 가지 접근 방식: 어디를 학습할 것인가?**

- **Partial tuning**: 백본의 일부 파라미터만 업데이트 → 백본 가중치가 변형되므로 다중 태스크 환경에서 공유 · 재사용이 어려움
- **Extra module**: 백본은 동결, 작은 task-specific module만 학습 → 구조 의존성과 추가 연산량 발생
- **Prompt tuning**: 백본은 동결, 입력에 learnable prompt token 추가 → 구조 수정 없이 적은 파라미터만으로 효율적인 task adaptation 가능



Partial tuning

Extra module

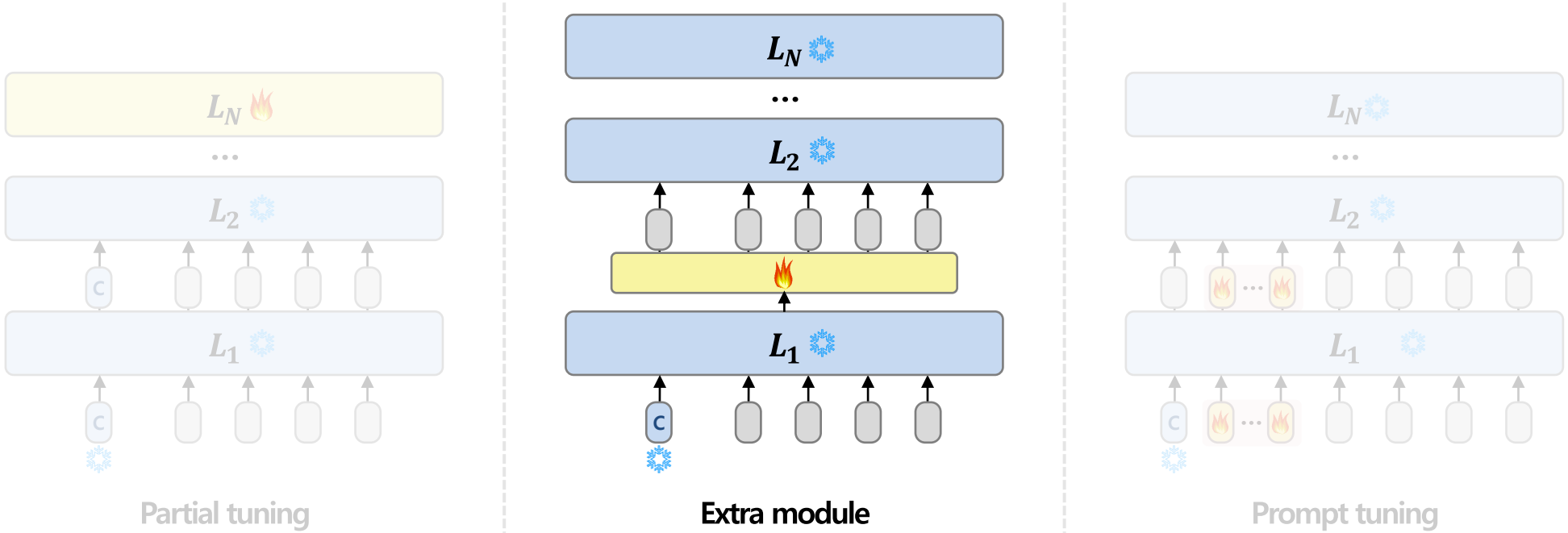
Prompt tuning

Introduction

Parameter-Efficient Fine-Tuning (PEFT)

❖ **PEFT의 세 가지 접근 방식: 어디를 학습할 것인가?**

- **Partial tuning**: 백본의 일부 파라미터만 업데이트 → 백본 가중치가 변형되므로 다중 태스크 환경에서 공유 · 재사용이 어려움
- **Extra module**: 백본은 동결, 작은 task-specific module만 학습 → 구조 의존성과 추가 연산량 발생
- **Prompt tuning**: 백본은 동결, 입력에 learnable prompt token 추가 → 구조 수정 없이 적은 파라미터만으로 효율적인 task adaptation 가능

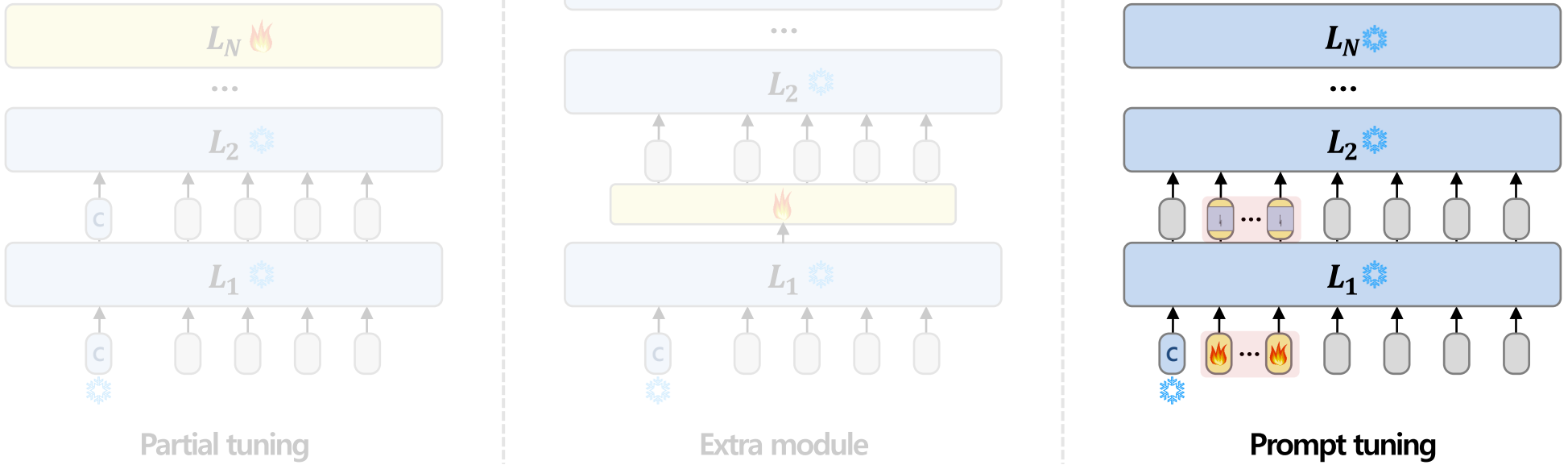


Introduction

Parameter-Efficient Fine-Tuning (PEFT)

❖ **PEFT의 세 가지 접근 방식: 어디를 학습할 것인가?**

- **Partial tuning**: 백본의 일부 파라미터만 업데이트 → 백본 가중치가 변형되므로 다중 태스크 환경에서 공유 · 재사용이 어려움
- **Extra module**: 백본은 동결, 작은 task-specific module만 학습 → 구조 의존성과 추가 연산량 발생
- **Prompt tuning**: 백본은 동결, 입력에 learnable prompt token 추가 → 구조 수정 없이 적은 파라미터만으로 효율적인 task adaptation 가능

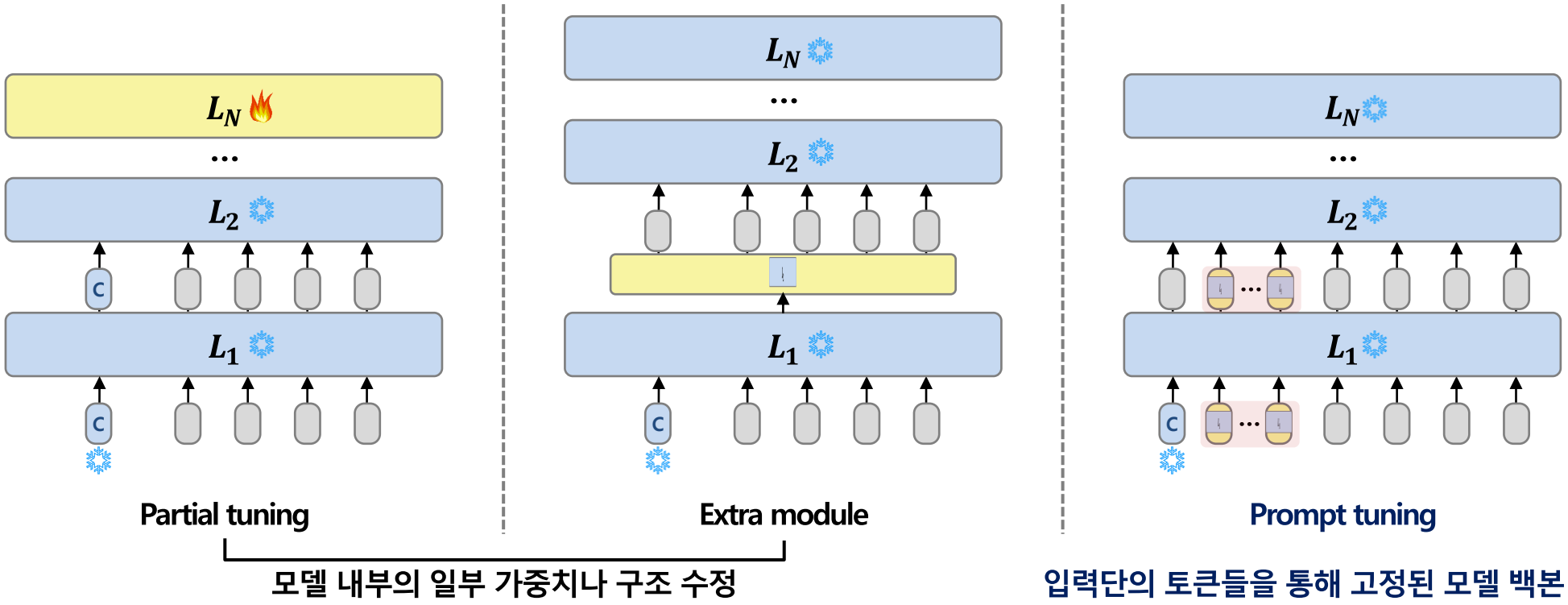


Introduction

Parameter-Efficient Fine-Tuning (PEFT)

❖ **PEFT의 세 가지 접근 방식: 어디를 학습할 것인가?**

- **Partial tuning**: 백본의 일부 파라미터만 업데이트 → 백본 가중치가 변형되므로 다중 태스크 환경에서 공유 · 재사용이 어려움
- **Extra module**: 백본은 동결, 작은 task-specific module만 학습 → 구조 의존성과 추가 연산량 발생
- **Prompt tuning**: 백본은 동결, 입력에 learnable prompt token 추가 → 구조 수정 없이 적은 파라미터만으로 효율적인 task adaptation 가능

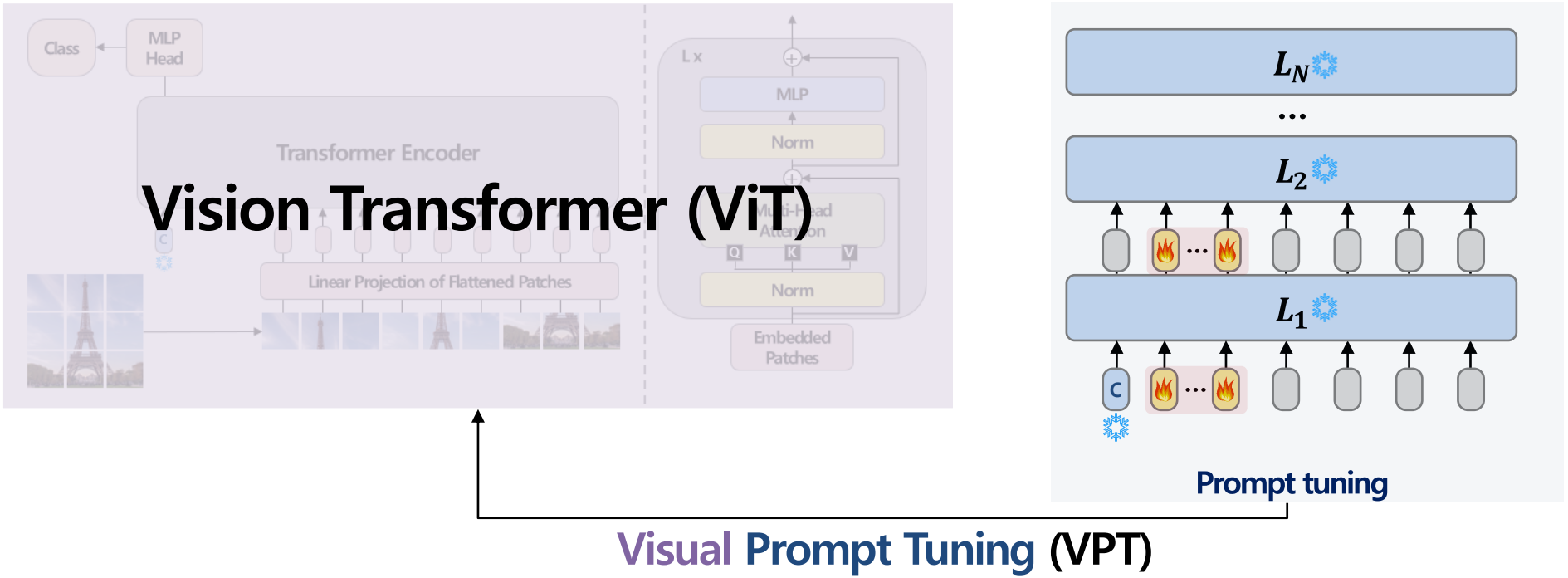


Introduction

Parameter-Efficient Fine-Tuning (PEFT)

❖ **PEFT의 세 가지 접근 방식: 어디를 학습할 것인가?**

- **Partial tuning**: 백본의 일부 파라미터만 업데이트 → 백본 가중치가 변형되므로 다중 태스크 환경에서 공유 · 재사용이 어려움
- **Extra module**: 백본은 동결, 작은 task-specific module만 학습 → 구조 의존성과 추가 연산량 발생
- **Prompt tuning**: 백본은 동결, 입력에 learnable prompt token 추가 → 구조 수정 없이 적은 파라미터만으로 효율적인 task adaptation 가능

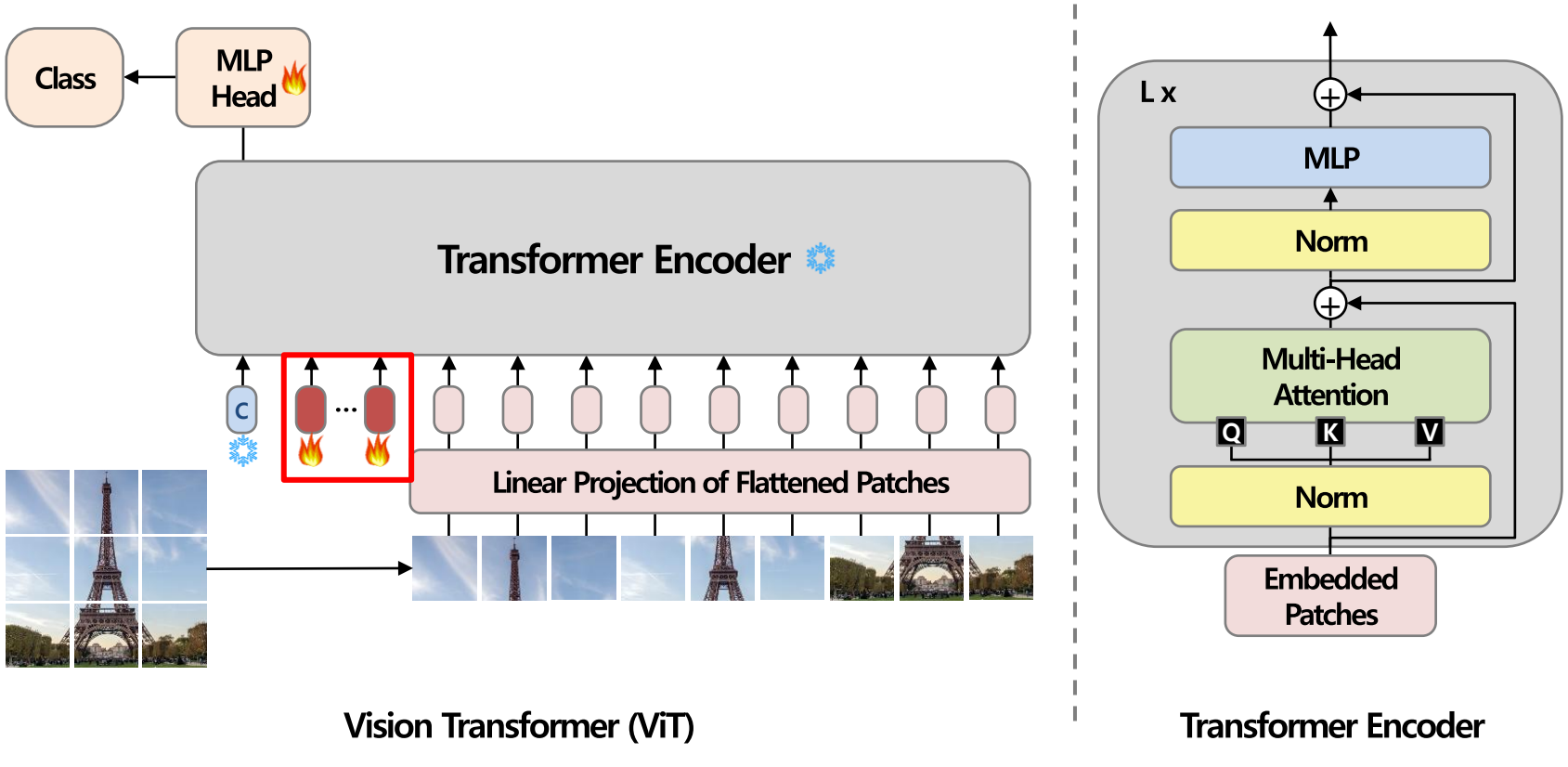


Visual Prompt Tuning (VPT)

How VPT Works

❖ **Visual Prompt Tuning (VPT)의 구조와 학습 방식은 어떻게 될까?**

- [CLS] 토큰과 패치 임베딩 사이에 **prompt token**을 삽입 → 이미지에서 추출된 정보가 아닌, 태스크에 맞춰 최적화되는 유도용 파라미터
- 백본은 동결, prompt token과 classification head만 학습



Vision Transformer (ViT)

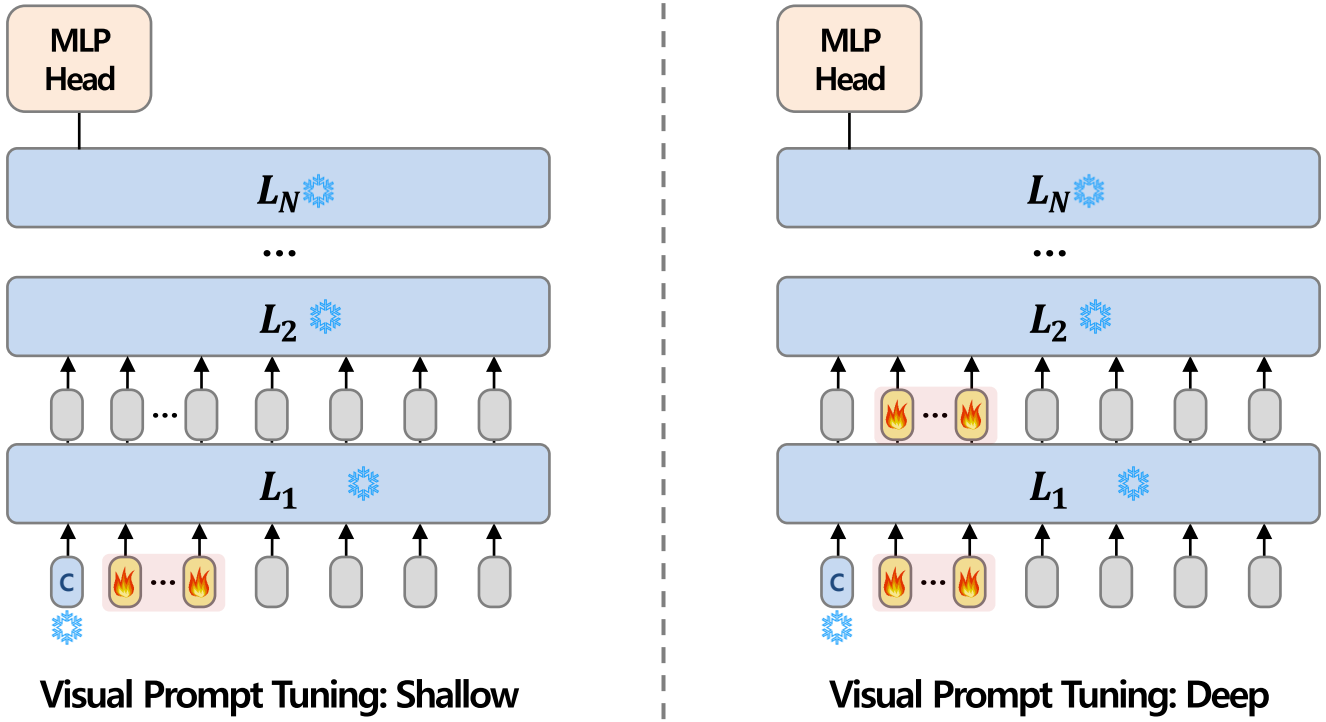
Transformer Encoder

Visual Prompt Tuning (VPT)

How VPT Works

❖ **VPT의 두 가지 Prompt 삽입 방식은 무엇일까?**

- **VPT-Shallow:** 첫 번째 layer 입력단에만 prompt 삽입 → 단일 prompt set만 학습, 매우 가벼움
- **VPT-Deep:** 모든 layer 입력마다 서로 다른 prompt 삽입 → layer별 task-specific 정보 반영, 표현력 향상



Visual Prompt Tuning: Shallow

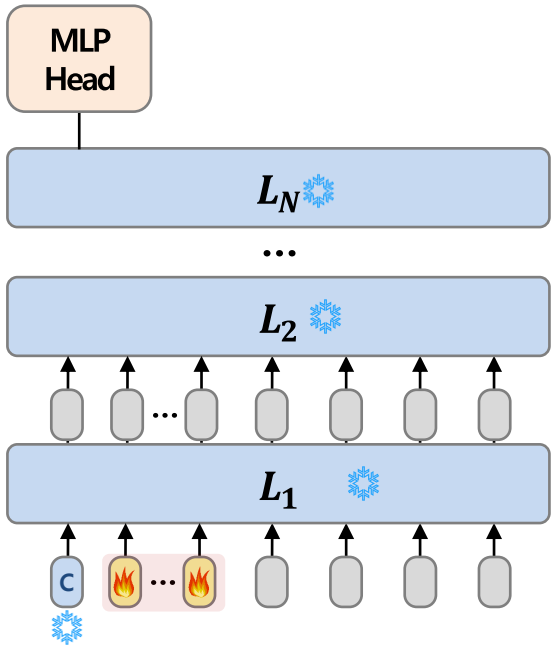
Visual Prompt Tuning: Deep

Visual Prompt Tuning (VPT)

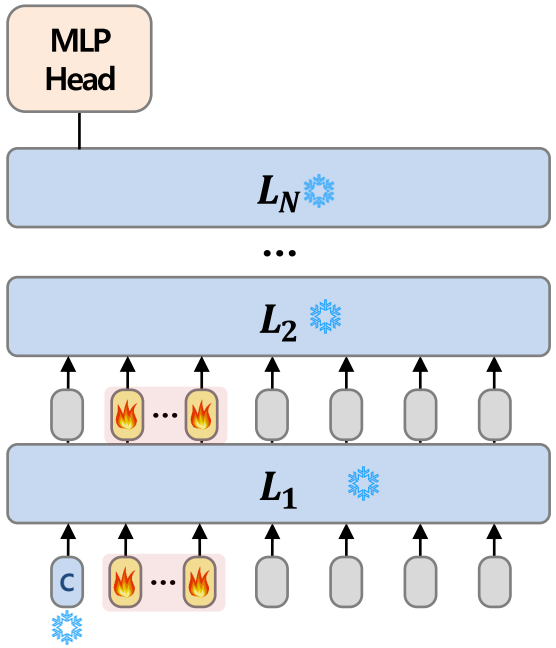
How VPT Works

❖ **VPT의 두 가지 Prompt 삽입 방식은 무엇일까?**

- **VPT-Shallow:** 첫 번째 layer 입력단에만 prompt 삽입 → 단일 prompt set만 학습, 매우 가벼움
- **VPT-Deep:** 모든 layer 입력마다 서로 다른 prompt 삽입 → layer별 task-specific 정보 반영, 표현력 향상



Visual Prompt Tuning: Shallow



Visual Prompt Tuning: Deep

	추가 파라미터	비율
VPT-Shallow	0.038M	0.04%
VPT-Deep	0.46M	0.53%

Visual Prompt Tuning (VPT)

Experiments

❖ VPT Experiment #1: Parameter Efficiency & Performance

- **비교 대상:** (a) Full fine-tuning / (b) Partial tuning / (c) Extra module
- **저장 비용:** Full fine-tuning **24.02×** ↔ VPT-Shallow **1.04×**, VPT-Deep **1.18×**
- **성능:** VPT-Deep은 **24개 task 중 20개에서 Full Fine-Tuning보다 높은 성능**

	ViT-B/16 (85.8M)	Total params	Scope: Input	Scope: Backbone	Extra params	FGVC	VTAB-1k: Natural	VTAB-1k: Specialized	VTAB-1k: Structured
	Total # of tasks					5	7	4	8
(a)	FULL	24.02×		✓		88.54	75.88	83.36	47.64
(b)	LINEAR	1.02×				79.32 (0)	68.93 (1)	77.16 (1)	26.84 (0)
	PARTIAL-1	3.00×				82.63 (0)	69.44 (2)	78.53 (0)	34.17 (0)
	MLP-3	1.35×			✓	79.80 (0)	67.80 (2)	72.83 (0)	30.62 (0)
(c)	SIDETUNE	3.69×		✓	✓	78.35 (0)	58.21 (0)	68.12 (0)	23.41 (0)
	BIAS	1.05×		✓		88.41 (3)	73.30 (3)	78.25 (0)	44.09 (2)
	ADAPTER	1.23×		✓	✓	85.66 (2)	70.39 (4)	77.11 (0)	33.43 (0)
(ours)	VPT-SHALLOW	1.04×	✓		✓	84.62 (1)	76.81 (4)	79.66 (0)	46.98 (4)
	VPT-DEEP	1.18×				**89.11 (4)**	**78.48 (6)**	**82.43 (2)**	**54.98 (8)**

Visual Prompt Tuning (VPT)

Experiments

❖ VPT Experiment #1: Parameter Efficiency & Performance

- **비교 대상:** (a) Full fine-tuning / (b) Partial tuning / (c) Extra module
- **저장 비용:** Full fine-tuning **24.02×** ↔ VPT-Shallow **1.04×**, VPT-Deep **1.18×**
- **성능:** VPT-Deep은 **24개 task 중 20개에서 Full Fine-Tuning보다 높은 성능**

	ViT-B/16 (85.8M)	Total params	Scope Input	Scope Backbone	Extra params	FGVC	VTAB-1k Natural	VTAB-1k Specialized	VTAB-1k Structured
	Total # of tasks					5	7	4	8
(a)	FULL	24.02×		✓		88.54	75.88	83.36	47.64
(b)	LINEAR	1.02×				79.32 (0)	68.93 (1)	77.16 (1)	26.84 (0)
	PARTIAL-1	3.00×				82.63 (0)	69.44 (2)	78.53 (0)	34.17 (0)
	MLP-3	1.35×			✓	79.80 (0)	67.80 (2)	72.83 (0)	30.62 (0)
(c)	SIDETUNE	3.69×		✓	✓	78.35 (0)	58.21 (0)	68.12 (0)	23.41 (0)
	BIAS	1.05×		✓		88.41 (3)	73.30 (3)	78.25 (0)	44.09 (2)
	ADAPTER	1.23×		✓	✓	85.66 (2)	70.39 (4)	77.11 (0)	33.43 (0)
(ours)	VPT-SHALLOW	1.04×	✓		✓	84.62 (1)	76.81 (4)	79.66 (0)	46.98 (4)
	VPT-DEEP	1.18×				**89.11 (4)**	**78.48 (6)**	**82.43 (2)**	**54.98 (8)**

Visual Prompt Tuning (VPT)

Experiments

❖ VPT Experiment #1: Parameter Efficiency & Performance

- **비교 대상:** (a) Full fine-tuning / (b) Partial tuning / (c) Extra module
- **저장 비용:** Full fine-tuning **24.02×** ↔ VPT-Shallow **1.04×**, VPT-Deep **1.18×**
- **성능:** VPT-Deep은 **24개 task 중 20개에서 Full Fine-Tuning보다 높은 성능**

	ViT-B/16 (85.8M)	Total params	Scope: Input	Scope: Backbone	Extra params	FGVC	VTAB-1k: Natural	VTAB-1k: Specialized	VTAB-1k: Structured
	Total # of tasks					5	7	4	8
(a)	FULL	24.02×		✓		88.54	75.88	83.36	47.64
(b)	LINEAR	1.02×				79.32 (0)	68.93 (1)	77.16 (1)	26.84 (0)
	PARTIAL-1	3.00×				82.63 (0)	69.44 (2)	78.53 (0)	34.17 (0)
	MLP-3	1.35×			✓	79.80 (0)	67.80 (2)	72.83 (0)	30.62 (0)
(c)	SIDETUNE	3.69×		✓	✓	78.35 (0)	58.21 (0)	68.12 (0)	23.41 (0)
	BIAS	1.05×		✓		88.41 (3)	73.30 (3)	78.25 (0)	44.09 (2)
	ADAPTER	1.23×		✓	✓	85.66 (2)	70.39 (4)	77.11 (0)	33.43 (0)
(ours)	VPT-SHALLOW	1.04×	✓		✓	84.62 (1)	76.81 (4)	79.66 (0)	46.98 (4)
	VPT-DEEP	1.18×				**89.11 (4)**	**78.48 (6)**	**82.43 (2)**	**54.98 (8)**

Visual Prompt Tuning (VPT)

Experiments

❖ VPT Experiment #1: Parameter Efficiency & Performance

- **비교 대상:** (a) Full fine-tuning / (b) Partial tuning / (c) Extra module
- **저장 비용:** Full fine-tuning **24.02×** ↔ VPT-Shallow **1.04×**, VPT-Deep **1.18×**
- **성능:** VPT-Deep은 **24개 task 중 20개에서 Full Fine-Tuning보다 높은 성능**

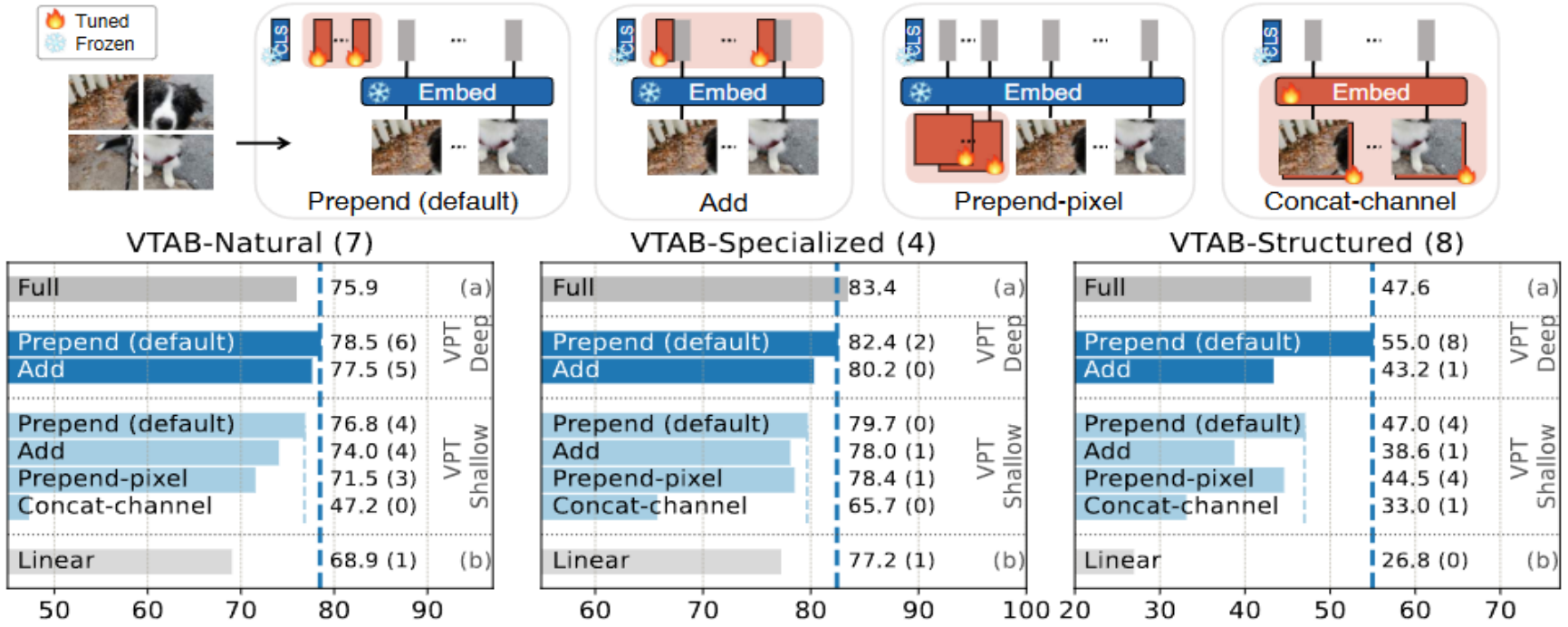
	ViT-B/16 (85.8M)	Total params	Scope Input	Scope Backbone	Extra params	FGVC	VTAB-1k Natural	VTAB-1k Specialized	VTAB-1k Structured
	Total # of tasks					5	7	4	8
(a)	FULL	24.02×		✓		88.54	75.88	83.36	47.64
(b)	LINEAR	1.02×				79.32 (0)	68.93 (1)	77.16 (1)	26.84 (0)
	PARTIAL-1	3.00×				82.63 (0)	69.44 (2)	78.53 (0)	34.17 (0)
	MLP-3	1.35×			✓	79.80 (0)	67.80 (2)	72.83 (0)	30.62 (0)
(c)	SIDETUNE	3.69×		✓	✓	78.35 (0)	58.21 (0)	68.12 (0)	23.41 (0)
	BIAS	1.05×		✓		88.41 (3)	73.30 (3)	78.25 (0)	44.09 (2)
	ADAPTER	1.23×		✓	✓	85.66 (2)	70.39 (4)	77.11 (0)	33.43 (0)
(ours)	VPT-SHALLOW	1.04×	✓		✓	84.62 (1)	76.81 (4)	79.66 (0)	46.98 (4)
	VPT-DEEP	1.18×				**89.11 (4)**	**78.48 (6)**	**82.43 (2)**	**54.98 (8)**

Visual Prompt Tuning (VPT)

Experiments

❖ VPT Experiment #2: Ablation on Model Design Variants ① Prompt Location

- **Embedding-space의 Prepend 방식**이 가장 안정적인 성능을 보였으며, **Add 및 pixel/channel-level 방식**은 상대적으로 성능이 저하됨

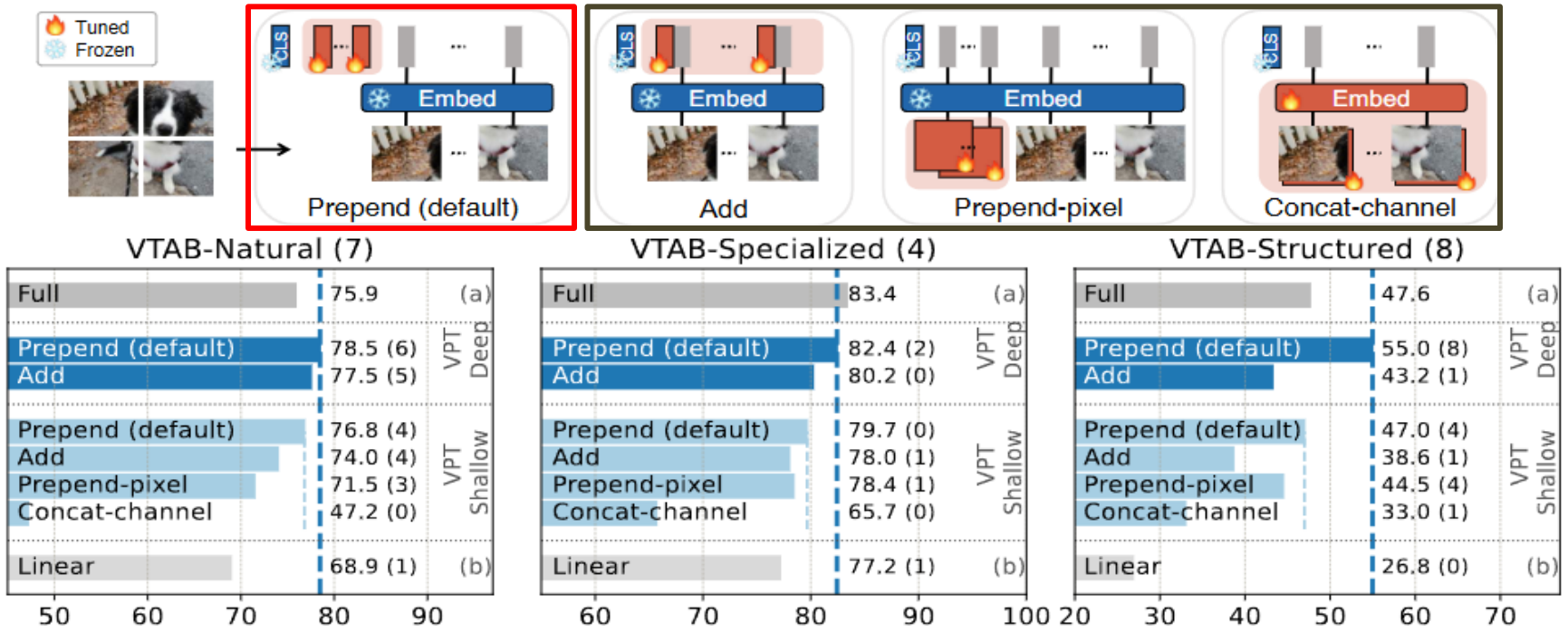


Visual Prompt Tuning (VPT)

Experiments

❖ VPT Experiment #2: Ablation on Model Design Variants ① Prompt Location

- **Embedding-space의 Prepend 방식**이 가장 안정적인 성능을 보였으며, **Add 및 pixel/channel-level 방식**은 상대적으로 성능이 저하됨

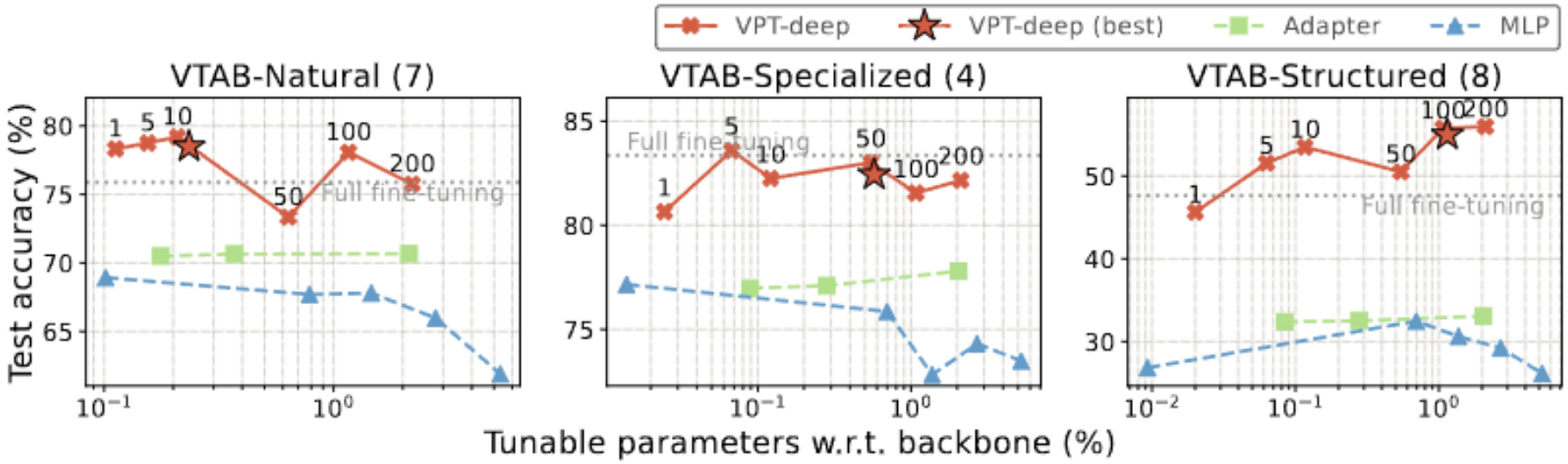


Visual Prompt Tuning (VPT)

Experiments

❖ **VPT Experiment #2: Ablation on Model Design Variants ② Prompt Length**

- Prompt length 증가가 성능 향상을 보장하지 않으며, 최적 prompt length는 Natural 10, Specialized 50, Structured 100으로 서로 다름

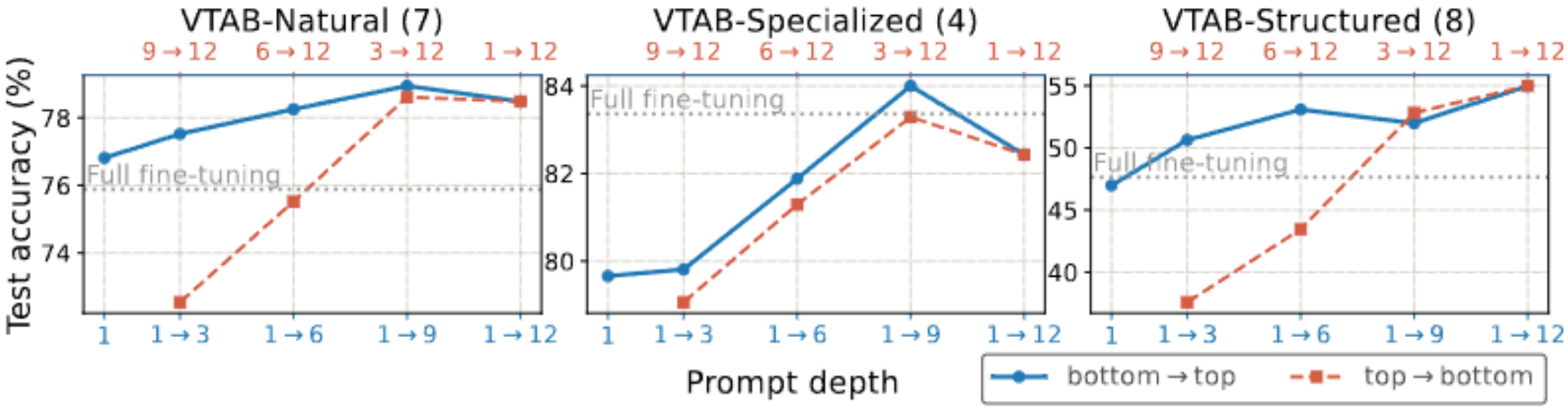


Visual Prompt Tuning (VPT)

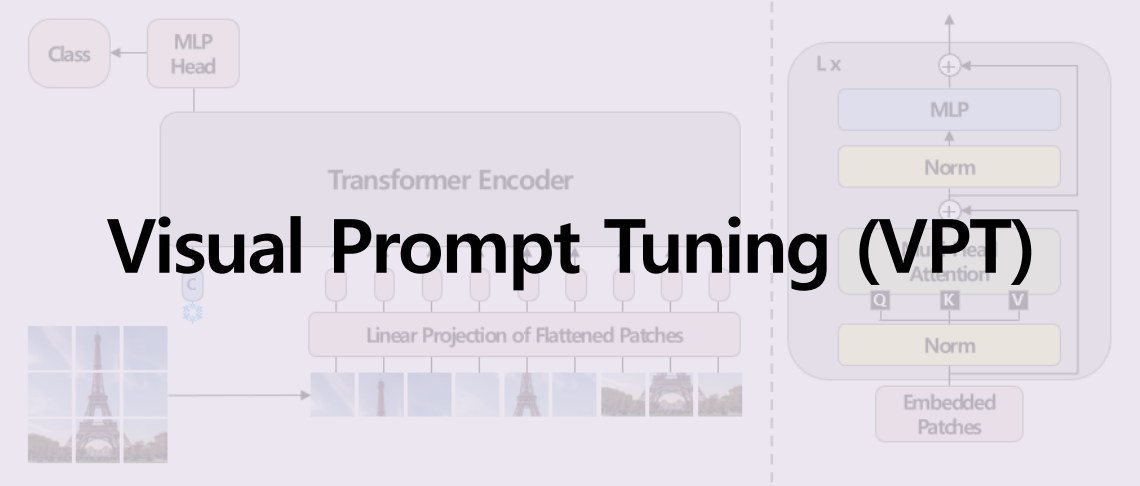
Experiments

❖ **VPT Experiment #3: Ablation on Model Design Variants ③ Prompt Depth**

- Prompt를 앞쪽 layer부터 삽입하는 Bottom→Top 방식이 전반적으로 더 높은 성능을 보여, **초기 layer에서부터 prompt가 개입하는 것**이 더 효과적임을 확인

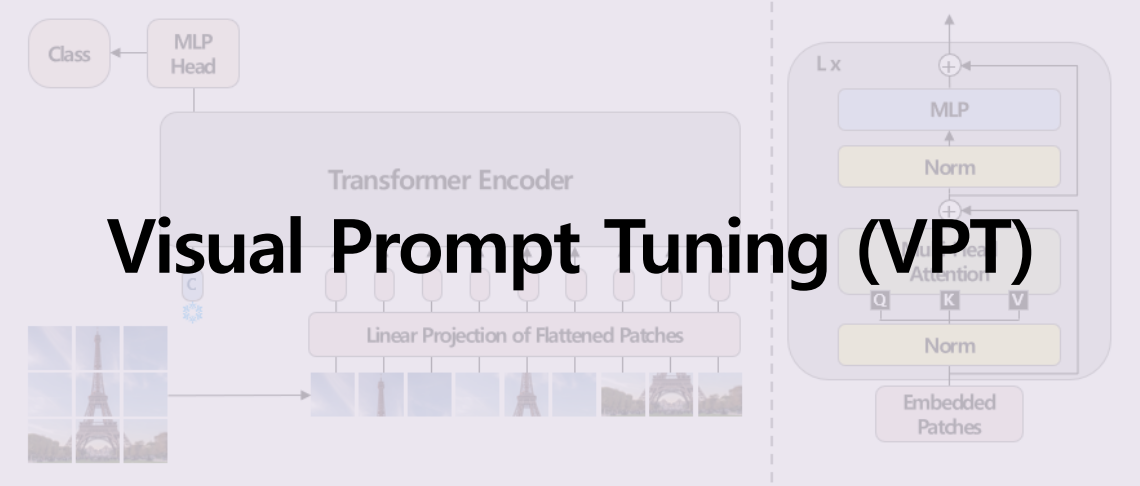


Visual Prompt Tuning (VPT)



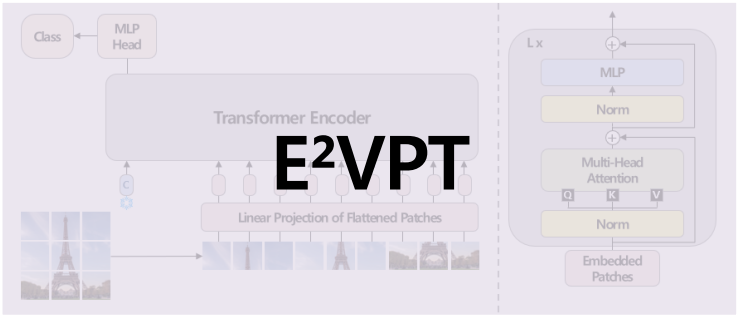
VPT의 핵심 설계 과제:
Prompt를 어디에, 얼마나 개입시킬 것인가?

Visual Prompt Tuning (VPT)

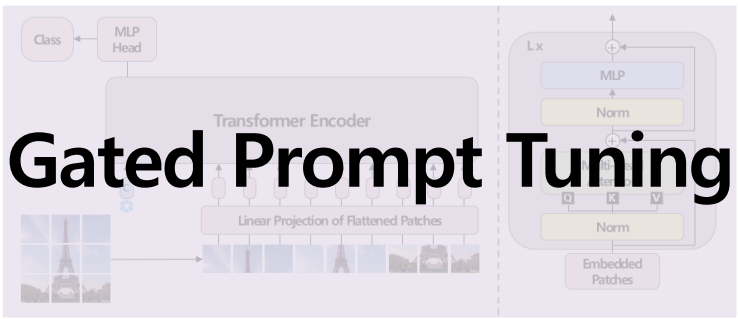


VPT의 핵심 설계 과제:
Prompt를 어디에, 얼마나 개입시킬 것인가?

self-attention 구조 내부로도 확장:



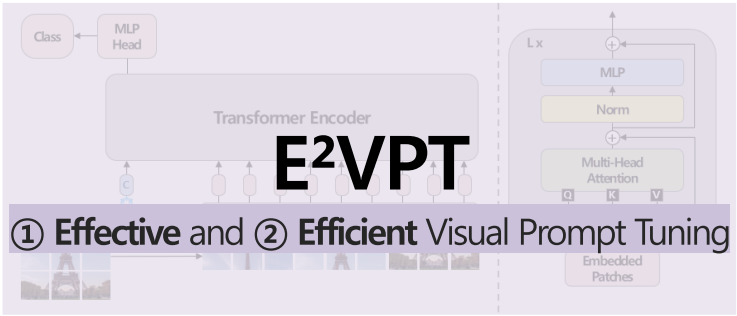
prompt의 layer별 개입 정도를 학습:



E²VPT

VPT의 핵심 설계 과제:
Prompt를 어디에, 얼마나 개입시킬 것인가?

self-attention 구조 내부로도 확장:

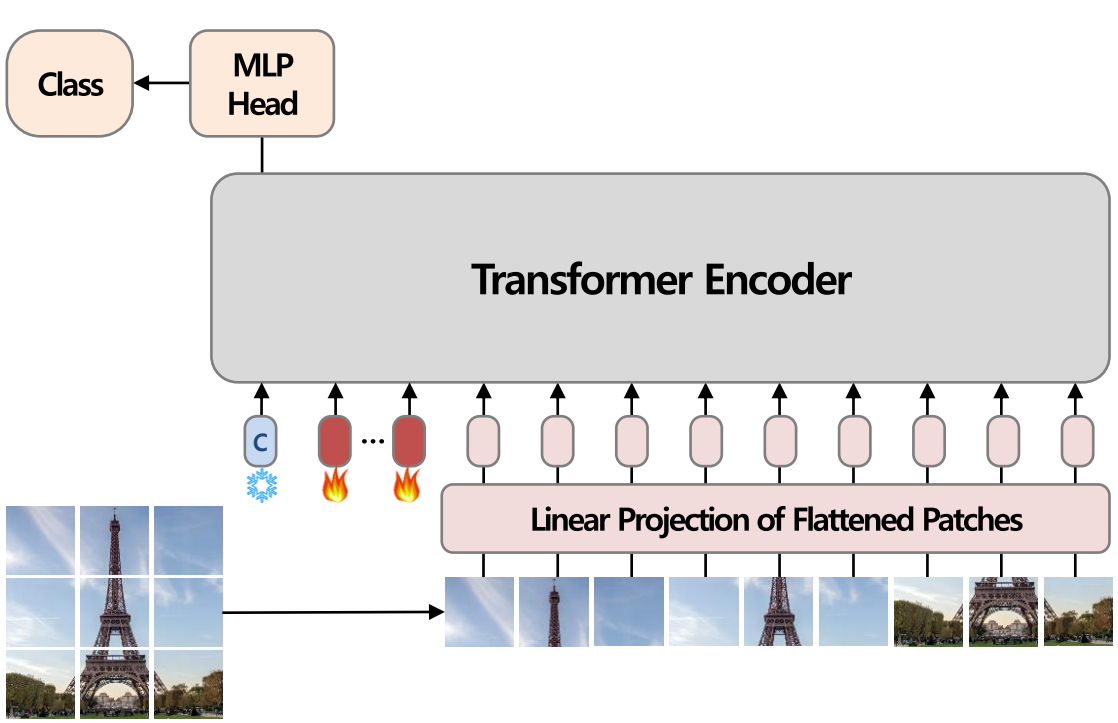


E²VPT

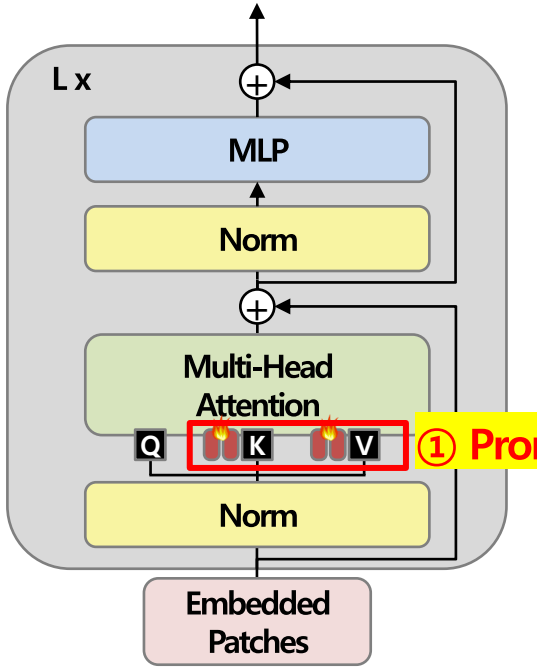
How E²VPT Works

❖ ① Effective Prompting: Key-Value Prompts

- 기존 Input Visual Prompt에 더해, **self-attention의 Key·Value에 task-specific prompt를 직접 삽입**



Vision Transformer (ViT)



① Prompt를 self-attention 내부 구조로도 확장

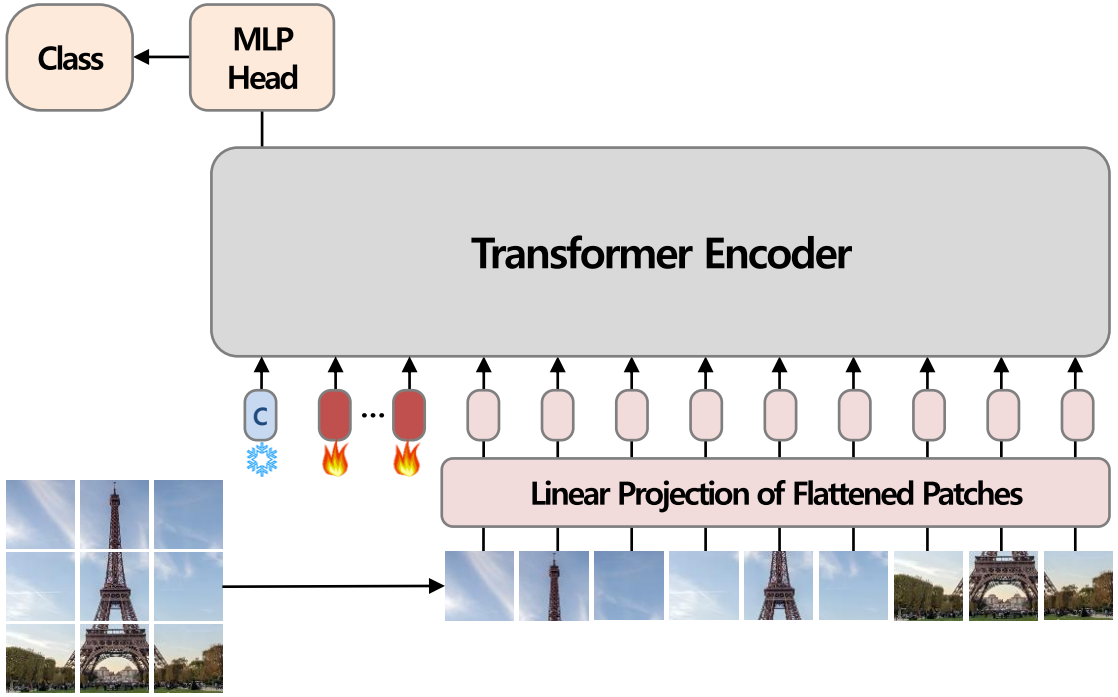
Transformer Encoder

E²VPT

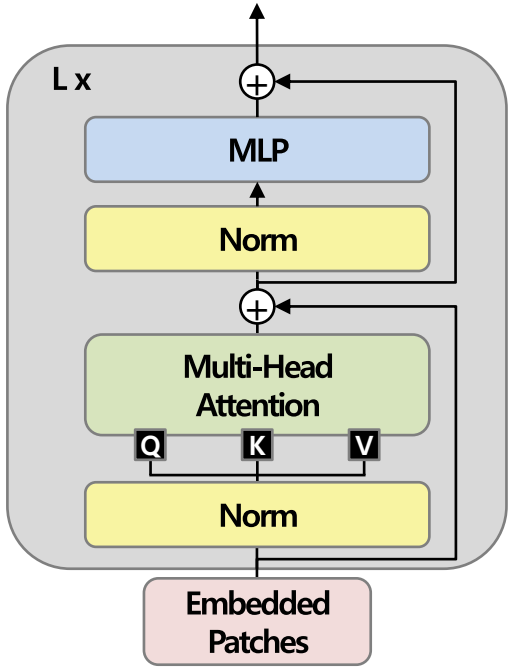
How E²VPT Works

❖ ① Effective Prompting: Key-Value Prompts

- 기존 Input Visual Prompt에 더해, **self-attention의 Key·Value에 task-specific prompt를 직접 삽입**



Vision Transformer (ViT)



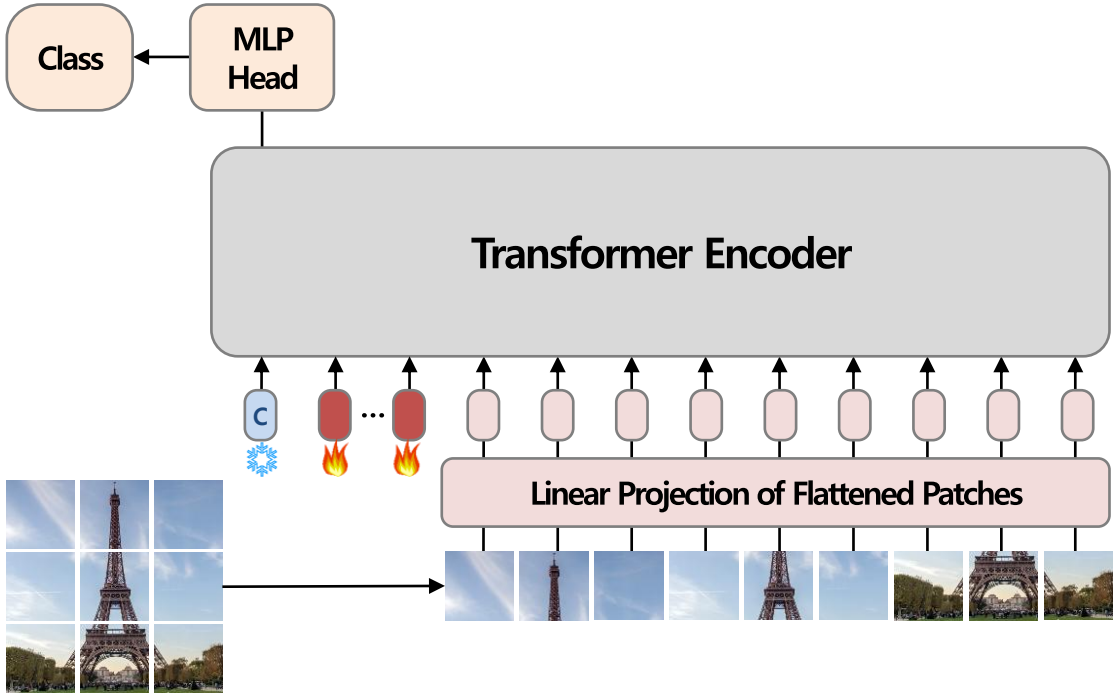
Transformer Encoder

E²VPT

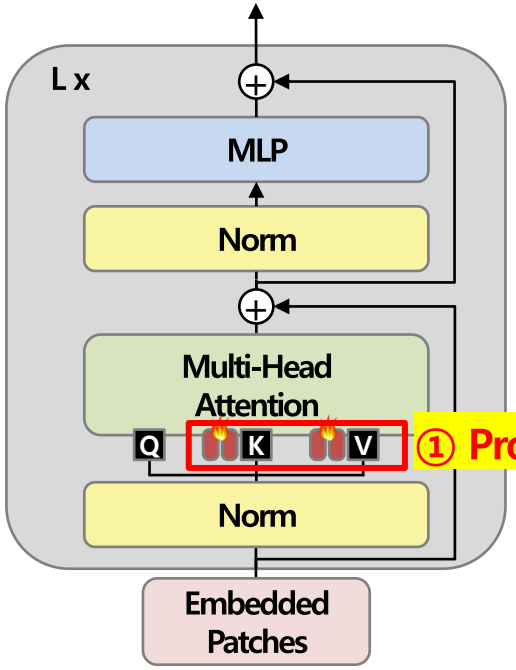
How E²VPT Works

❖ ① Effective Prompting: Key-Value Prompts

- 기존 Input Visual Prompt에 더해, **self-attention의 Key·Value에 task-specific prompt를 직접 삽입**



Vision Transformer (ViT)



① Prompt를 self-attention 내부 구조로도 확장

$$K' = concat(K, P_K), V' = concat(V, P_V)$$

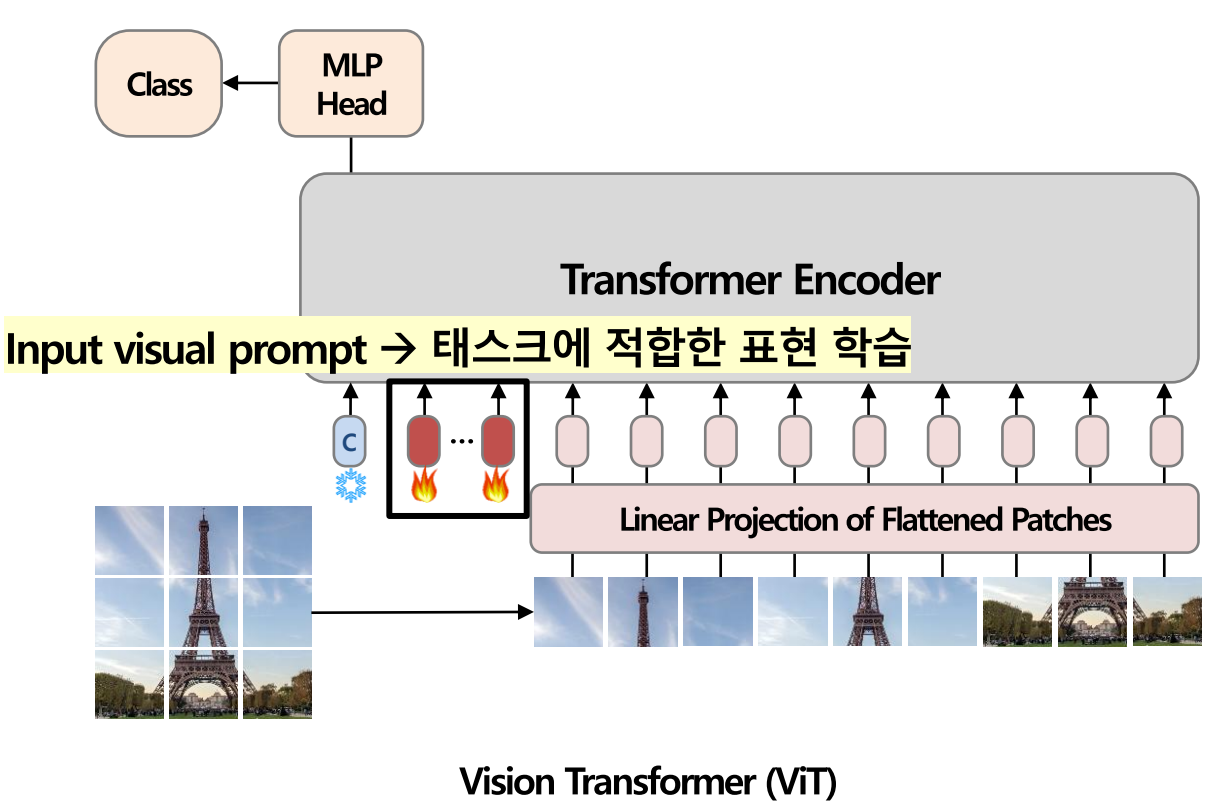
Transformer Encoder

E²VPT

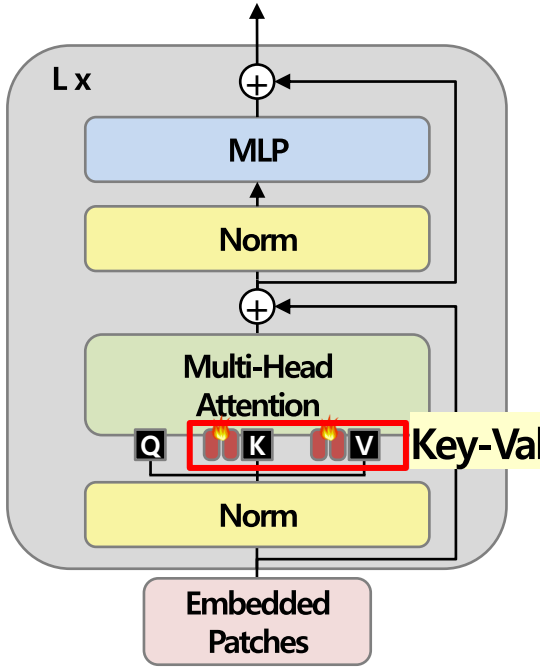
How E²VPT Works

❖ ① Effective Prompting: Key-Value Prompts

- 기존 Input Visual Prompt에 더해, **self-attention의 Key·Value에 task-specific prompt를 직접 삽입**



Vision Transformer (ViT)



Key-Value prompt → 어떤 정보에 주목할지 결정

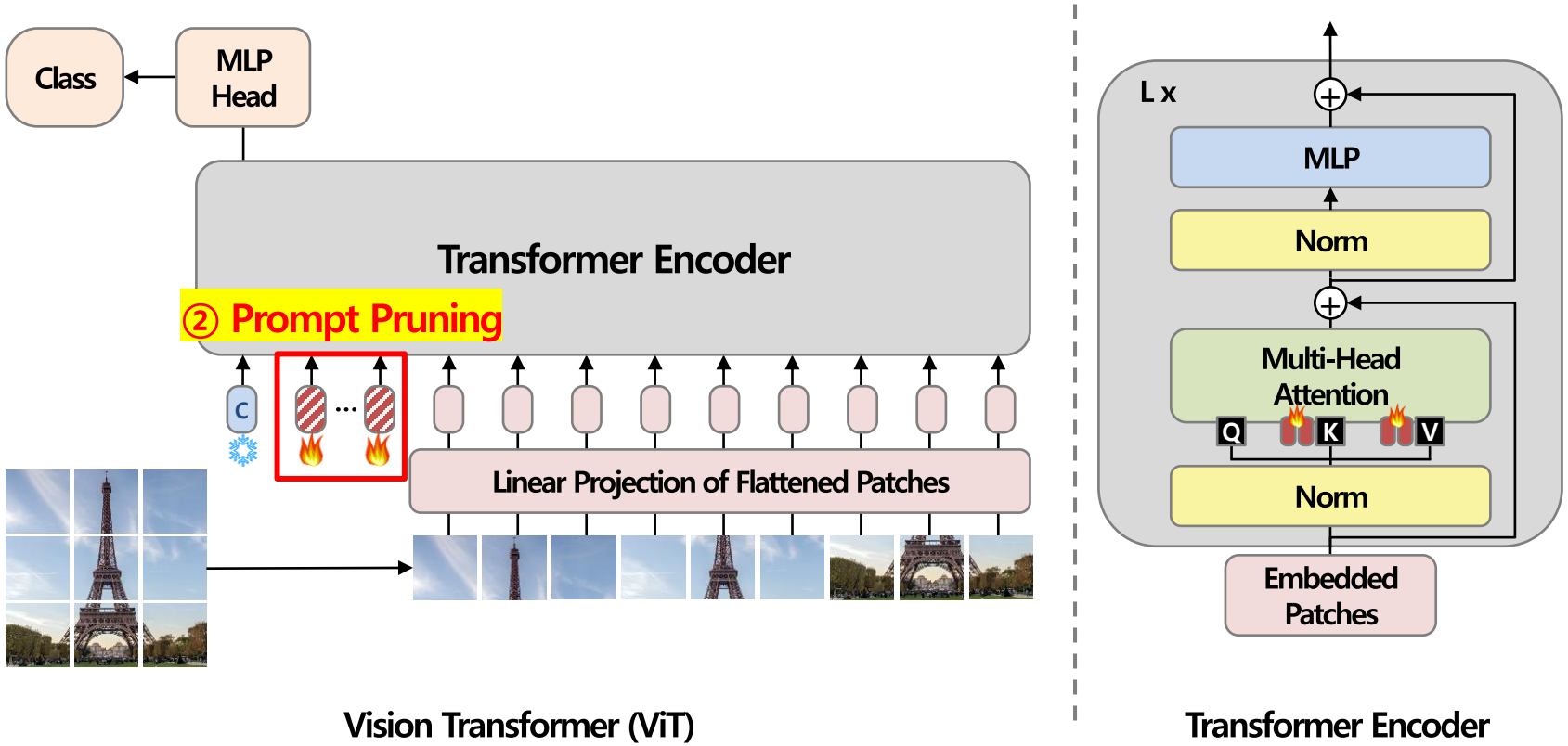
Transformer Encoder

E²VPT

How E²VPT Works

❖ ② **Efficient Prompting: Prompt Pruning**

- 중요도가 낮은 visual prompt의 token 또는 segment를 제거하여 성능 저하 없이 파라미터 효율성 향상



E²VPT

How E²VPT Works

❖ ② **Efficient Prompting: Prompt Pruning**

- **Lottery Ticket Hypothesis (ICLR 2019):** 큰 신경망 안에 전체 네트워크와 **같은 성능을 낼 수 있는 작은 부분 네트워크가 존재한다**는 가설

Published as a conference paper at ICLR 2019

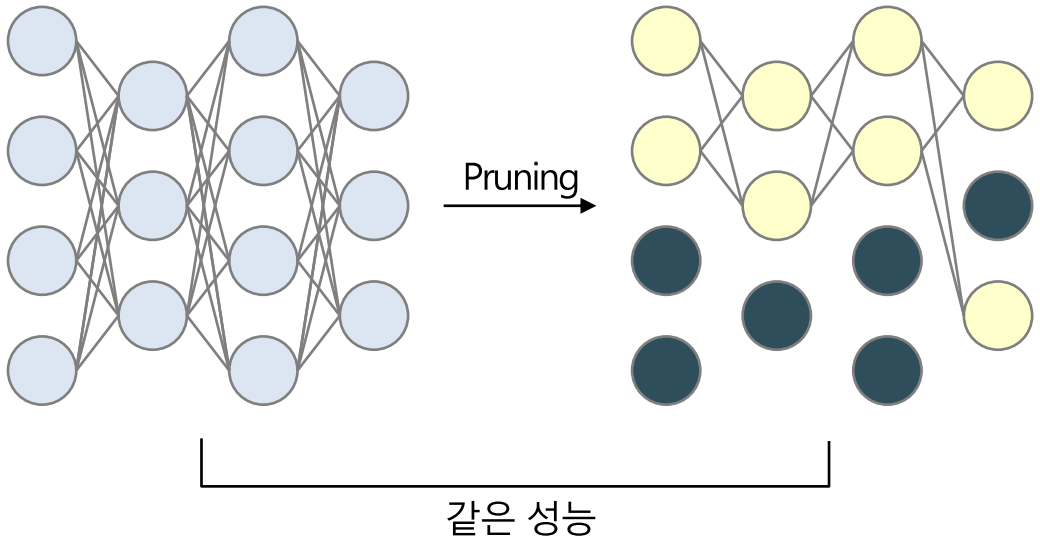
THE LOTTERY TICKET HYPOTHESIS:
FINDING SPARSE, TRAINABLE NEURAL NETWORKS

Jonathan Frankle
MIT CSAIL
jfrankle@csail.mit.edu

Michael Carbin
MIT CSAIL
mcarbin@csail.mit.edu

ABSTRACT

Neural network pruning techniques can reduce the parameter counts of trained networks by over 90%, decreasing storage requirements and improving computational performance of inference without compromising accuracy. However, contemporary experience is that the sparse architectures produced by pruning are difficult to train from the start, which would similarly improve training performance.

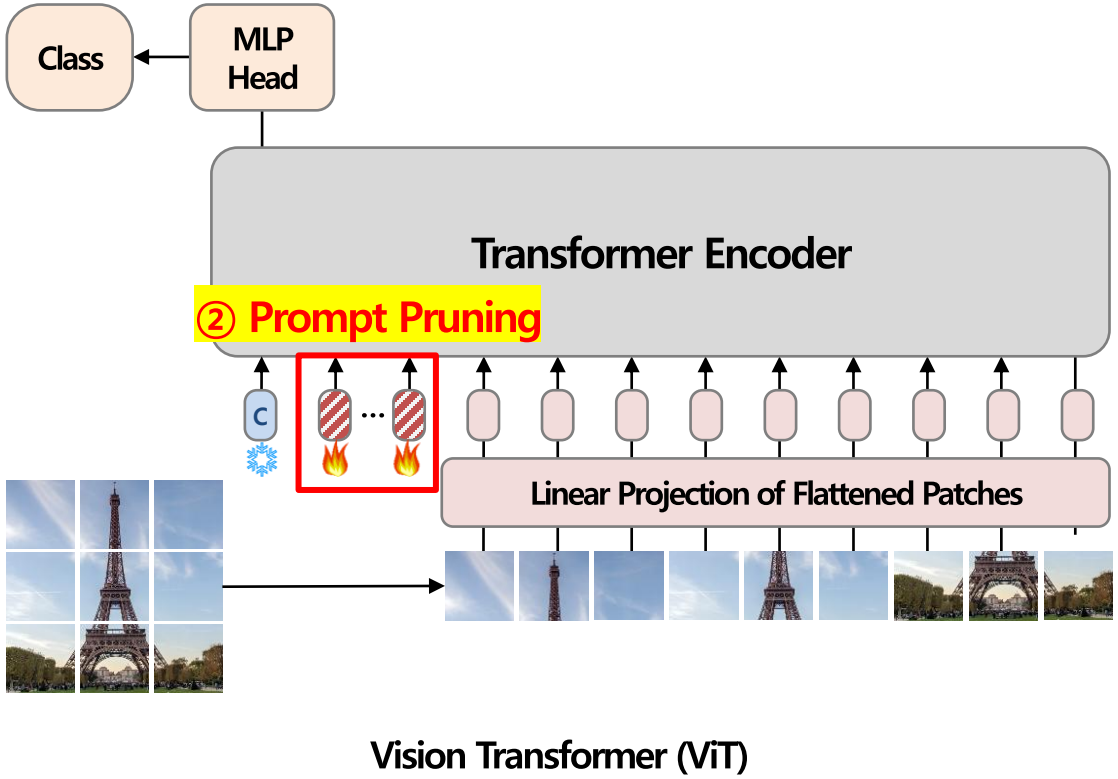


E²VPT

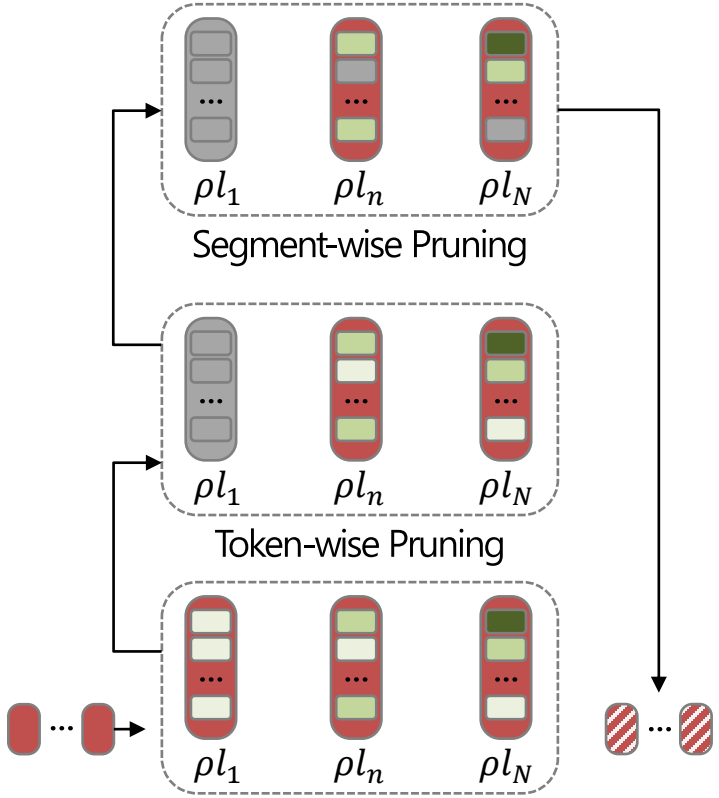
How E²VPT Works

❖ ② **Efficient Prompting: Prompt Pruning**

- 중요도가 낮은 visual prompt의 token 또는 segment를 제거하여 성능 저하 없이 파라미터 효율성 향상



Vision Transformer (ViT)

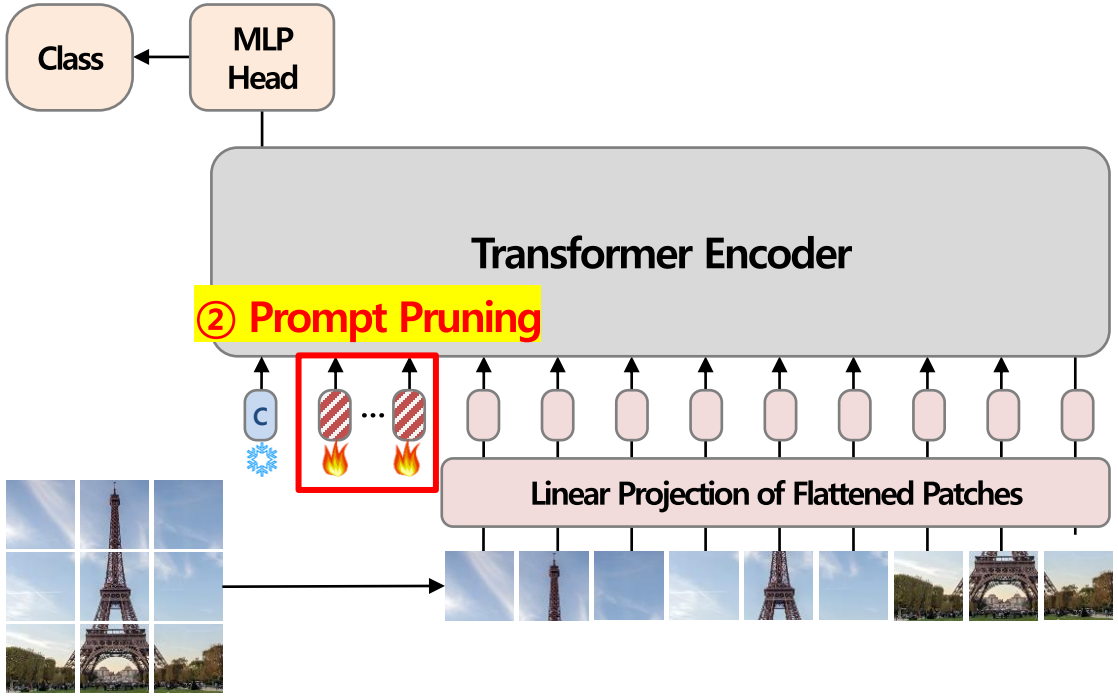


E²VPT

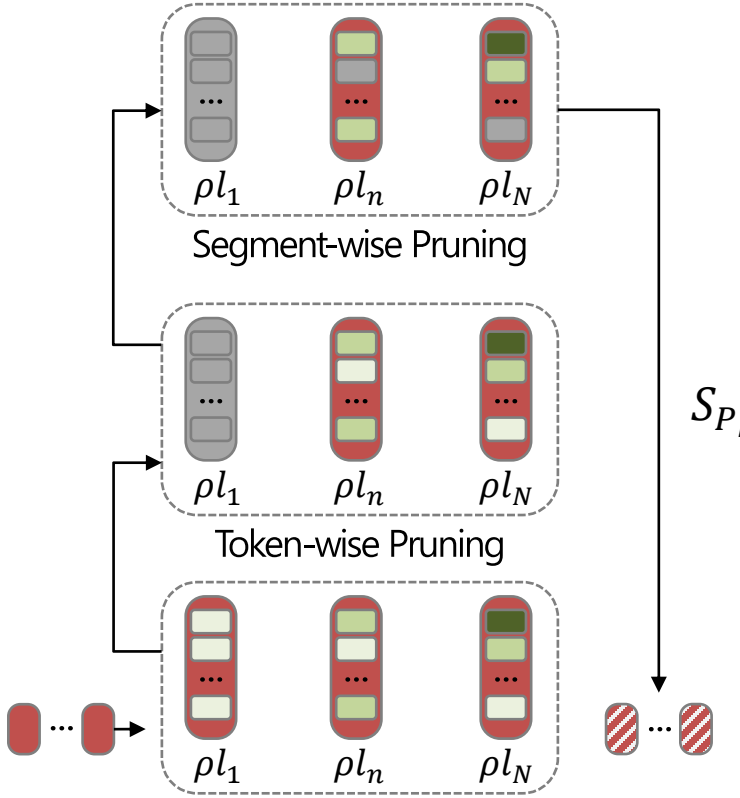
How E²VPT Works

❖ ② Efficient Prompting: Prompt Pruning

- 중요도가 낮은 visual prompt의 token 또는 segment를 제거하여 성능 저하 없이 파라미터 효율성 향상



Vision Transformer (ViT)



$$S_{P_k} = E_{x \sim D_x} \left| \frac{\partial L(x)}{\partial \rho_k} \right|$$

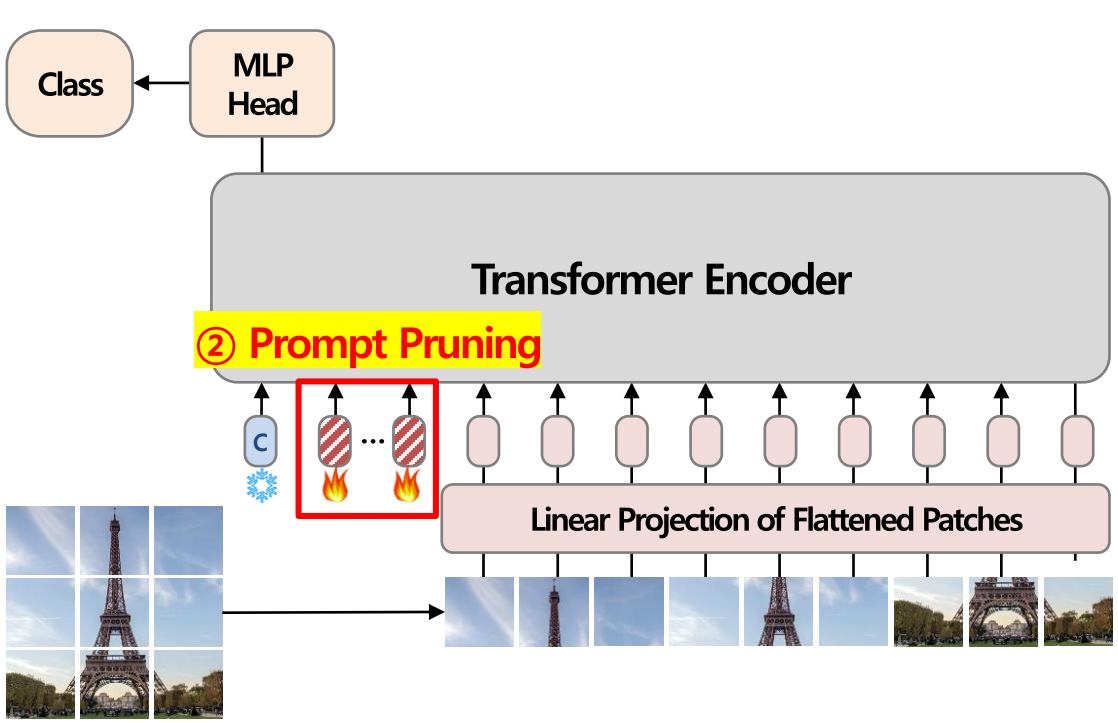
Prompt Pruning

E²VPT

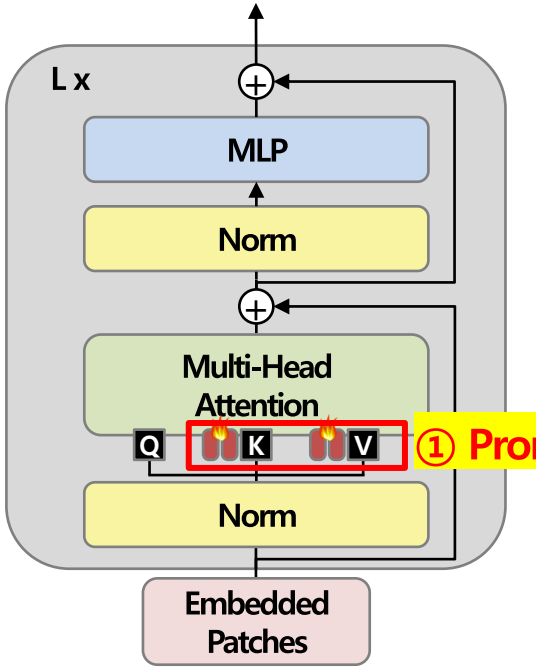
How E²VPT Works

❖ ② **Efficient Prompting: Prompt Pruning**

- 중요도가 낮은 visual prompt의 token 또는 segment를 제거하여 성능 저하 없이 파라미터 효율성 향상



Vision Transformer (ViT)



① Prompt를 self-attention 내부 구조로도 확장

E²VPT

Experiments

❖ E²VPT Experiment #1: Parameter Efficiency & Performance

- **비교 대상:** Full fine-tuning / Partial tuning / Extra module
- **학습 파라미터:** VPT **0.73%** → E²VPT **1.18×**
- **성능:** E²VPT는 24개 task 중 **VPT 대비 21개**, **Full Fine-Tuning 대비 19개** task에서 높은 성능 → 더 적은 학습 파라미터로 더 높은 성능 달성

ViT-Base/16 [12] (85.8M)	Tuned/ Total	Scope Input	Scope Backbone	Extra params	FGVC [34] [5]	VTAB-1k [96] [19] *Natural* [7]	VTAB-1k [96] [19] *Specialized* [4]	VTAB-1k [96] [19] *Structured* [8]
Full [CVPR22][32]	100.00%		✓		88.54%	75.88%	83.36%	47.64%
Linear [CVPR22][32]	0.08%				79.32% [0]	68.93% [1]	77.16% [1]	26.84% [0]
Partial-1 [NeurIPS14][91]	8.34%				82.63% [0]	69.44% [2]	78.53% [0]	34.17% [0]
MLP-3 [CVPR20][9]	1.44%			✓	79.80% [0]	67.80% [2]	72.83% [0]	30.62% [0]
Sidetune [ECCV20][98]	10.08%		✓	✓	78.35% [0]	58.21% [0]	68.12% [0]	23.41% [0]
Bias [NeurIPS17][70]	0.80%		✓		88.41% [3]	73.30% [3]	78.25% [0]	44.09% [2]
Adapter [NeurIPS20][6]	1.02%		✓	✓	85.66% [2]	70.39% [4]	77.11% [0]	33.43% [0]
VPT [ECCV22][34]	0.73%	✓		✓	89.11% [4]	78.48% [6]	82.43% [2]	54.98% [8]
Ours	0.39%	✓	✓	✓	**89.22%** [4] {4}	**80.01%** [6] {5}	**84.43%** [3] {4}	**57.39%** [8] {7}

E²VPT

Experiments

❖ E²VPT Experiment #2: Ablation on Key Design Components

- **실험 목적:** E²VPT의 세 구성 요소별 기여도 분석
- **구성 요소:** Visual Prompt / Key-Value Prompt / Pruning & Rewinding
- **평가 태스크:** VTAB-1k Natural **SVHN**, FGVC **NABirds**

Fine-tuning Techniques			VTAB-1k *Natural* SVHN [62]			FGVC NABirds [77]		
Visual Prompts	Key-Value Prompts	Pruning & Rewinding	Pruning	Tuned / Total	Accuracy	Pruning	Tuned / Total	Accuracy
✓			0.0%	0.54%	78.1%	0.0%	1.02%	84.2%
✓	✓		0.0%	0.55%	83.8%	0.0%	1.05%	84.5%
✓		✓	56.3%	0.42%	79.0%	34.4%	0.63%	84.2%
✓	✓	✓	62.5%	0.43%	**85.3%**	40.0%	0.65%	**84.6%**

E²VPT

Experiments

❖ E²VPT Experiment #2: Ablation on Key Design Components

- **실험 목적:** E²VPT의 세 구성 요소별 기여도 분석
- **구성 요소:** Visual Prompt / Key-Value Prompt / Pruning & Rewinding
- **평가 태스크:** VTAB-1k Natural **SVHN**, FGVC **NABirds**

Fine-tuning Techniques			VTAB-1k *Natural* SVHN [62]			FGVC NABirds [77]		
Visual Prompts	Key-Value Prompts	Pruning & Rewinding	Pruning	Tuned / Total	Accuracy	Pruning	Tuned / Total	Accuracy
✓			0.0%	0.54%	78.1%	0.0%	1.02%	84.2%
✓	✓		0.0%	0.55%	83.8%	0.0%	1.05%	84.5%
✓		✓	56.3%	0.42%	79.0%	34.4%	0.63%	84.2%
✓	✓	✓	62.5%	0.43%	**85.3%**	40.0%	0.65%	**84.6%**

Key-Value Prompt 추가 시, 파라미터 증가는 거의 없지만 성능 향상

E²VPT

Experiments

❖ E²VPT Experiment #2: Ablation on Key Design Components

- **실험 목적:** E²VPT의 세 구성 요소별 기여도 분석
- **구성 요소:** Visual Prompt / Key-Value Prompt / Pruning & Rewinding
- **평가 태스크:** VTAB-1k Natural **SVHN**, FGVC **NABirds**

Fine-tuning Techniques			VTAB-1k *Natural* SVHN [62]			FGVC NABirds [77]		
Visual Prompts	Key-Value Prompts	Pruning & Rewinding	Pruning	Tuned / Total	Accuracy	Pruning	Tuned / Total	Accuracy
✓			0.0%	0.54%	78.1%	0.0%	1.02%	84.2%
✓	✓		0.0%	0.55%	83.8%	0.0%	1.05%	84.5%
✓		✓	56.3%	0.42%	79.0%	34.4%	0.63%	84.2%
✓	✓	✓	62.5%	0.43%	**85.3%**	40.0%	0.65%	**84.6%**

Pruning & Rewinding 적용 시, 많은 프롬프트를 제거하고도 성능 유지

E²VPT

Experiments

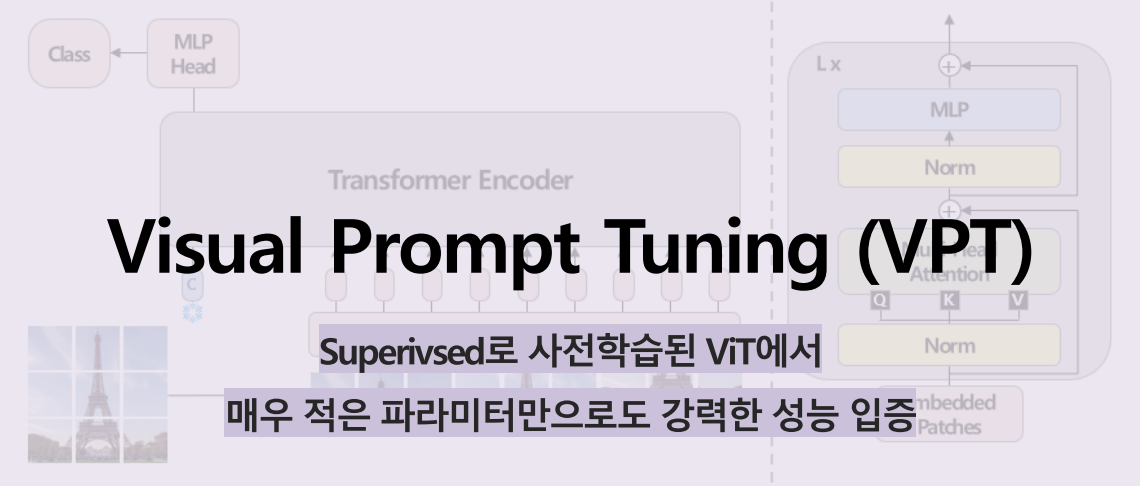
❖ E²VPT Experiment #2: Ablation on Key Design Components

- **실험 목적:** E²VPT의 세 구성 요소별 기여도 분석
- **구성 요소:** Visual Prompt / Key-Value Prompt / Pruning & Rewinding
- **평가 태스크:** VTAB-1k Natural **SVHN**, FGVC **NABirds**

Fine-tuning Techniques			VTAB-1k *Natural* SVHN [62]			FGVC NABirds [77]		
Visual Prompts	Key-Value Prompts	Pruning & Rewinding	Pruning	Tuned / Total	Accuracy	Pruning	Tuned / Total	Accuracy
✓			0.0%	0.54%	78.1%	0.0%	1.02%	84.2%
✓	✓		0.0%	0.55%	83.8%	0.0%	1.05%	84.5%
✓		✓	56.3%	0.42%	79.0%	34.4%	0.63%	84.2%
✓	✓	✓	62.5%	0.43%	**85.3%**	40.0%	0.65%	**84.6%**

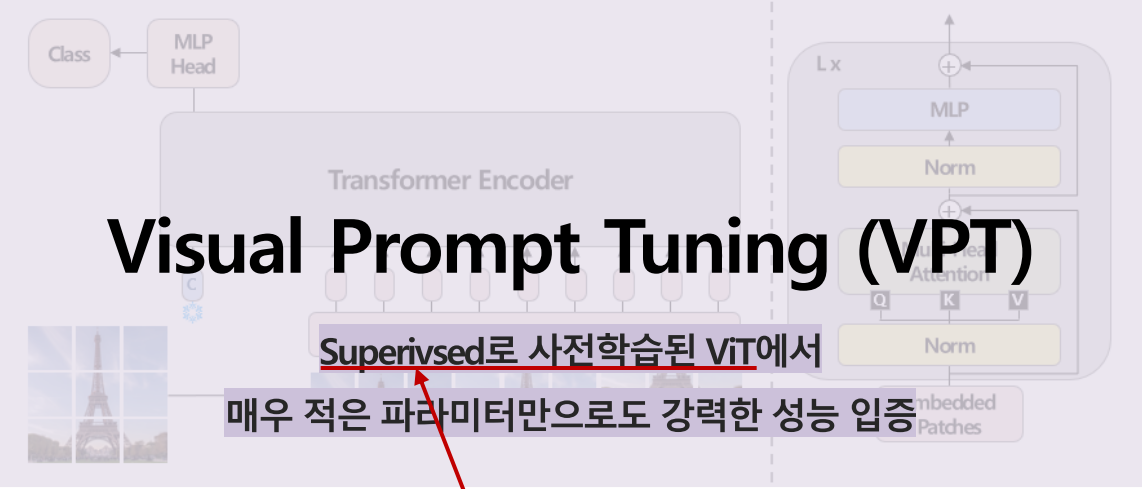
세 요소를 모두 적용 시, 최고 성능 달성

Gated Prompt Tuning



VPT의 핵심 설계 과제:
Prompt를 어디에, 얼마나 개입시킬 것인가?

Gated Prompt Tuning



→ 여전히 self-supervised로 사전학습된 ViT에서는
VPT가 잘 작동하지 않는다

VPT의 핵심 설계 과제:
Prompt를 어디에, 얼마나 개입시킬 것인가?

Gated Prompt Tuning

Self-supervised Learning

❖ Self-supervised Learning이란?

- 정의: 정답 레이블 없이 데이터 자체에서 학습 신호를 생성하여 representation을 학습하는 방식
- 대표적인 방법
  - **MAE:** 이미지 일부 patch를 가린 뒤 가려진 부분을 복원하도록 학습 → 이미지의 **구조 · 형태와 같은 시각적 정보**를 학습
  - **MoCo v3:** 동일 이미지의 서로 다른 augmented view는 가깝게 표현되도록 학습 → augmentation에 **불변한 representation** 학습
- 즉, self-supervised ViT는 layer별 representation 특성이 다르지만, VPT-Deep는 모든 layer에 prompt를 일괄 삽입 → 성능 저하

		MAE				MoCo v3			
	ViT-B/16 (85.8M)	Total params	VTAB-1k Natural	VTAB-1k Specialized	VTAB-1k Structured	Total params	VTAB-1k Natural	VTAB-1k Specialized	VTAB-1k Structured
	Total # of tasks		7	4	8		7	4	8
(a)	FULL	19.01×	59.29	79.68	53.82	19.01×	71.95	84.72	51.98
(b)	LINEAR	1.01×	18.87 (0)	53.72 (0)	23.70 (0)	1.01×	67.46 (4)	81.08 (0)	30.33 (0)
	PARTIAL-1	2.58×	**58.44 (5)**	**78.28** (1)	47.64 (1)	2.58×	72.31 (5)	**84.58 (2)**	47.89 (1)
(c)	BIAS	1.03×	54.55 (1)	75.68 (1)	**47.70** (0)	1.03×	72.89 (3)	81.14 (0)	**53.43** (4)
	ADAPTER	1.17×	54.90 (3)	75.19 (1)	38.98 (0)	1.22×	**74.19** (4)	82.66 (1)	47.69 (2)
(ours)	VPT-SHALLOW	1.01×	39.96 (1)	69.65 (0)	27.50 (0)	1.01×	67.34 (3)	82.26 (0)	37.55 (0)
	VPT-DEEP	1.04×	36.02 (0)	60.61 (1)	26.57 (0)	1.01×	70.27 (4)	83.04 (0)	42.38 (0)

Gated Prompt Tuning

VPT의 핵심 설계 과제:
Prompt를 어디에, 얼마나 개입시킬 것인가?

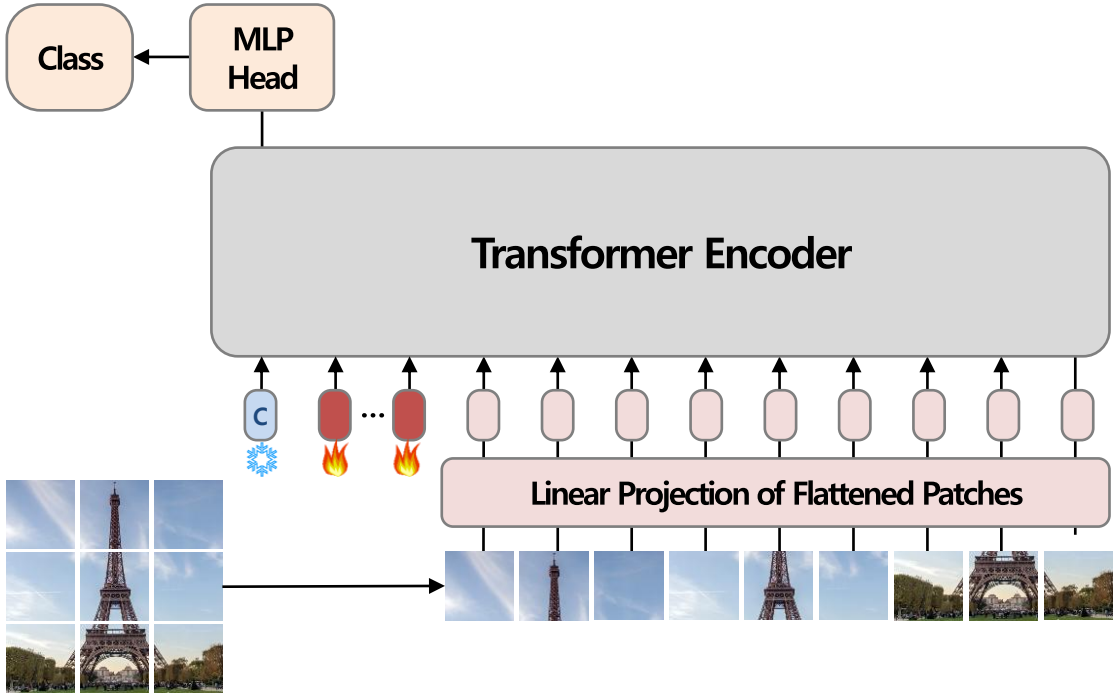
prompt의 layer별 개입 정도를 학습:

Gated Prompt Tuning

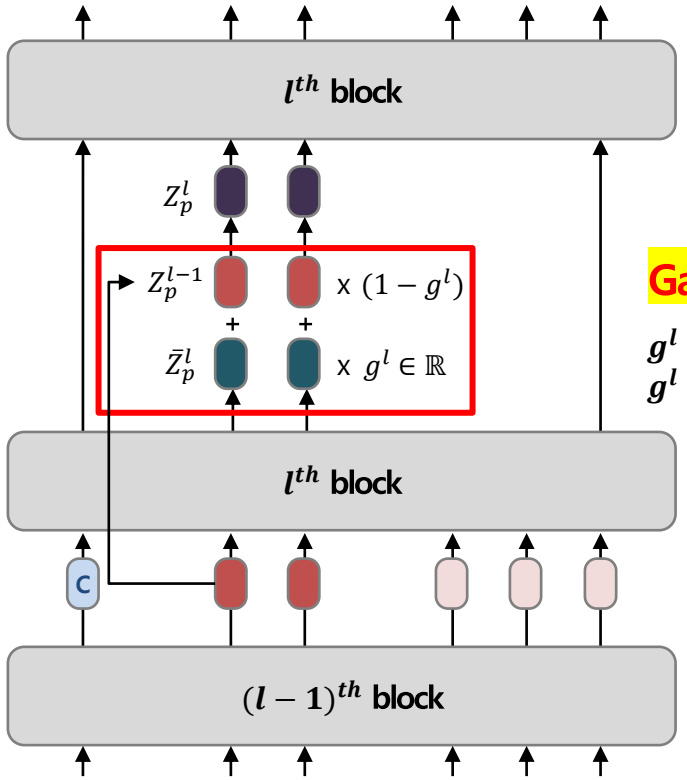
How Gated Prompt Tuning Works

❖ **Learnable Gate: Block별 Prompt 개입 정도 조절**

- 각 ViT block에 **학습 가능한 gate** g^l 를 적용하여 prompt representation의 반영 정도를 조절
- VPT-Shallow와 VPT-Deep의 장점을 모두 취함



Vision Transformer (ViT)



Gated Prompt Tuning

$g^l \uparrow$: 현재 block 표현 반영 ↑
$g^l \downarrow$: 이전 prompt 표현 유지 ↑

Learnable Gate (g^l)

Gated Prompt Tuning

Experiments

❖ **Gated Prompt Tuning Experiment #1: Classification results on FGVC**

- **비교 대상:** VPT-Shallow / VPT-Deep
- **평가 환경:** MAE / MoCo v3로 self-supervised pre-trained ViT-B/16
- **성능: MAE 73.39%, MoCo v3 83.00%** → VPT-Shallow 대비 큰 폭 향상, **VPT-Deep 대비 우수하거나 대등한 성능**

SSL	METHOD	TOTAL PARAMS	CUB	FLOWERS	CARS	DOGS	NABIRDS	AVG
MAE	VPT-SHALLOW	1.02×	42.15	69.15	43.38	77.07	57.43	57.84
	VPT-DEEP	1.02×	68.33	**80.05**	67.67	78.83	65.22	72.02
	OURS	1.02×	**70.56**	78.55	**71.70**	**78.9**	**67.26**	**73.39**
MOCO V3	VPT-SHALLOW	1.02×	79.05	90.47	71.91	81.97	72.92	79.26
	VPT-DEEP	1.02×	82.67	**94.41**	**79.18**	83.33	75.99	**83.12**
	OURS	1.02×	**82.86**	93.71	79.02	**83.37**	**76.02**	83.00

Gated Prompt Tuning

Experiments

❖ Gated Prompt Tuning Experiment #2: Classification results on VTAB-1K

- **비교 대상:** VPT-Shallow / VPT-Deep
- **평가 환경:** MAE / MoCo v3로 self-supervised pre-trained ViT-B/16
- **성능: MAE 49.22%, MoCo v3 65.80%** → 두 환경 모두 **VPT-Shallow / VPT-Deep 대비 높은 성능**

SSL	METHOD	TOTAL PARAMS	NATURAL (7)	SPECIALIZED (4)	STRUCTURED (8)	AVG
MAE	VPT-SHALLOW	1.01×	39.96†	69.65†	27.50†	40.96†
	VPT-DEEP	1.04×	36.02†	60.61†	26.57†	37.22†
	OURS	1.01×	**47.61**	**76.86**	**36.80**	**49.22**
MoCo v3	VPT-SHALLOW	1.01×	67.34†	82.26†	37.55†	57.94†
	VPT-DEEP	1.01×	70.27†	83.04†	42.38†	61.22†
	OURS	1.01×	**74.84**	**83.38**	**49.10**	**65.80**

Gated Prompt Tuning

Experiments

❖ **Gated Prompt Tuning Experiment #3: Semantic Segmentation on ADE20K**

- **평가 환경:** SETR-PUP / MAE, MoCo v3로 self-supervised pre-trained ViT-B/16
- **평가 지표:** mIOU ↑ (SS: Single-Scale / MS: Multi-Scale inference)
- **성능: MAE 38.44 / 39.81, MoCo v3 36.81 / 38.55** → VPT-Deep보다 적은 prompt로 더 높은 성능

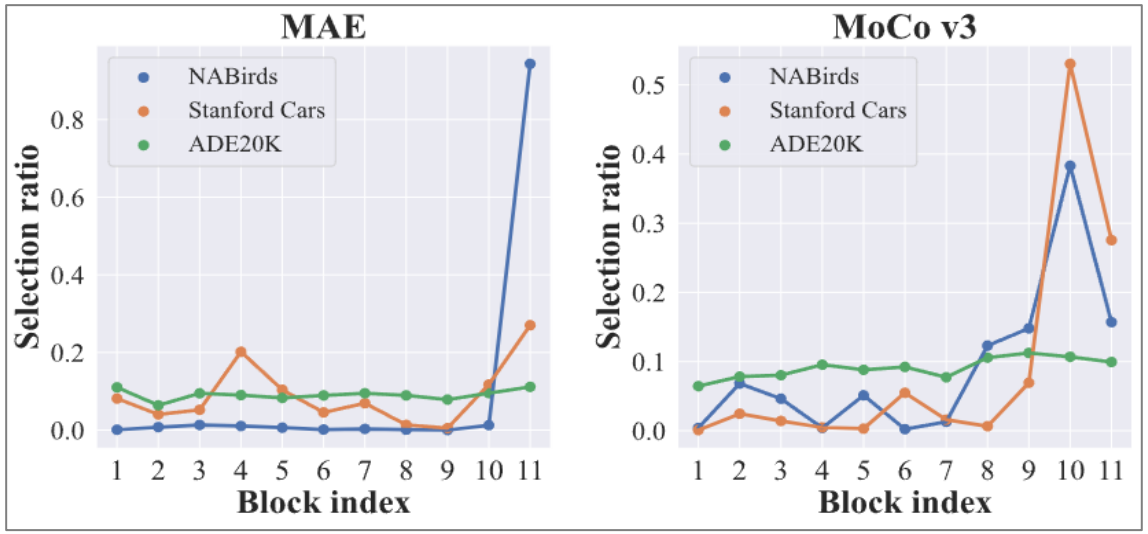
SSL	METHOD	# OF PTS	MIOU (SS)	MIOU (MS)
MAE	VPT-SHALLOW	100	34.20	35.23
	VPT-DEEP	10(×12)	37.76	38.80
	OURS	100	**38.44**	**39.81**
MOCO V3	VPT-SHALLOW	100	34.55	36.18
	VPT-DEEP	10(×12)	35.50	37.15
	OURS	100	**36.81**	**38.55**

Gated Prompt Tuning

Experiments

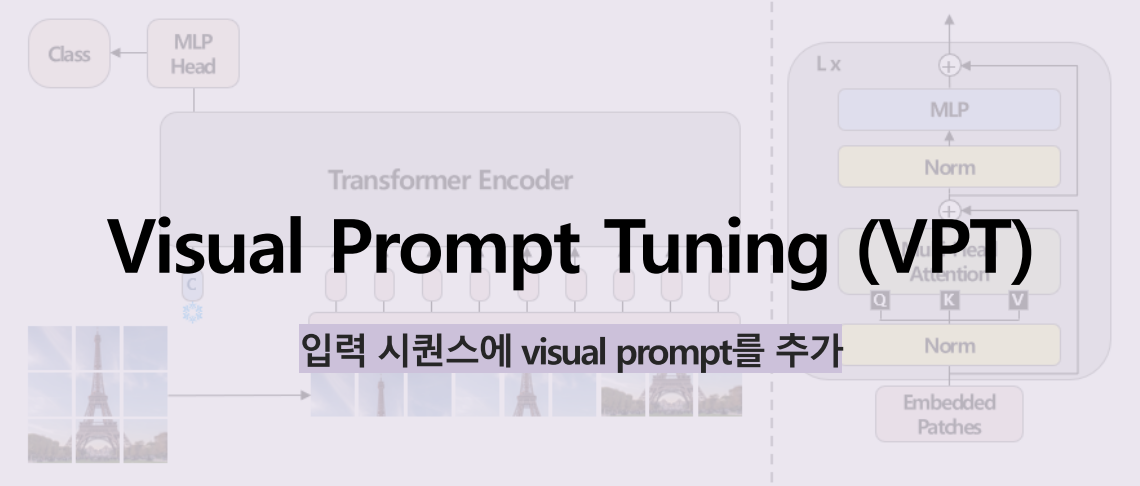
❖ **Gated Prompt Tuning Experiment #4: Analysis on Learned Gates**

- **Selection ratio:** 각 block이 최종 prompt representation에 미치는 영향
- **Task별 차이:** MAE에서 NABirds는 **11번째 block**, Stanford Cars는 **4·11번째 block**에 집중
- **Backbone별 차이:** 동일 task에서도 **MAE와 MoCo v3가 서로 다른 block을 선택**
- **ADE20K:** 여러 block을 비교적 고르게 활용 → **segmentation에는 multi-level feature가 중요**



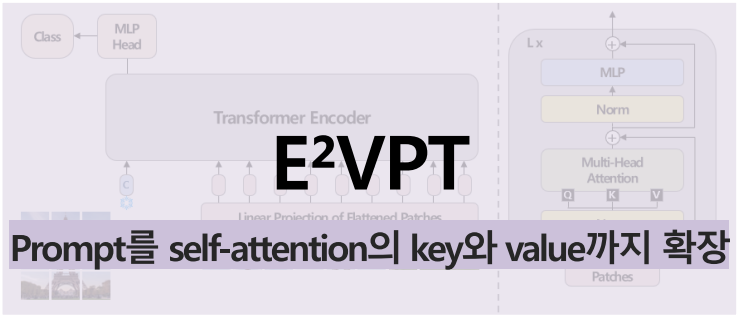
Task와 SSL backbone에 따라 필요한 block을 선택적으로 반영

Conclusion



VPT의 핵심 설계 과제:
Prompt를 어디에, 얼마나 개입시킬 것인가?

self-attention 구조 내부로도 확장:



prompt의 layer별 개입 정도를 학습:

Conclusion

VPT 계열 연구 방향

SA²VP: Spatially Aligned-and-Adapted Visual Prompt

Wenjie Pei[1*], Tongqi Xia[1*], Fanglin Chen[1], Jinsong Li[2], Jiandong Tian[3], Guangming Lu[1†]

[1]Harbin Institute of Technology, Shenzhen
[2]Shenzhen Jiang & Associates Creative Design Co., Ltd
[3]Shenyang Institute of Automation, Chinese Academy of Sciences
wenjiecoder@outlook.com, tongqixia325@gmail.com, chenfanglin@hit.edu.cn, as@jad.cn, tianjd@sia.cn, luguangm@hit.edu.cn

Abstract

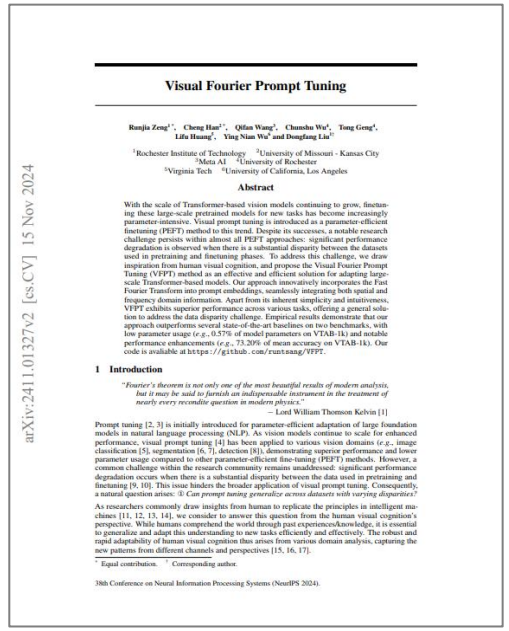
As a prominent parameter-efficient fine-tuning technique in NLP, prompt tuning is being explored its potential in computer vision. Typical methods for visual prompt tuning follow the sequential modeling paradigm stemming from NLP, which represents an input image as a flattened sequence of token embeddings and then learns a set of unordered parameterized tokens prefixed to the sequence representation as the visual prompts for task adaptation of large vision models. While such sequential modeling paradigm of visual prompt has shown great promise, there are two potential limitations. First, the learned visual prompts cannot model the underlying spatial relations in the input image, which is crucial for image encoding. Second, since all prompt tokens play the same role of prompting for all image tokens without distinction, it lacks the fine-grained prompting capability, i.e., individual prompting for different image tokens. In this work, we propose the Spatially Aligned-and-Adapted Visual Prompt model (SA²VP), which learns a two-dimensional prompt token map with equal (or scaled) size to the image token map, thereby being able to spatially align with the image map. Each prompt token is designated to prompt knowledge only for the spatially corresponding image tokens. As a result, our model can conduct individual prompting for different image tokens in a fine-grained manner. Moreover, benefiting from the capability of preserving the spatial structure by the learned prompt token map, our SA²VP is able to model the spatial relations in the input image, leading to more effective prompting. Extensive experiments on three challenging benchmarks for image classification demonstrate the superiority of our model over other state-of-the-art methods for visual prompt tuning. Code is available at https://github.com/tommy-sjq/SA2VP.

Introduction

Adapting large pre-trained vision models to a specific downstream vision task via parameter-efficient fine-tuning (PEFT) has proven to be an effective and efficient strategy, particularly in the scenarios where the training data is limited. It not only distills the prior knowledge relevant to the downstream task from the pre-trained model to achieve better performance, but also substantially improves the training efficiency by incurring only a small amount of parameter-tuning overhead. As a prominent PEFT method, prompt tuning has achieved remarkable success in NLP field and is being explored its potential in computer vision (Bahng et al. 2022; Huang et al. 2023; Jia et al. 2022; Das et al. 2023; Han et al. 2023; Bar et al. 2022).

Most of existing methods for visual prompt tuning (Jia et al. 2022; Das et al. 2023; Han et al. 2023) follow the sequential modeling paradigm, which stems from the learning way of prompt in NLP, as shown in Figure 1. A prominent example is VPT (Jia et al. 2022), which encodes an input image as a sequential representation using pre-trained vision Transformer models and then learns a set of prompt tokens as auxiliary input prefixed to the image representation. The learned vision prompts are expected to distill the relevant knowledge from the pre-trained model, thereby adapting the pre-trained model to the downstream target task. While such sequential modeling paradigm of visual prompts has shown great promise, it suffers from two potential limitations. First,

Figure 1: Typical methods for visual prompting tuning follow the sequential modeling paradigm which learns an unordered sequence of prompts prefixed to the image representation. All prompt tokens play the same role of prompting for all image tokens. In contrast, our SA²VP learns a 2D prompt token map, which aligns with the image token map spatially. As a result, our SA²VP is able to 1) conduct individual prompting for different image tokens in a fine-grained manner and 2) model the spatial relations in the input image by preserving the spatial structure in the prompt map.



Visual Fourier Prompt Tuning

Runjia Zeng[1*], Cheng Han[2*], Qifan Wang[3], Chunshu Wu[4], Tong Geng[4], Lifu Huang[5], Ying Nian Wu[6] and Dongfang Liu[1†]

[1]Rochester Institute of Technology [2]University of Missouri - Kansas City
[3]Meta AI [4]University of Rochester
[5]Virginia Tech [6]University of California, Los Angeles

Abstract

With the scale of Transformer-based vision models continuing to grow, fine-tuning these large-scale pretrained models for new tasks has become increasingly parameter-intensive. Visual prompt tuning is introduced as a parameter-efficient finetuning (PEFT) method to this trend. Despite its successes, a notable research challenge persists within almost all PEFT approaches: significant performance degradation is observed when there is a substantial disparity between the datasets used in pretraining and finetuning phases. To address this challenge, we draw inspiration from human visual cognition, and propose the Visual Fourier Prompt Tuning (VFPT) method as an effective and efficient solution for adapting large-scale Transformer-based models. Our approach innovatively incorporates the Fast Fourier Transform into prompt embeddings, seamlessly integrating both spatial and frequency domain information. Apart from its inherent simplicity and intuitiveness, VFPT exhibits superior performance across various tasks, offering a general solution to address the data disparity challenge. Empirical results demonstrate that our approach outperforms several state-of-the-art baselines on two benchmarks, with low parameter usage (e.g., 0.57% of model parameters on VTAB-1k) and notable performance enhancements (e.g., 73.20% of mean accuracy on VTAB-1k). Our code is available at https://github.com/runtsang/VFPT.

1 Introduction

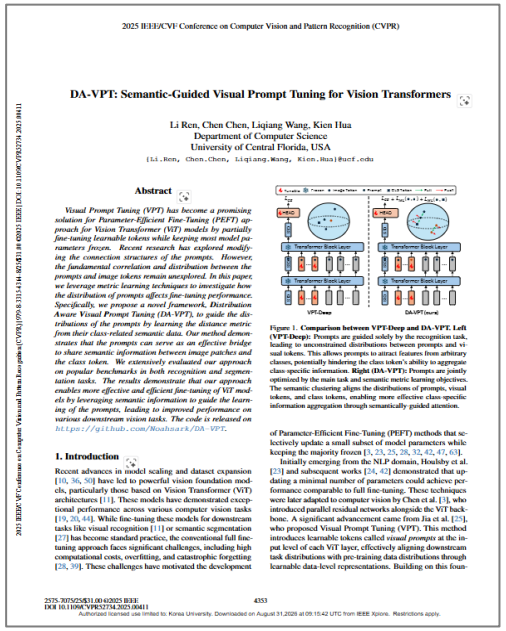
"Fourier's theorem is not only one of the most beautiful results of modern analysis, but it may be said to furnish an indispensable instrument in the treatment of nearly every recondite question in modern physics."
— Lord William Thomson Kelvin [1]

Prompt tuning [2, 3] is initially introduced for parameter-efficient adaptation of large foundation models in natural language processing (NLP). As vision models continue to scale for enhanced performance, visual prompt tuning [4] has been applied to various vision domains (e.g., image classification [5], segmentation [6, 7], detection [8]), demonstrating superior performance and lower parameter usage compared to other parameter-efficient fine-tuning (PEFT) methods. However, a common challenge within the research community remains unaddressed: significant performance degradation occurs when there is a substantial disparity between the data used in pretraining and finetuning [9, 10]. This issue hinders the broader application of visual prompt tuning. Consequently, a natural question arises: *① Can prompt tuning generalize across datasets with varying disparities?*

As researchers commonly draw insights from human to replicate the principles in intelligent machines [11, 12, 13, 14], we consider to answer this question from the human visual cognition's perspective. While humans comprehend the world through past experiences/knowledge, it is essential to generalize and adapt this understanding to new tasks efficiently and effectively. The robust and rapid adaptability of human visual cognition thus arises from various domain analysis, capturing the new patterns from different channels and perspectives [15, 16, 17].

* Equal contribution. † Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).



2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

DA-VPT: Semantic-Guided Visual Prompt Tuning for Vision Transformers

Li Ren, Chen Chen, Liqiang Wang, Kien Hua
Department of Computer Science
University of Central Florida, USA
{Li.Ren, Chen.Chen, Liqiang.Wang, Kien.Hua}@ucf.edu

Abstract

Visual Prompt Tuning (VPT) has become a promising solution for Parameter-Efficient Fine-Tuning (PEFT) approach for Vision Transformer (ViT) models by partially fine-tuning learnable tokens while keeping most model parameters frozen. Recent research has explored modifying the connection structures of the prompts. However, the fundamental correlation and distribution between the prompts and image tokens remain unexplored. In this paper, we leverage metric learning techniques to investigate how the distribution of prompts affects fine-tuning performance. Specifically, we propose a novel framework, Distribution Aware Visual Prompt Tuning (DA-VPT), to guide the distributions of the prompts by learning the distance metric from their class-related semantic data. Our method demonstrates that the prompts can serve as an effective bridge to share semantic information between image patches and the class token. We extensively evaluated our approach on popular benchmarks in both recognition and segmentation tasks. The results demonstrate that our approach enables more effective and efficient fine-tuning of ViT models by leveraging semantic information to guide the learning of the prompts, leading to improved performance on various downstream vision tasks. The code is released on https://github.com/Wuhuark/DA-VPT.

1. Introduction

Recent advances in model scaling and dataset expansion [10, 36, 50] have led to powerful vision foundation models, particularly those based on Vision Transformer (ViT) architectures [11]. These models have demonstrated exceptional performance across various computer vision tasks [19, 20, 44]. While fine-tuning these models for downstream tasks like visual recognition [11] or semantic segmentation [27] has become standard practice, the conventional full fine-tuning approach faces significant challenges, including high computational costs, overfitting, and catastrophic forgetting [28, 39]. These challenges have motivated the development of Parameter-Efficient Fine-Tuning (PEFT) methods that selectively update a small subset of model parameters while keeping the majority frozen [3, 23, 25, 28, 32, 42, 47, 63].

Initially emerging from the NLP domain, Houlsby et al. [23] and subsequent works [24, 42] demonstrated that updating a minimal number of parameters could achieve performance comparable to full fine-tuning. These techniques were later adapted to computer vision by Chen et al. [3], who introduced parallel residual networks alongside the ViT backbone. A significant advancement came from Jia et al. [25], who proposed Visual Prompt Tuning (VPT). This method introduces learnable tokens called *visual prompts* at the input level of each ViT layer, effectively aligning downstream task distributions with pre-training data distributions through learnable data-level representations. Building on this foun-

Figure 1. **Comparison between VPT-Deep and DA-VPT. Left (VPT-Deep):** Prompts are guided solely by the recognition task, leading to unconstrained distributions between prompts and visual tokens. This allows prompts to attract features from arbitrary classes, potentially hindering the class token's ability to aggregate class specific information. **Right (DA-VPT):** Prompts are jointly optimized by the main task and semantic metric learning objectives. The semantic clustering aligns the distributions of prompts, visual tokens, and class tokens, enabling more effective class specific information aggregation through semantically guided attention.

2575-2375/25$31.00©2025 IEEE
DOI 10.1109/CVPR52734.2025.00411

SA²VP: Spatially Aligned-and-Adapted Visual Prompt
(Pei et al., AAAI 2024)

→ prompt의 공간 구조를 2차원으로 정교화

Visual Fourier Prompt Tuning
(Zeng et al., NeurIPS 2024)

→ prompt의 표현 영역을 주파수 도메인으로 확장

DA-VPT: Semantic-Guided Visual Prompt Tuning for Vision Transformers
(Ren et al., CVPR 2025)

→ prompt 학습에 의미적 제약을 부여

Thank You

References

[1] Jia, M., Tang, L., Chen, B. C., Cardie, C., Belongie, S., Hariharan, B., & Lim, S. N. (2022). Visual prompt tuning. In European Conference on Computer Vision (pp. 709-727).

[2] Han, C., Wang, Q., Cui, Y., Cao, Z., Wang, W., Qi, S., & Liu, D. (2023). E²VPT: An effective and efficient approach for visual prompt tuning. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 17491-17502).

[3] Yoo, S., Kim, E., Jung, D., Lee, J., & Yoon, S. (2023). Improving visual prompt tuning for self-supervised vision transformers. In International Conference on Machine Learning (pp. 39848-39861).