

Medical Imaging & Intelligent Reality  
Lab.  
Convergence Medicine/Radiology

# Visual Prompt Tuning



Git: <https://github.com/KMnP/vpt>

Presenter: Sunggu Kyung

Email: [babbu3682@gmail.com](mailto:babbu3682@gmail.com)

# Presentation Contents

1

Abstract

2

Introduction

3

Methods

4

Experiments

5

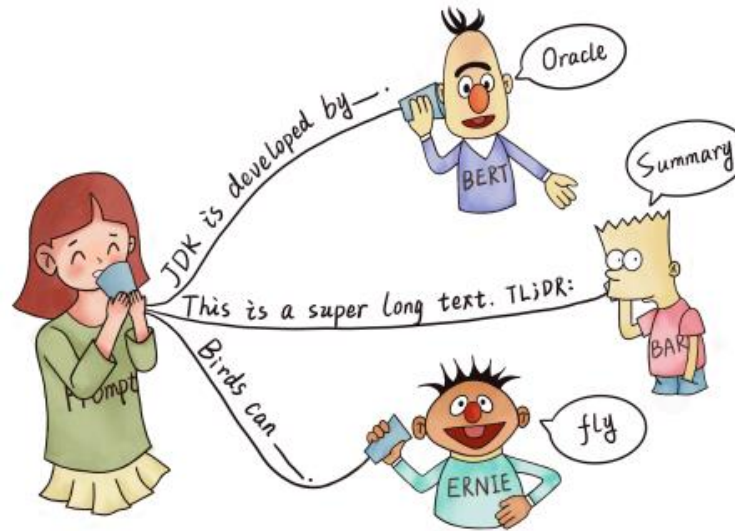
Conclusion

# Background prompt

# prompt

## Definition:

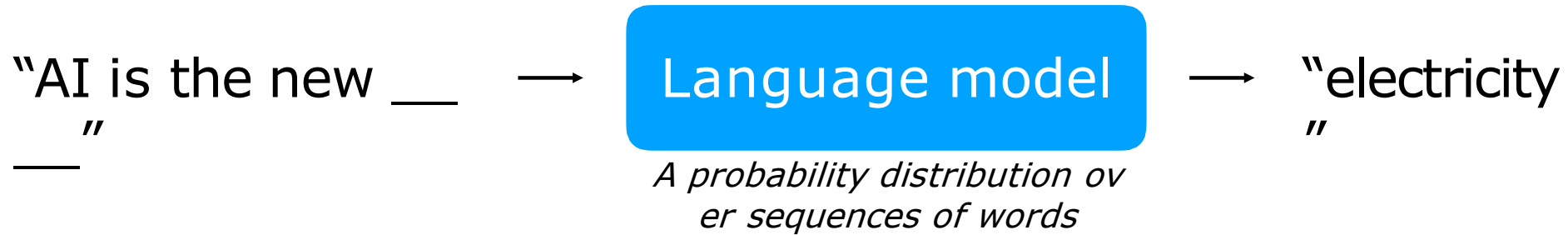
A prompt is a question, statement, or request that is input into an AI system to elicit a specific response or output.



In the context of NLP, Prompt Engineering involves crafting prompts that effectively convey the desired task or question and encourage coherent, accurate, and relevant responses from the AI model.

# Language model

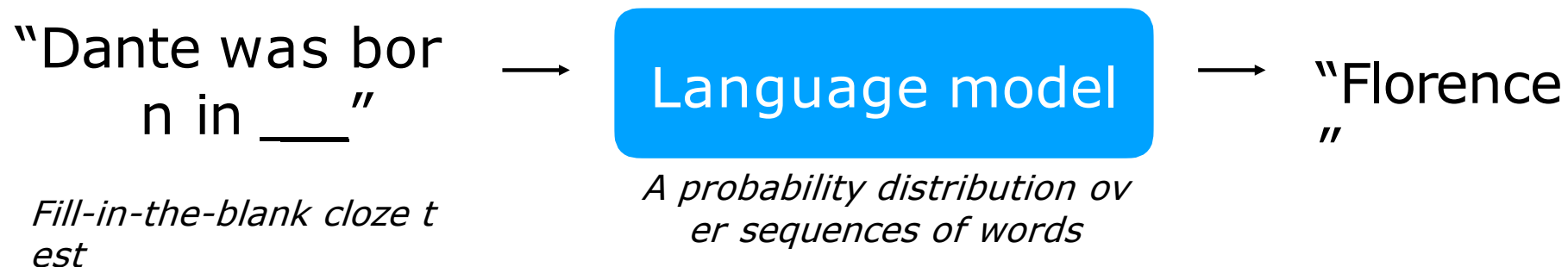
Autoregressive training:  
Predict the next word (token) based on previous words, e.  
g., GPT



Training data is huge (gigabytes of text) and contains diverse sources such as Wikipedia, news articles, books, and so on

# Language model

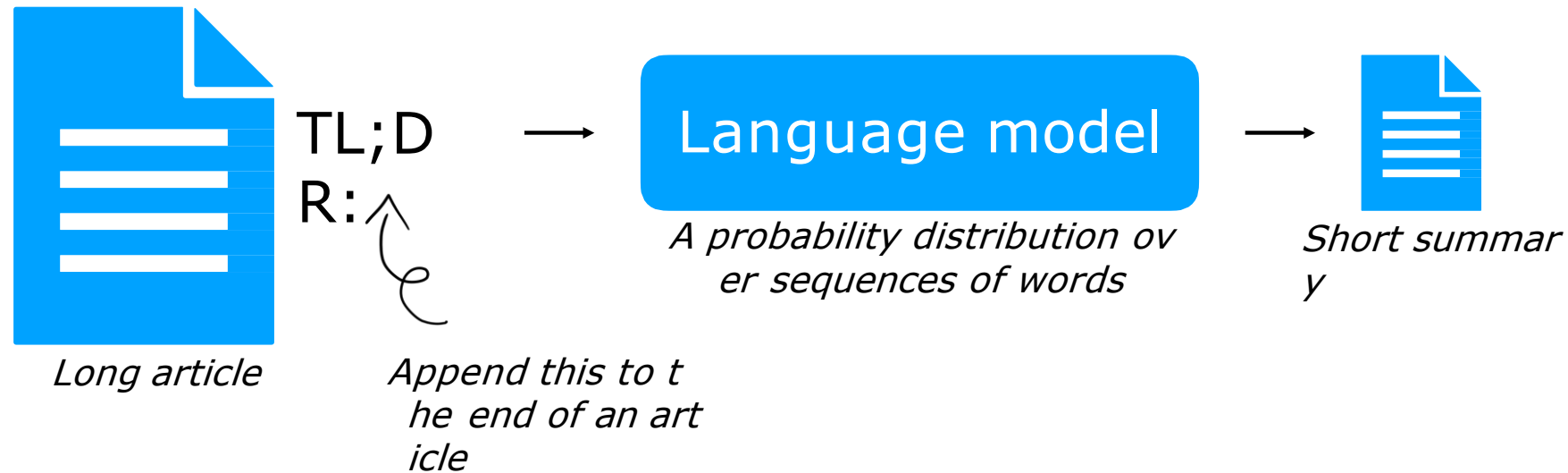
Prompting is used to elicit knowledge from pre-trained language models



**Idea: Convert input into a language modeling format**

# Language model

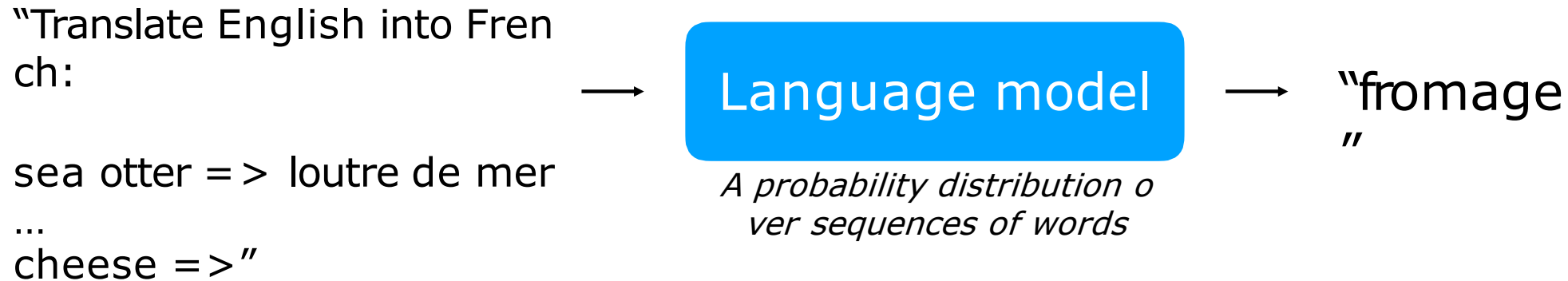
Prompting is used to elicit knowledge from pre-trained language models



**Idea: Convert input into a language modeling format**

# Language model

Prompting is used to elicit knowledge from pre-trained language models



*Task description + examples (in-context learning)*

**Idea: Convert input into a language modeling format**

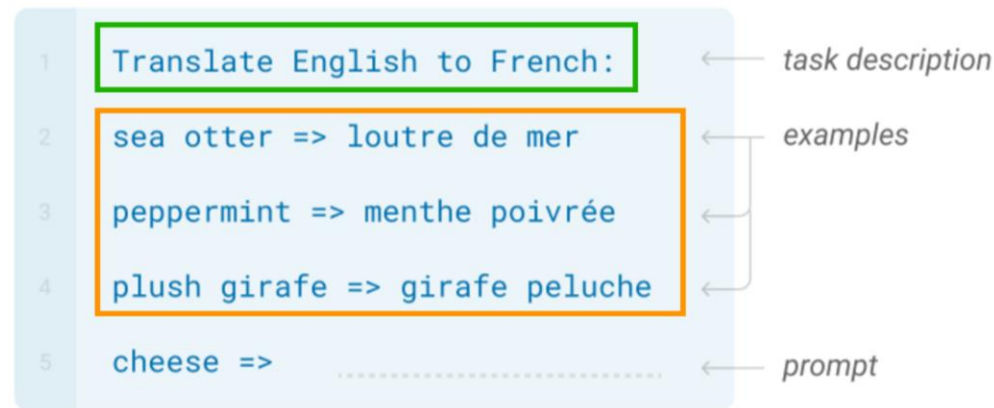


# Zero-Shot or Few-Shot

- Steer the behavior of language models for desired outcomes *without* updating the model weights

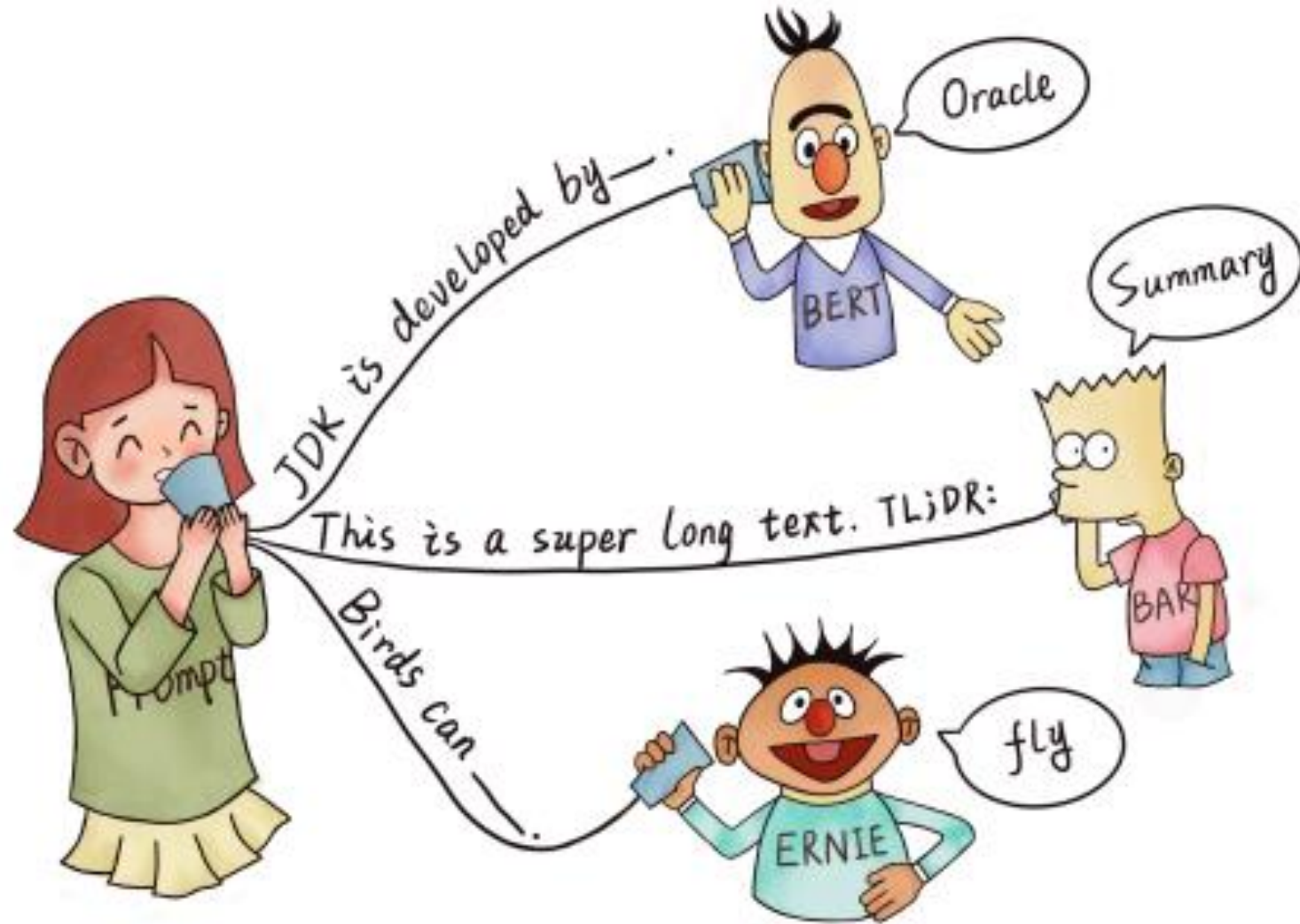
## Few-shot

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



Natural language task description + examples as demonstrations  
(No model update!!)

# Zero-Shot or Few-Shot

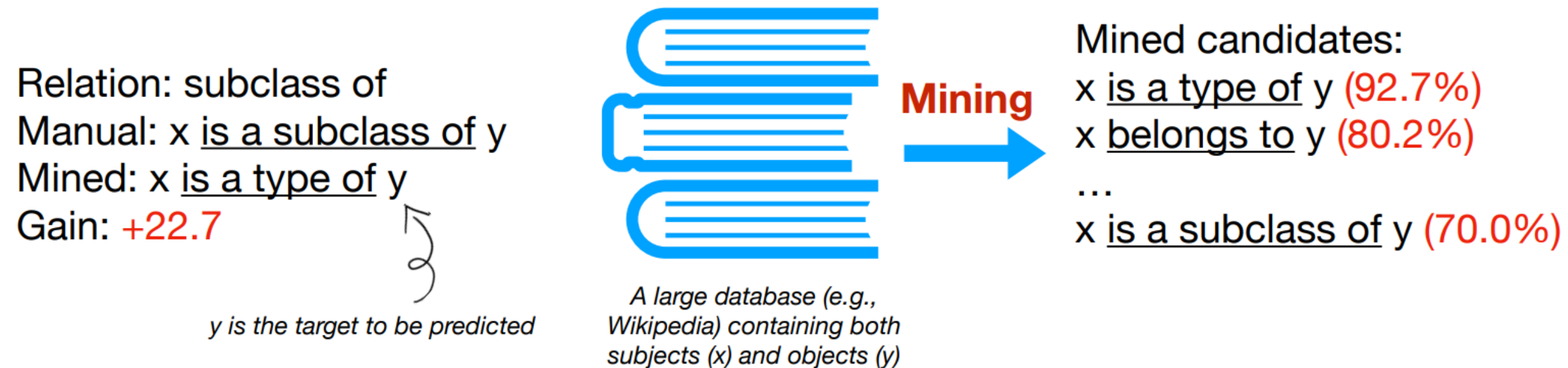


# Limitation

... but manually crafting a good prompt is **non-trivial**  
(a bad prompt might fail to retrieve the correct knowledge)

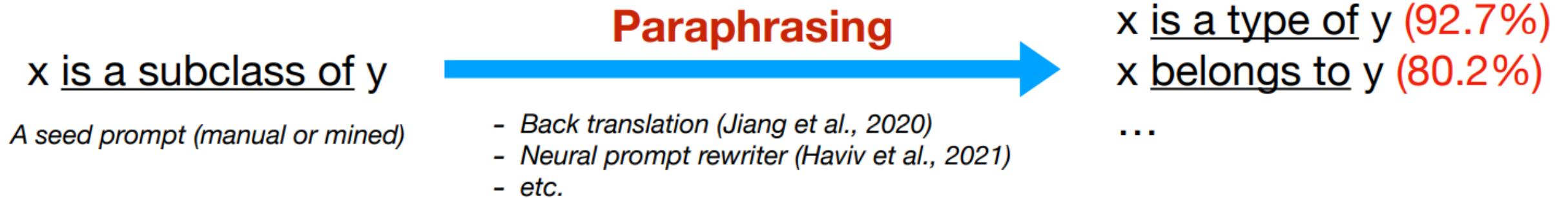
# Hard prompt

- Mining-based prompt generation



# Hard prompt

- Paraphrasing-based prompt generation



# Prompt engineering?

a bad photo of a {}.  
a photo of many {}.  
a sculpture of a {}.  
a photo of the hard to see {}.  
a low resolution photo of the {}.  
a rendering of a {}.  
graffiti of a {}.  
a bad photo of the {}.  
a cropped photo of the {}.  
a tattoo of a {}.  
the embroidered {}.  
a photo of a hard to see {}.  
a bright photo of a {}.  
a photo of a clean {}.  
a photo of a dirty {}.  
a dark photo of the {}.  
a drawing of a {}.  
a photo of my {}.  
the plastic {}.  
a photo of the cool {}.  
a close-up photo of a {}.  
a black and white photo of the {}.  
a painting of the {}.  
a painting of a {}.


a pixelated photo of the {}.  
a sculpture of the {}.  
a bright photo of the {}.  
a cropped photo of a {}.  
a plastic {}.  
a photo of the dirty {}.  
a jpeg corrupted photo of a {}.  
a blurry photo of the {}.  
a photo of the {}.  
a good photo of the {}.  
a rendering of the {}.  
a {} in a video game.  
a photo of one {}.  
a doodle of a {}.  
a close-up photo of the {}.  
a photo of a {}.  
the origami {}.  
the {} in a video game.  
a sketch of a {}.  
a doodle of the {}.  
a origami {}.  
a low resolution photo of a {}.  
the toy {}.  
a rendition of the {}.


a photo of the clean {}.  
a photo of a large {}.  
a rendition of a {}.  
a photo of a nice {}.  
a photo of a weird {}.  
a blurry photo of a {}.  
a cartoon {}.  
art of a {}.  
a sketch of the {}.  
a embroidered {}.  
a pixelated photo of a {}.  
itap of the {}.  
a jpeg corrupted photo of the {}.  
a good photo of a {}.  
a plushie {}.  
a photo of the nice {}.  
a photo of the small {}.  
a photo of the weird {}.  
the cartoon {}.  
art of the {}.  
a drawing of the {}.  
a photo of the large {}.  
a black and white photo of a {}.  
the plushie {}.

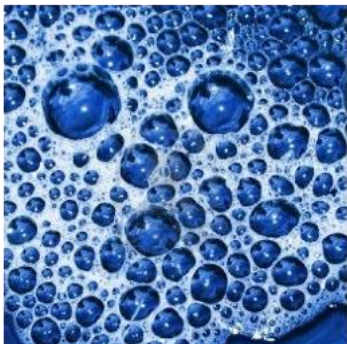
**A slight change in wording could lead to big changes in performance**

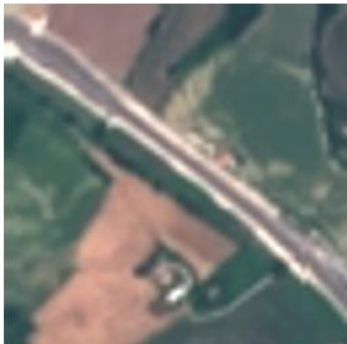


# Prompt engineering is also hard

Caltech101	Prompt	Accuracy
	a [CLASS].	82.68
	a photo of [CLASS].	80.81
	a photo of a [CLASS].	86.29
	$[V]_1 [V]_2 \dots [V]_M$ [CLASS].	<b>91.83</b>

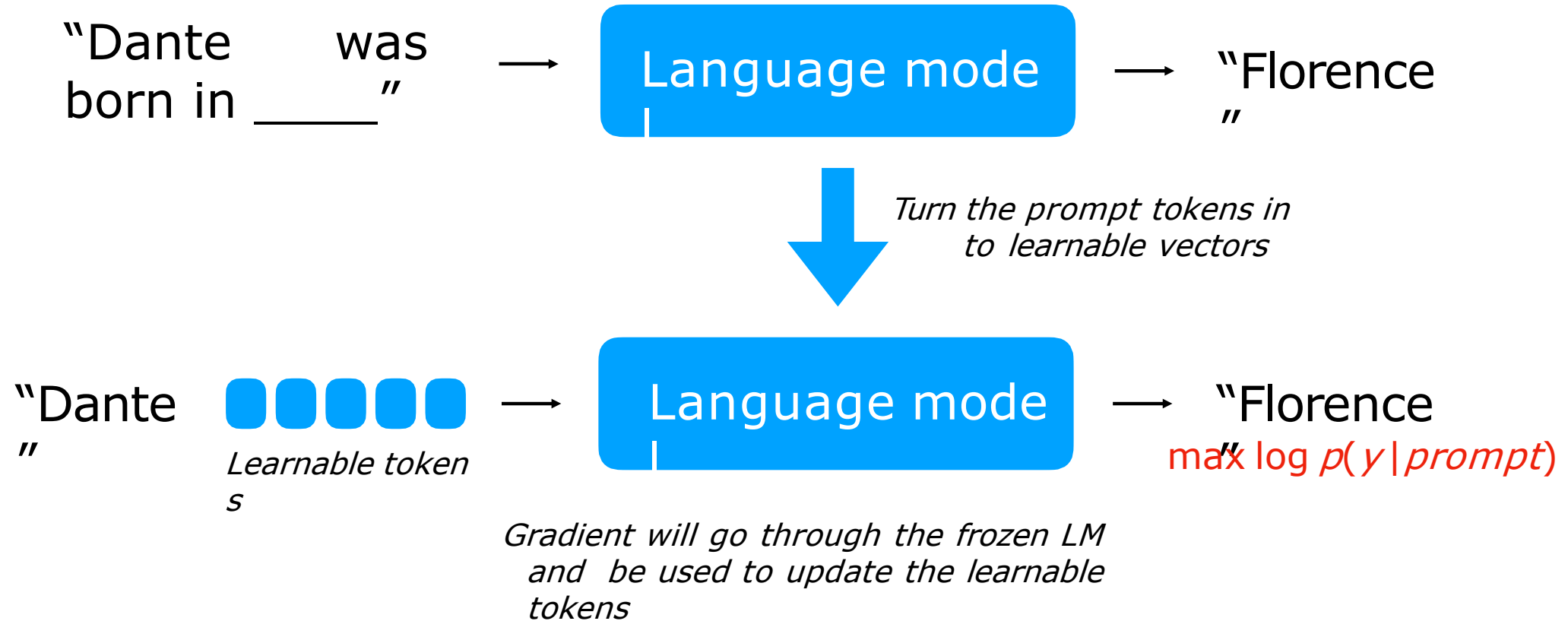
Flowers102	Prompt	Accuracy
	a photo of a [CLASS].	60.86
	a <b>flower</b> photo of a [CLASS].	65.81
	a photo of a [CLASS], a <b>type of flower</b> .	66.14
	$[V]_1 [V]_2 \dots [V]_M$ [CLASS].	<b>94.51</b>

Describable Textures (DTD)	Prompt	Accuracy
	a photo of a [CLASS].	39.83
	a photo of a [CLASS] <b>texture</b> .	40.25
	[CLASS] texture.	42.32
	$[V]_1 [V]_2 \dots [V]_M$ [CLASS].	<b>63.58</b>

EuroSAT	Prompt	Accuracy
	a photo of a [CLASS].	24.17
	a <b>satellite</b> photo of [CLASS].	37.46
	a centered satellite photo of [CLASS].	37.56
	$[V]_1 [V]_2 \dots [V]_M$ [CLASS].	<b>83.53</b>

**A slight change in wording could lead to big changes in performance**

# Soft prompt

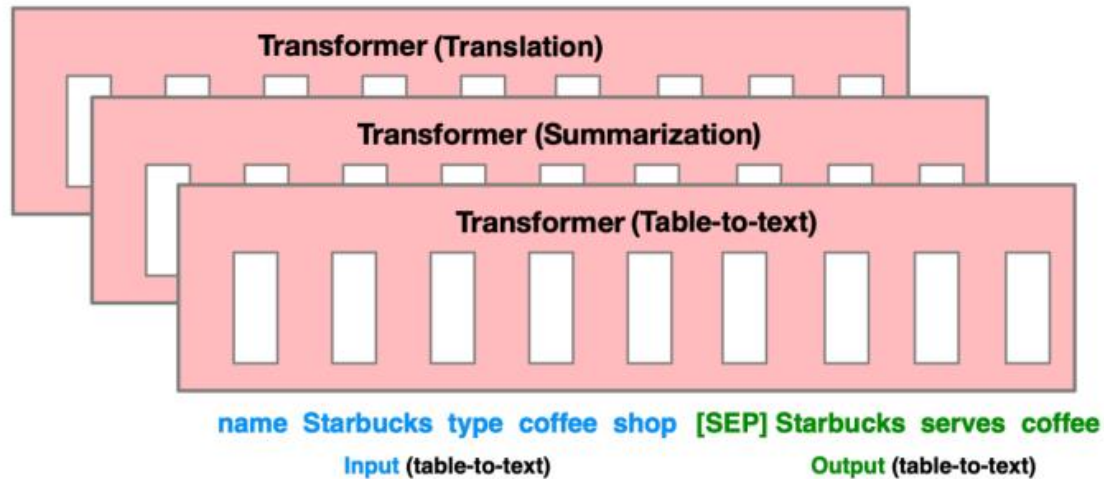




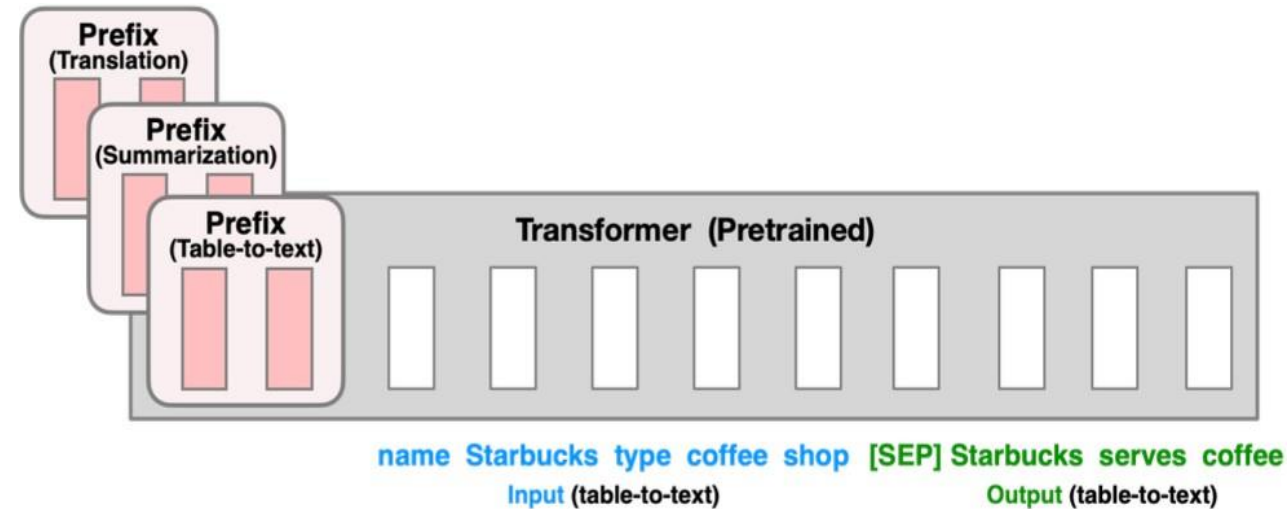
# Soft prompt

- Only learns task/user-specific prompt vectors
- Only needs to store these vectors for each task/user

- Fine-tuning



- Prefix-tuning



# Insights about soft prompt in NLP

- Can handle low-data regimes

With only 0.1% parameters

summarization  
(higher is better)

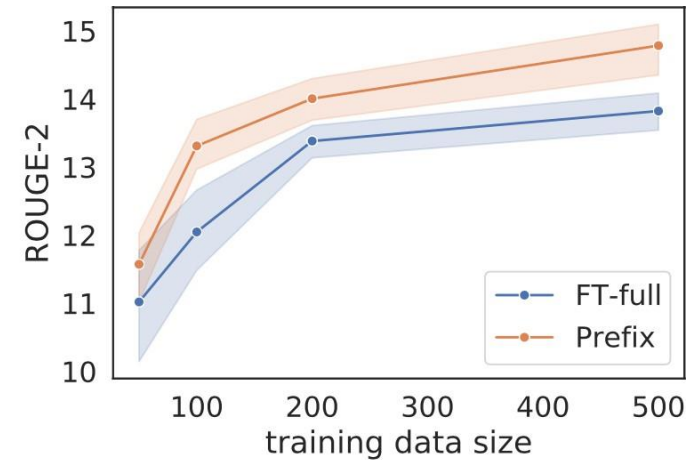
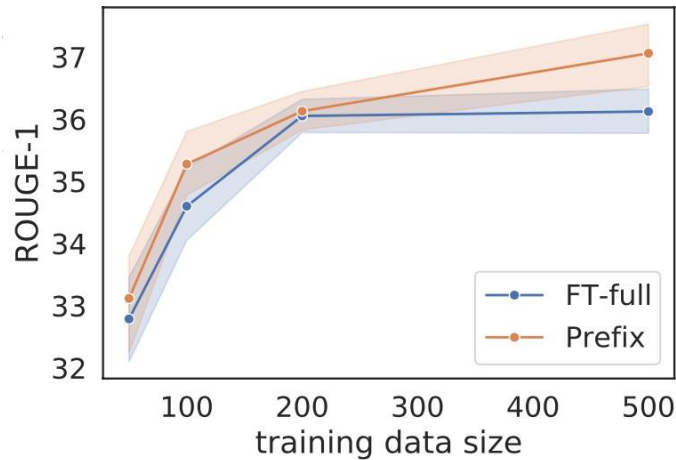
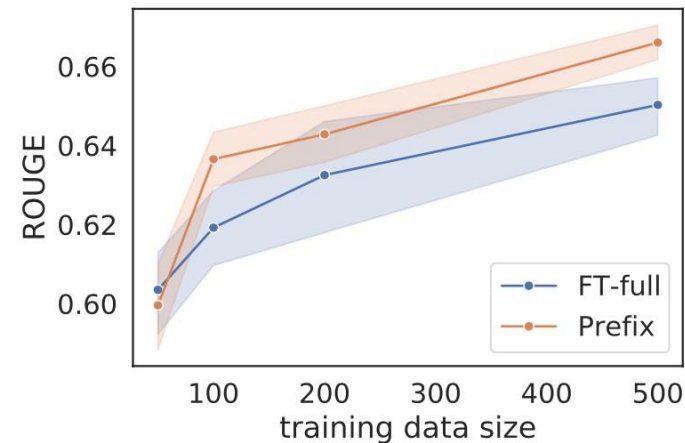
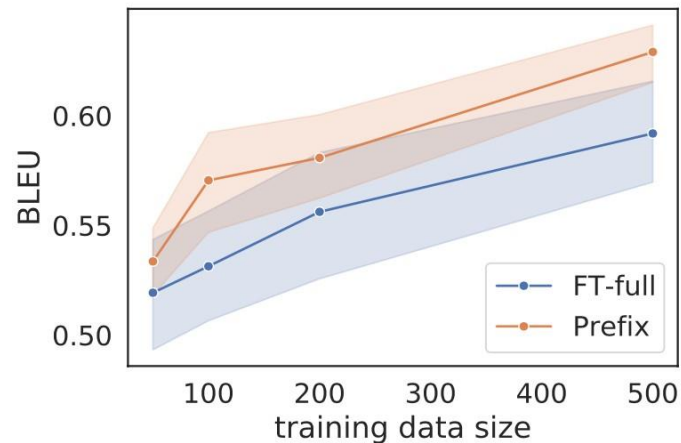


table-to-text  
(higher is better)



# Insights about soft prompt in NLP

- Is domain-generalizable

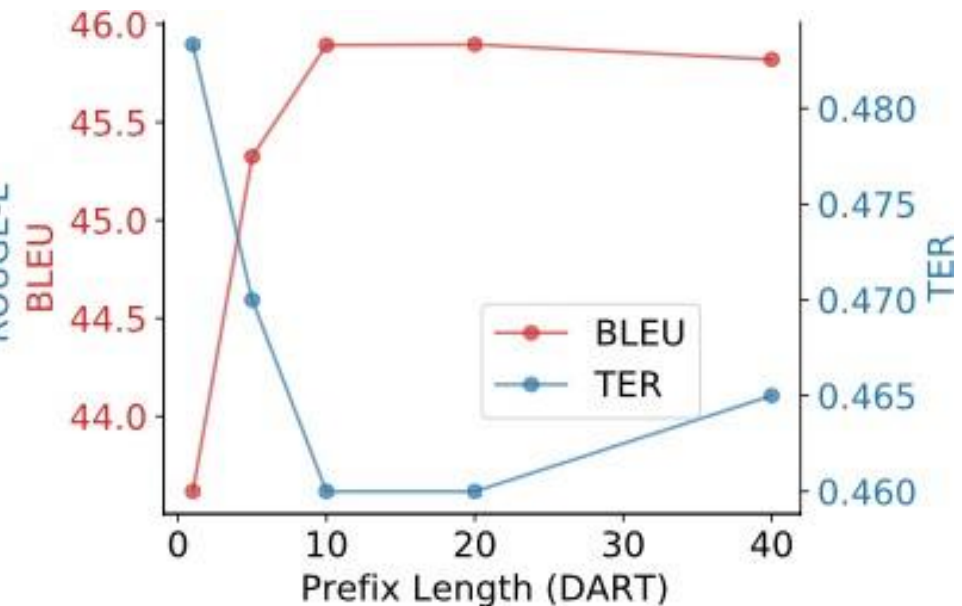
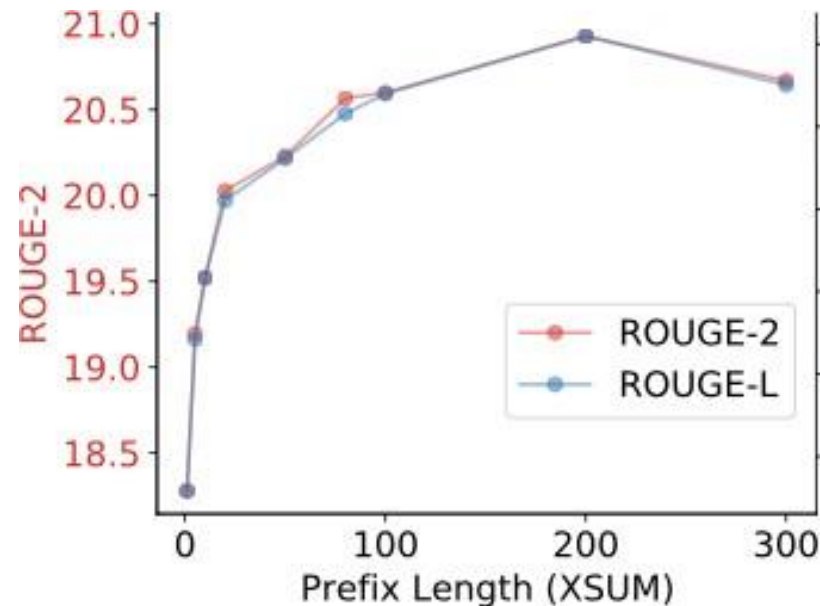
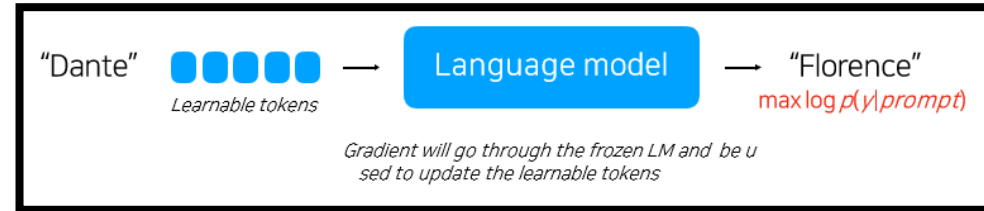
F1 score

Dataset	Domain	Model	Prompt	$\Delta$
SQuAD	Wiki	94.9 $\pm$ 0.2	94.8 $\pm$ 0.1	-0.1
TextbookQA	Book	54.3 $\pm$ 3.7	<b>66.8</b> $\pm$ 2.9	+12.5
BioASQ	Bio	77.9 $\pm$ 0.4	<b>79.1</b> $\pm$ 0.3	+1.2
RACE	Exam	59.8 $\pm$ 0.6	<b>60.7</b> $\pm$ 0.5	+0.9
RE	Wiki	88.4 $\pm$ 0.1	<b>88.8</b> $\pm$ 0.2	+0.4
DuoRC	Movie	<b>68.9</b> $\pm$ 0.7	67.7 $\pm$ 1.1	-1.2
DROP	Wiki	<b>68.9</b> $\pm$ 1.7	67.1 $\pm$ 1.9	-1.8

*"Prompt tuning tends to give stronger zero-shot performance than model tuning, especially on datasets with large domain shifts like TextbookQA."*

# Insights about soft prompt in NLP

- Longer prompt works better but should not be too long



# Background

## Visual prompt

# Visual Prompt Tuning

Menglin Jia<sup>\*1,2</sup>, Luming Tang<sup>\*1</sup>  
Bor-Chun Chen<sup>2</sup>, Claire Cardie<sup>1</sup>, Serge Belongie<sup>3</sup>  
Bharath Hariharan<sup>1</sup>, and Ser-Nam Lim<sup>2</sup>

<sup>1</sup>Cornell University

<sup>2</sup>Meta AI

<sup>3</sup>University of Copenhagen

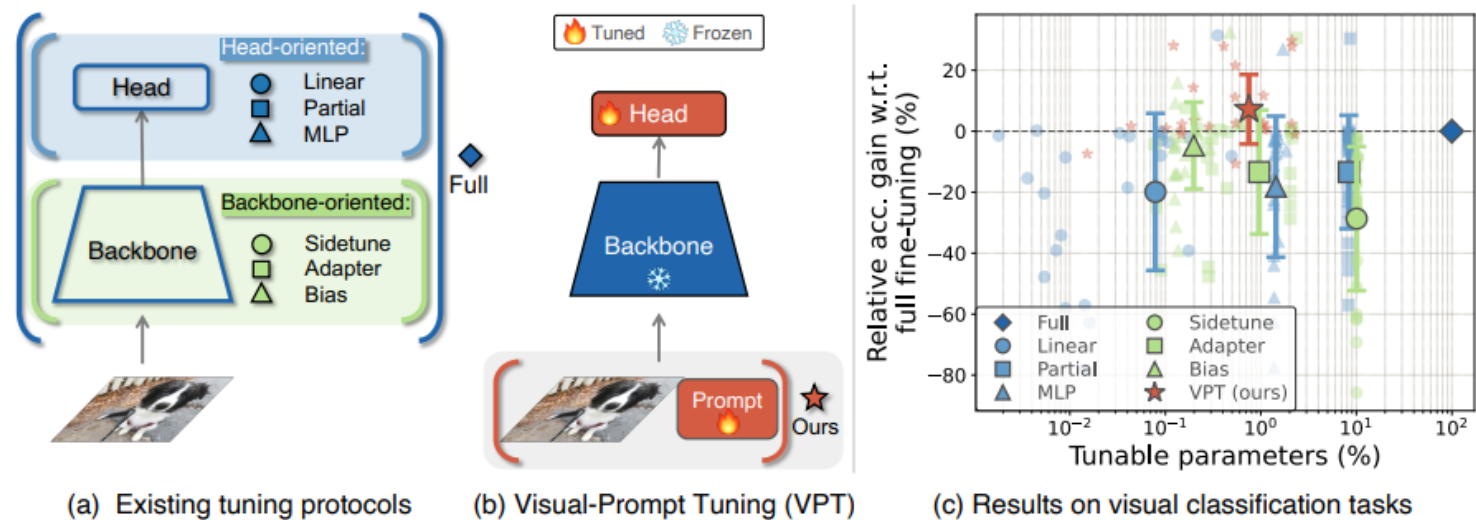
# Abstract

- 현재 전이 학습 방식: 모델 전체 파라미터를 업데이트 (전체 미세 조정).
- 시각적 프롬프트 튜닝 (VPT): 비전에서 대규모 트랜스포머 모델에 대한 전체 미세 조정의 효율적이고 효과적인 대안으로 제시.
- 큰 언어 모델을 효율적으로 튜닝하는 최근의 발전에서 영감을 얻음.
- VPT는 입력 공간에서 매우 작은 양의 학습 가능한 파라미터를 도입하면서 모델 백본을 고정.
- 광범위한 실험을 통해, VPT가 다른 파라미터 효율적인 튜닝 방식에 비해 큰 성능 향상을 보임.
- 가장 중요하게, VPT는 모델 용량과 학습 데이터 규모에 걸쳐 많은 경우에 전체 미세 조정보다 성능이 우수.
- 작업 별 저장 비용도 줄임.

• Introduction	• Methods	• Results	• Conclusion
----------------	-----------	-----------	--------------

- 다양한 인식 애플리케이션에서는 대규모로 사전 훈련된 기반이 되는 모델을 적용함으로써 가장 정확한 결과를 얻는다. 이는 자연어 처리(NLP)에서의 발전과 유사하다.
- 이런 접근법은 여러 인식 문제에서 빠른 진전을 이룰 수 있음을 의미한다.
- 그러나 이런 큰 모델들을 하위 작업에 맞게 조정하는 것은 도전적이다.
- 가장 흔한 전략은 사전 훈련된 모델을 해당 작업에 완전히 미세 조정하는 것이다.
- 하지만 이 전략은 각 작업마다 별도의 모델 파라미터 복사본을 저장하고 배포해야 하므로 비용이 많이 든다.
- 현대의 Transformer 기반 아키텍처는 ConvNet보다 크기가 훨씬 크다. (예: ViT-Huge는 632M 파라미터, ResNet-50은 25M 파라미터).
- 따라서, 효과와 효율성 면에서 큰 사전 훈련된 Transformer를 하위 작업에 어떻게 조정하는 것이 최선인지를 고민한다.





- 파라미터의 하위 집합만을 미세 조정하는 방법으로 ConvNets에는 인기가 있었다. → classifier head, bias term, adapter을 추가하는 것을 고려했다.
- 그러나, Transformer에 대해 유사한 전략을 구현할 수 있지만, 일반적으로 이러한 전략들은 전체 미세 조정의 정확도보다 성능이 떨어진다.
- NLP의 최근 Prompting 기술에서 영감을 받아, 다운스트림 시각 작업을 위해 Transformer 모델을 조정하는 새로운 간단하고 효율적인 방법, 즉 Visual-Prompt Tuning (VPT)을 제안

• Introduction	• Methods	• Results	• Conclusion
----------------	-----------	-----------	--------------

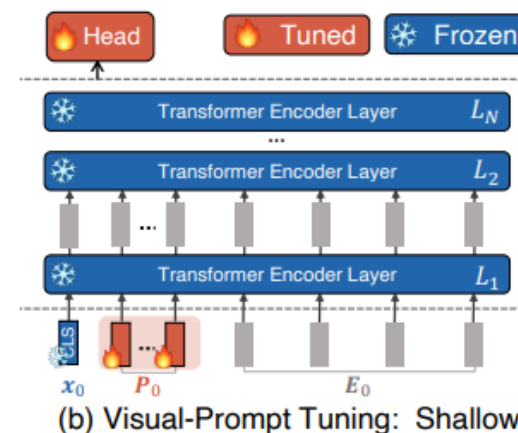
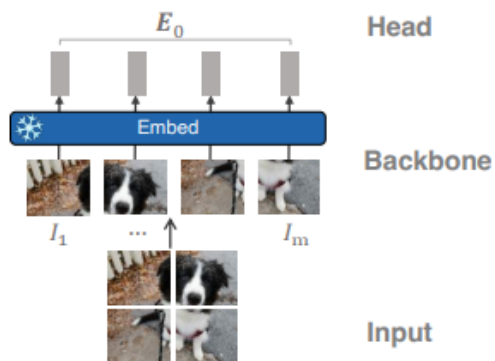
## • Transfer learning

- 비전 Transformer의 적응에 대한 관심은 상대적으로 적었으며, 이 새로운 유형의 아키텍처에서 언급된 방법들이 얼마나 잘 수행되는지는 알려져 있지 않다.
- 큰 규모의 사전 훈련된 Transformer 기반 언어 모델의 지배력을 고려할 때, 많은 방법들이 다양한 다운스트림 NLP 작업을 위해 LM을 효율적으로 미세 조정하기 위해 제안되었다.
- 우리의 연구는 VPT가 일반적으로 NLP에서 잘 정립된 두 가지 방법에 비해 Transformer 모델을 비전 작업에 적응시키는 데 더 향상된 성능을 제공함을 보여준다.

## • Prompting

- Prompting은 사전 훈련된 LM이 작업을 “이해”할 수 있도록 입력 텍스트에 언어 지시사항을 앞에 붙이는 것을 원래 의미한다.
- 미세 조정 중에 그라디언트를 통해 직접 최적화하여 프롬프트를 작업별 연속 벡터로 취급하는 최근의 작업들이 제안되었다.
- 프롬프트 튜닝은 전체 미세 조정에 비해 비교적 성능은 동일하지만 파라미터 저장 공간이 1000배 적다.
- 프롬프팅은 최근 비전-언어 모델에도 적용되었지만, 텍스트 인코더의 입력에만 한정되어 있다

- Visual-Prompt Tuning (VPT)



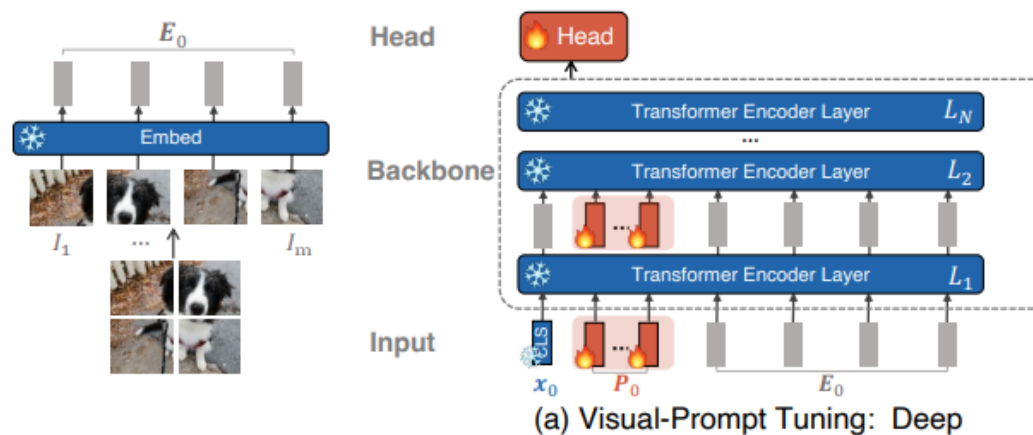
**VPT-Shallow.** Prompts are inserted into the first Transformer layer  $L_1$  only. Each prompt token is a learnable  $d$ -dimensional vector. A collection of  $p$  prompts is denoted as  $\mathbf{P} = \{\mathbf{p}^k \in \mathbb{R}^d \mid k \in \mathbb{N}, 1 \leq k \leq p\}$ , the shallow-prompted ViT is:

$$[\mathbf{x}_1, \mathbf{Z}_1, \mathbf{E}_1] = L_1([\mathbf{x}_0, \mathbf{P}, \mathbf{E}_0]) \quad (4)$$

$$[\mathbf{x}_i, \mathbf{Z}_i, \mathbf{E}_i] = L_i([\mathbf{x}_{i-1}, \mathbf{Z}_{i-1}, \mathbf{E}_{i-1}]) \quad i = 2, 3, \dots, N \quad (5)$$

$$\mathbf{y} = \text{Head}(\mathbf{x}_N) \quad , \quad (6)$$

- Visual-Prompt Tuning (VPT)



**VPT-Deep.** Prompts are introduced at *every* Transformer layer's input space. For  $(i+1)$ -th Layer  $L_{i+1}$ , we denote the collection of input learnable prompts as  $\mathbf{P}_i = \{\mathbf{p}_i^k \in \mathbb{R}^d \mid k \in \mathbb{N}, 1 \leq k \leq m\}$ . The deep-prompted ViT is formulated as:

$$[\mathbf{x}_i, \_, \mathbf{E}_i] = L_i([\mathbf{x}_{i-1}, \mathbf{P}_{i-1}, \mathbf{E}_{i-1}]) \quad i = 1, 2, \dots, N \quad (7)$$

$$\mathbf{y} = \text{Head}(\mathbf{x}_N) \quad (8)$$

• Introduction	• Methods	• Results	• Conclusion
----------------	-----------	-----------	--------------

- **Storing Visual Prompts.**

- VPT는 여러 다운스트림 작업이 있을 때 유용하다.
- 각 작업에 대해 학습된 프롬프트와 분류 헤드만 저장하고 사전 훈련된 Transformer 모델의 원본 복사본을 재사용함으로써 저장 비용을 크게 줄인다.
- 예를 들어, 86 million 파라미터와  $d = 768$ 을 갖는 ViT-Base가 주어지면, 50개의 얇은 프롬프트와 깊은 프롬프트는 추가로  $p \times d = 50 \times 768 = 0.038M$ , 그리고  $N \times p \times d = 0.46M$  파라미터를 생성하며, 이는 각각 ViT-Base 파라미터의 0.04%와 0.53%에 해당한다.

## 4. Experiments Setup - Pre-trained Backbones

- 사전 훈련된 기반 모델: Vision Transformers (ViT)와 Swin Transformers 두 가지 Transformer 아키텍처로 실험. 모든 기반 모델은 ImageNet-21k에서 사전 훈련됨.
- 기준선 (Baselines): VPT의 두 가지 변형을 다음과 같은 일반적으로 사용되는 미세 조정 프로토콜과 비교한다:
  - Full: 모든 기반 및 분류 헤드 파라미터를 완전히 업데이트한다.
  - 분류 헤드에 중점을 둔 방법: 사전 훈련된 기반을 특징 추출기로 취급하고, 가중치는 튜닝 중에 고정된다:
    - Linear: 분류 헤드로 선형 레이어만 사용한다.
    - Partial-k: 모델의 마지막 k 레이어를 미세 조정하고 나머지는 고정한다.
    - MLP-k: 분류 헤드로 선형 레이어 대신 k 레이어를 가진 MLP를 사용한다.
  - 미세 조정 중에 일부 기반 파라미터를 업데이트하거나 기반에 새로운 학습 가능한 파라미터를 추가하는 방법:
    - Sidetune: “사이드” 네트워크를 훈련시키고 헤드로 공급되기 전에 사전 훈련된 특징과 사이드 튜닝된 특징 사이를 선형 보간한다.
    - Bias: 사전 훈련된 기반의 편향 항만 미세 조정한다.
    - Adapter: Transformer 계층 내부에 잔류 연결을 가진 새로운 MLP 모듈을 삽입한다.

## 4. Experiments Setup - Downstream Tasks

- FGVC는 CUB-200-2011, NABirds, Oxford Flowers, Stanford Dogs, Stanford Cars을 포함한 5개의 세분화된 시각 분류 작업으로 구성되어 있다.
- 특정 데이터셋이 공개적으로 훈련 세트와 테스트 세트만을 가지고 있는 경우, 훈련 세트를 훈련(90%)과 검증(10%)으로 무작위로 분할하고, 검증 세트를 사용하여 하이퍼파라미터를 선택한다.
- VTAB-1k [86]는 세 가지 그룹으로 구성된 19가지 다양한 시각 분류 작업의 모음:
  - Natural - 표준 카메라를 사용하여 촬영된 자연 이미지가 포함된 작업
  - Specialized - 의료 및 위성 이미지와 같은 전문 장비로 촬영된 이미지가 포함된 작업
  - Structured - 객체 카운팅과 같은 기하학적 이해가 필요한 작업

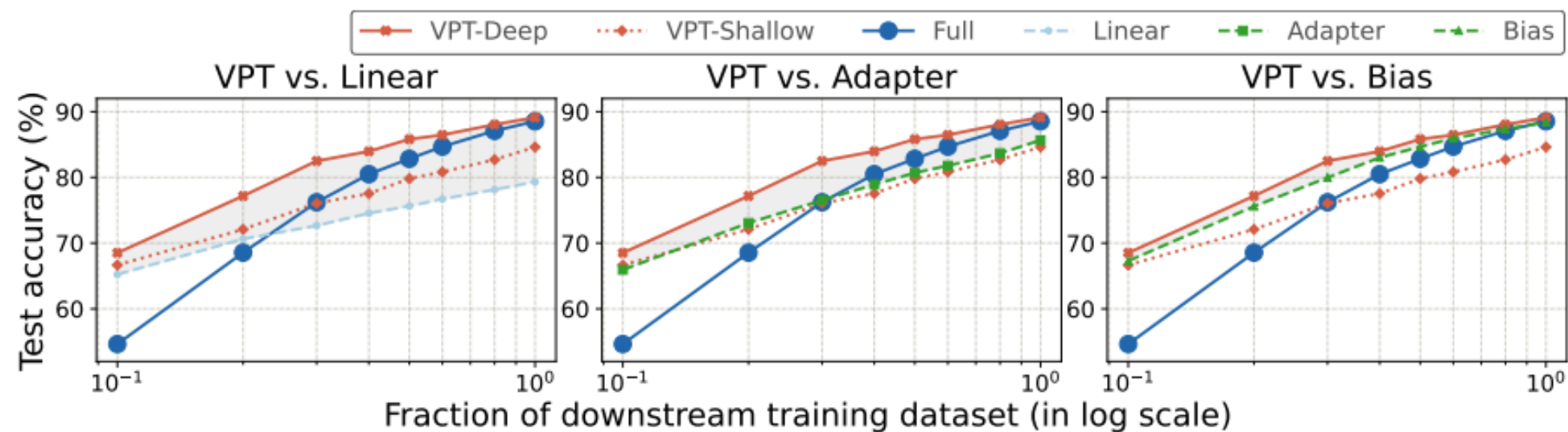
4.2 Main Results

**Table 1.** ViT-B/16 pre-trained on supervised ImageNet-21k. For each method and each downstream task group, we report the average test accuracy score and **number of wins in (·)** compared to FULL. “Total params” denotes total parameters needed for all 24 downstream tasks. “Scope” denotes the tuning scope of each method. “Extra params” denotes the presence of additional parameters besides the pre-trained backbone and linear head. Best results among all methods except FULL are **bolded**. VPT outshines the full fine-tuning 20 out of 24 cases with significantly less trainable parameters

	ViT-B/16 (85.8M)	Total params	Scope		Extra params	FGVC	VTAB-1k		
			Input	Backbone			Natural	Specialized	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×	✓			<b>89.11 (4)</b>	<b>78.48 (6)</b>	<b>82.43 (2)</b>	<b>54.98 (8)</b>

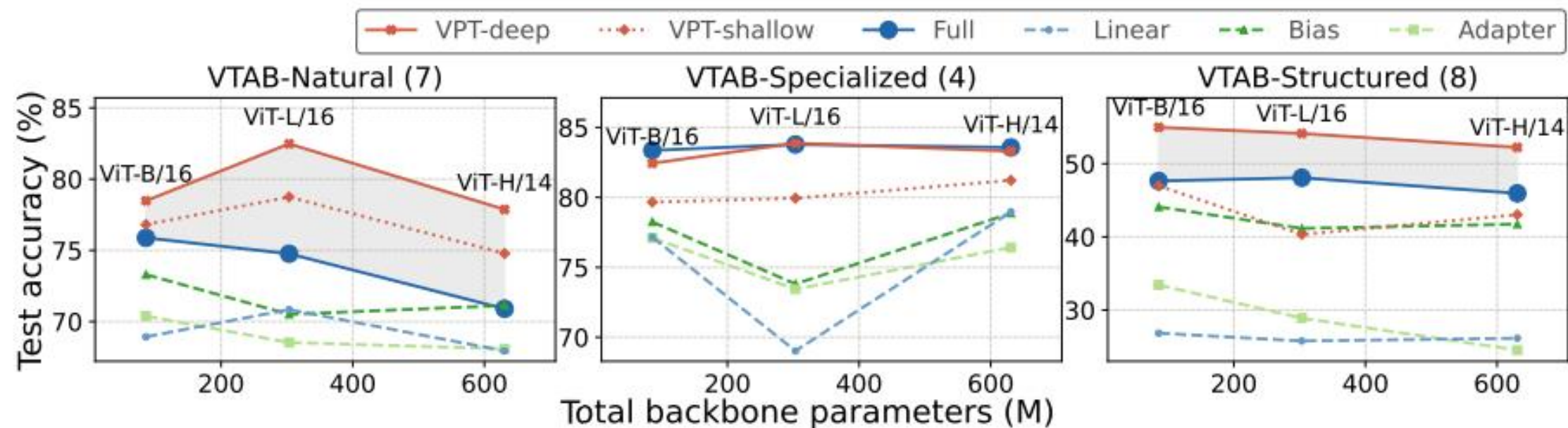


## 4.2 Main Results



**Fig. 3.** Performance comparison on different downstream data scales, averaged across 5 FGVC tasks. VPT-DEEP is compared with LINEAR (left), ADAPTER (middle) and BIAS (right). Highlighted region shows the accuracy difference between VPT-DEEP and the compared method. Results of VPT-SHALLOW are FULL presented in all plots for easy reference. The size of markers are proportional to the percentage of tunable parameters in log scale

## 4.2 Main Results



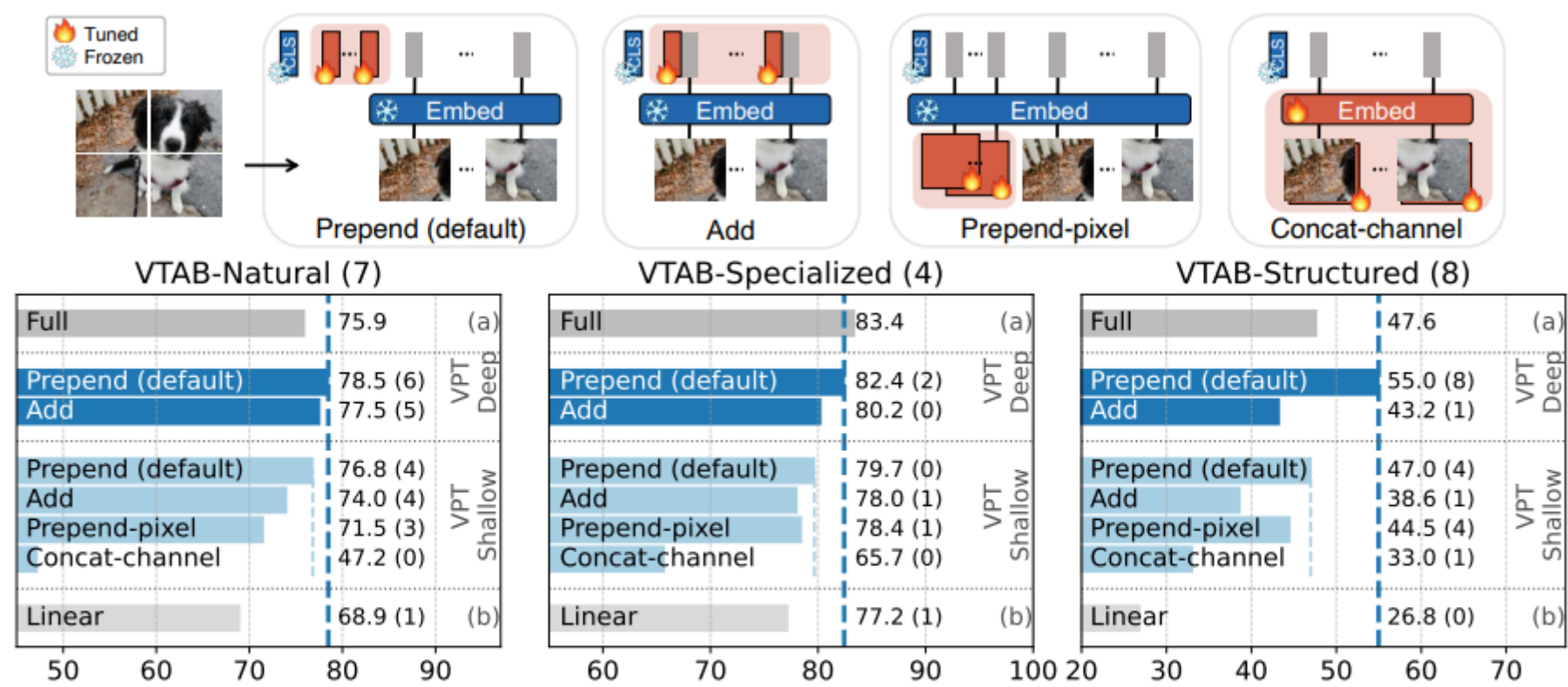
**Fig. 4.** VPT *vs.* FULL across model scales (ViT-B, ViT-L and ViT-H), for 3 VTAB task groups. Highlighted region shows the accuracy difference between VPT-DEEP and the full fine-tuning (FULL). The size of markers are proportional to the percentage of trainable parameters in log scale

4.2 Main Results

**Table 2.** Different Transformer architecture: Swin-B pre-trained on supervised ImageNet-21k as backbone. For each method and each downstream task group, we report the average test accuracy score and **number of wins** in (·) compared to FULL. The column “Total params” denotes total parameters needed for all 19 downstream tasks. Best results among all methods except FULL are **bolded**

	Swin-B (86.7M)	Total params	Natural	VTAB-1k Specialized	Structured
	Total # of tasks		7	4	8
(a)	FULL	19.01×	79.10	86.21	59.65
(b)	LINEAR	1.01×	73.52 (5)	80.77 (0)	33.52 (0)
	MLP-3	1.47×	73.56 (5)	75.21 (0)	35.69 (0)
	PARTIAL	3.77×	73.11 (4)	81.70 (0)	34.96 (0)
(c)	BIAS	1.06×	74.19 (2)	80.14 (0)	42.42 (0)
(ours)	VPT-SHALLOW	1.01×	<b>79.85 (6)</b>	82.45 (0)	37.75 (0)
	VPT-DEEP	1.05×	76.78 (6)	<b>84.53 (0)</b>	<b>53.35 (0)</b>

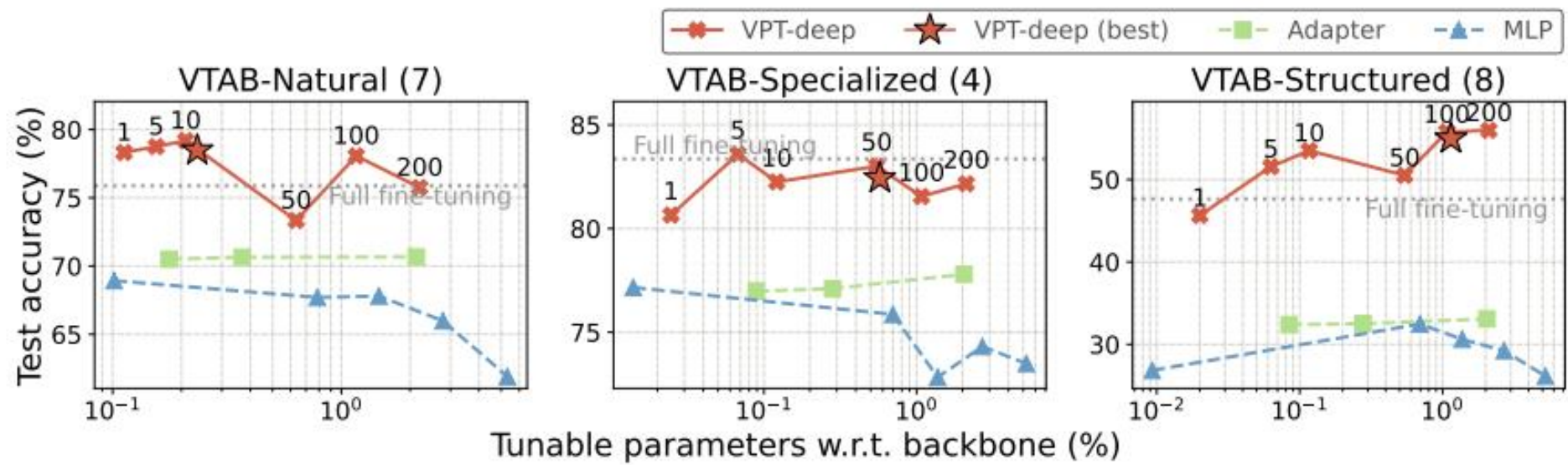
4.3 Ablation on Model Design Variants



**Fig. 5.** Ablation on prompt location. We illustrate different location choices at top, and present the results at bottom. For easy comparison, two blue dashed lines represent the performance of the default VPT-DEEP and VPT-SHALLOW respectively

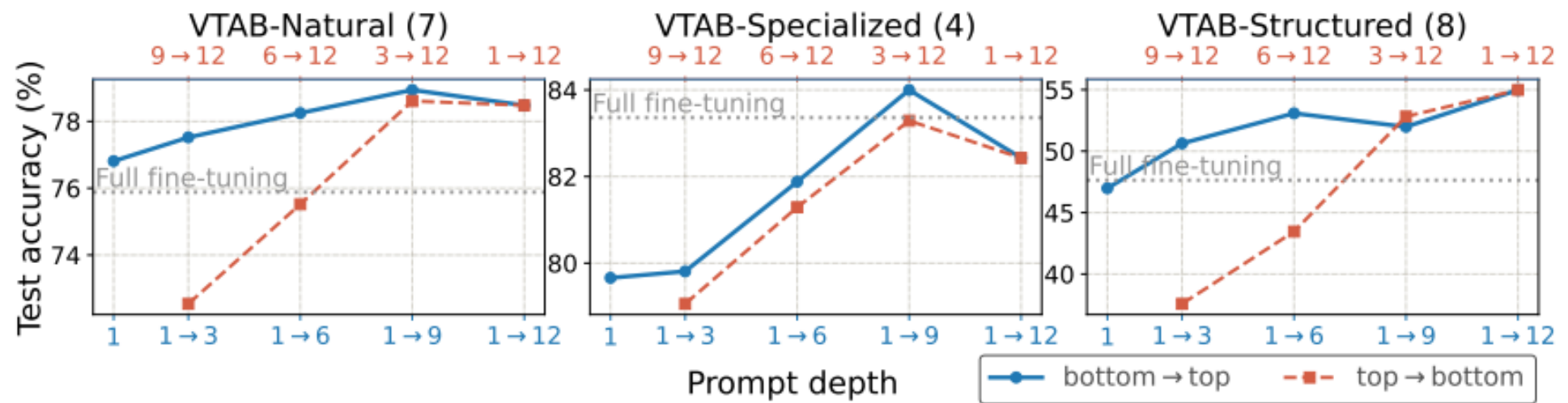


### 4.3 Ablation on Model Design Variants



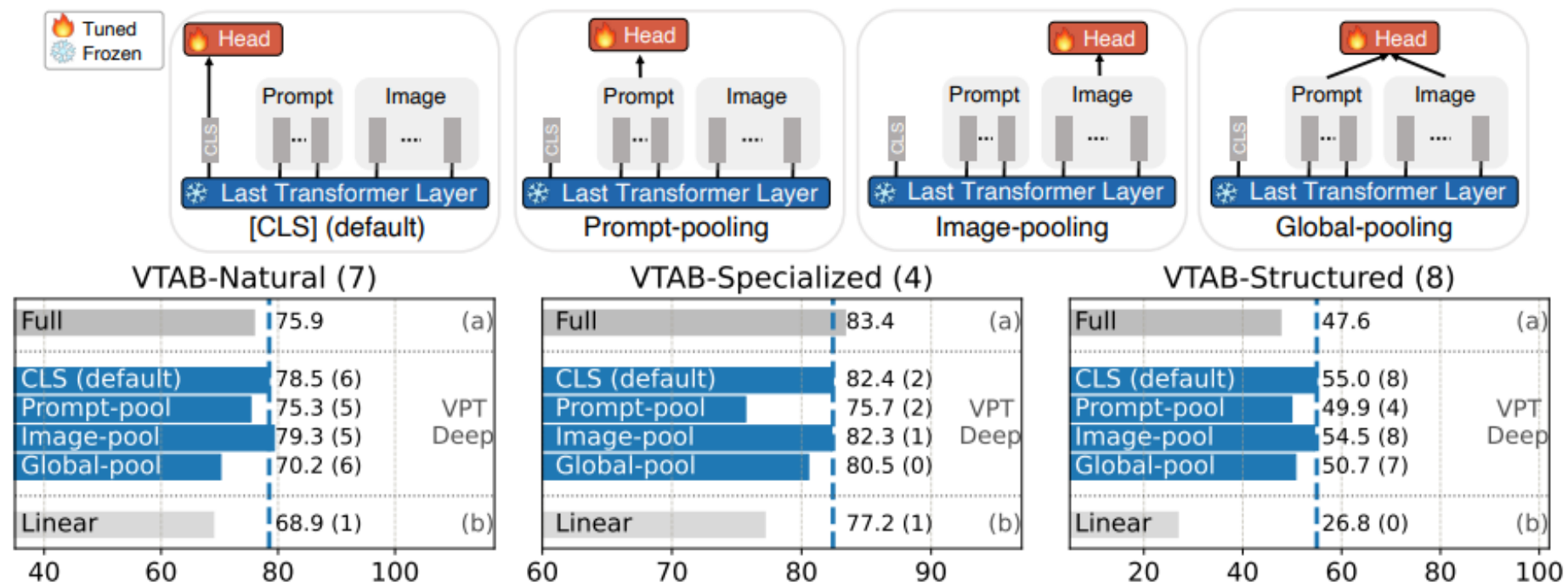
**Fig. 6.** Ablation on prompt length. We vary the number of prompts for VPT-DEEP and show the averaged results for each VTAB subgroup. The averaged best VPT-DEEP results for each task is also shown for easy reference

### 4.3 Ablation on Model Design Variants



**Fig. 7.** Ablation on prompt depth. We select the best prompt length for each variant with `val` sets.  $i \rightarrow j$  indicates the Transformer layer indices that prompts are inserted into. The 1-st layer refers to the one closest to input. ViT-B has 12 layers in total

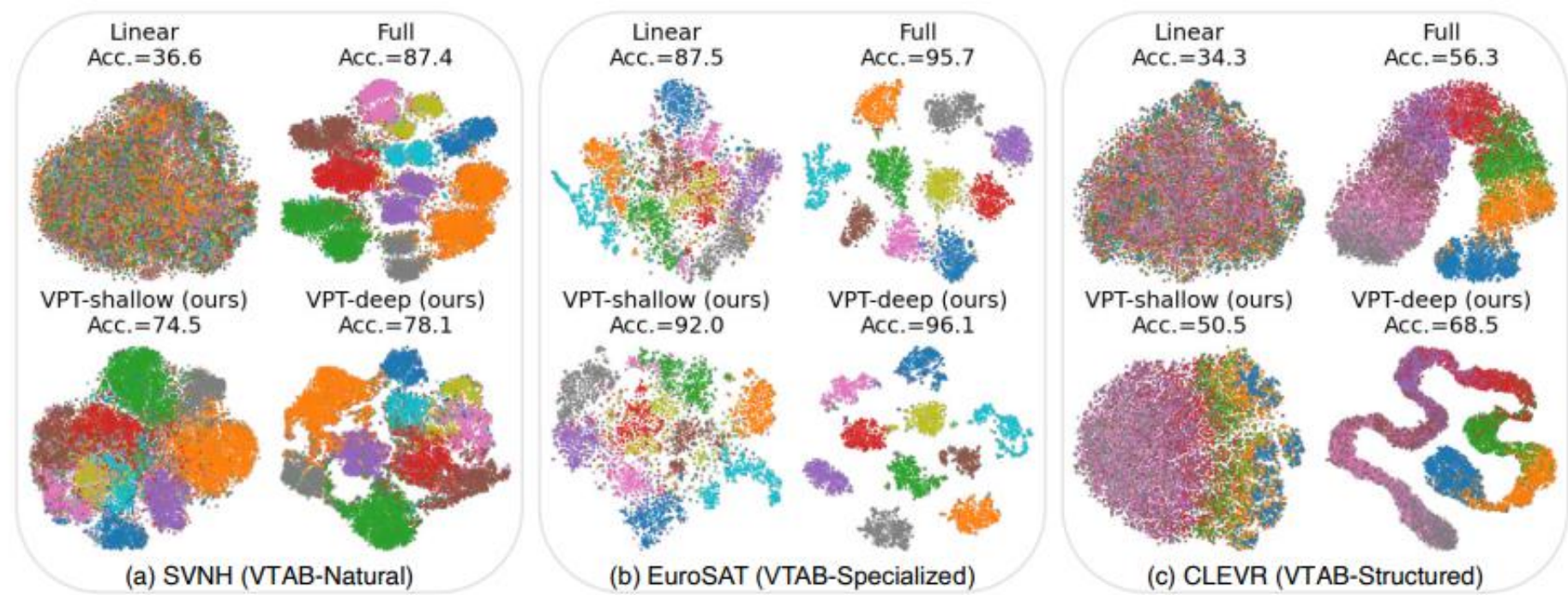
4.3 Ablation on Model Design Variants



**Fig. 8.** Ablation on final output. Illustration of different strategies is included at top, and results of those are presented at the bottom section. For easy comparison, the blue dashed line represents the performance of default VPT-DEEP



### 4.3 Ablation on Model Design Variants



**Fig. 9.** t-SNE visualizations of the final [CLS] embedding  $\mathbf{x}_N$  of 3 VTAB tasks from the **test** set, from Tab. 1. VPT could produce linearly separable features without updating backbone parameters



4.3 Ablation on Model Design Variants

**Table 3.** Semantic Segmentation: ADE20k [90] validation results with SETR [89] on ViT-L. The best mIoU scores among all methods but FULL are **bolded**. Results of fully fine-tuning a ResNet-101 [10] are included. SS/MS: single/multi-scale inference

Backbone	ViT-L/16					ResNet-101
Method	FULL [89]	HEAD ONLY	BIAS	VPT-DEEP	VPT+BIAS	FULL [10]
mIoU-SS	48.31	35.12	43.40	42.11	<b>44.04</b>	45.47
mIoU-MS	50.07	37.46	45.33	44.06	<b>45.63</b>	46.27
Tunable params (M)	318.31	13.18	13.46	13.43	15.79	63.0

4.3 Ablation on Model Design Variants

**Table 4.** Different pre-trained objectives: MAE [30] and MoCo v3 [11] with a ViT-B backbone. For each method and each downstream task group, we report the average test accuracy score and number of wins in (·) compared to FULL. “Total params” denotes total parameters needed for all 24 downstream tasks. Best results among all methods except FULL are **bolded**

		MAE				MoCo v3			
	ViT-B/16 (85.8M)	Total params	Natural	Specialized	Structured	Total params	Natural	Specialized	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×	<b>58.44 (5)</b>	<b>78.28 (1)</b>	47.64 (1)	2.58×	72.31 (5)	<b>84.58 (2)</b>	47.89 (1)
(c)	BIAS	1.03×	54.55 (1)	75.68 (1)	<b>47.70 (0)</b>	1.03×	72.89 (3)	81.14 (0)	<b>53.43 (4)</b>
	ADAPTER	1.17×	54.90 (3)	75.19 (1)	38.98 (0)	1.22×	<b>74.19 (4)</b>	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)

### 4.3 Ablation on Model Design Variants

**Table 5.** Apply VPT to ConvNets: ResNet-50 and ConvNeXt-Base. For each method and each downstream task group, we report the average test accuracy score and **number of wins in (·)** compared to FULL. “Total params” denotes total parameters needed for all 19 downstream tasks. Best results among all methods except FULL are **bolded**

		ConvNeXt-Base (87.6M)				ResNet-50 (23.5M)			
		Total params	Natural	VTAB-1k Specialized	Structured	Total params	Natural	VTAB-1k Specialized	Structured
Total # of tasks			7	4	8		7	4	8
(a)	FULL	19.01×	77.97	83.71	60.41	19.08×	59.72	76.66	54.08
(b)	LINEAR	1.01×	74.48 (5)	81.50 (0)	34.76 (1)	1.08×	63.75 (6)	77.60 (3)	30.96 (0)
	PARTIAL-1	2.84×	73.76 (4)	81.64 (0)	39.55 (0)	4.69×	64.34 (6)	<b>78.64</b> (2)	<b>45.78</b> (1)
	MLP-3	1.47×	73.78 (5)	81.36 (1)	35.68 (1)	7.87×	61.79 (6)	70.77 (1)	33.97 (0)
(c)	BIAS	1.04×	69.07 (2)	72.81 (0)	25.29 (0)	1.10×	63.51 (6)	77.22 (2)	33.39 (0)
(ours)	Visual-Prompt Tuning	1.02×	<b>78.48</b> (6)	<b>83.00</b> (1)	<b>44.64</b> (1)	1.09×	<b>66.25</b> (6)	77.32 (2)	37.52 (0)

- Visual Prompt Tuning(VPT)를 제시한다. 이는 다양한 다운스트림 작업에 대해 큰 비전 Transformer 모델을 활용하는 새로운 파라미터 효율적인 접근법이다.
- VPT는 입력 공간에 작업별 학습 가능한 프롬프트를 도입하며, 사전 훈련된 기반을 고정한다.
- VPT는 저장 비용을 크게 줄이면서 다른 미세 조정 프로토콜(전체 미세 조정 포함)을 능가할 수 있음을 보여준다.
- 우리의 실험은 비전 Transformer의 다양한 사전 훈련 목표에 대한 미세 조정 동작과 효율적으로 더 넓은 비전 인식 작업으로 전송하는 방법에 대한 흥미로운 질문을 제기한다.
- 따라서, 우리는 우리의 작업이 비전 분야에서 큰 기반 모델의 잠재력을 최대한 활용하는 방법에 대한 미래의 연구를 자극하길 바란다.

# Reference

- <https://www.youtube.com/watch?v=bVOk-hSYyZw&t=332s>
- <https://www.youtube.com/watch?v=oqbIWH9yhiY&t=217s>

**Thank you for your Attention....!**