THE UNIVERSITY OF MELBOURNE

Neural Network Reprogrammability

Session II: Mechanics of Reprogrammability

Zesheng Ye

School of Computing and Information Systems

The University of Melbourne

Date: 21/01/2026 (AAAI 2026 Tutorial)

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Outline

I. Input Manipulation

II. Output Alignment

III. Can Input Manipulation with VLMs be better supervised?

IV. Useful Resources

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

PART 1: Input Manipulation

(a) Fundamentals

Input Manipulation Overview

TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Recall that IM practices differ in terms of …

- Manipulation formats
  - fixed or trainable
- Manipulation location
  - input space
  - embedding space
  - hidden spaces
- Manipulation operator
  - additive
  - concatenative
  - parametric

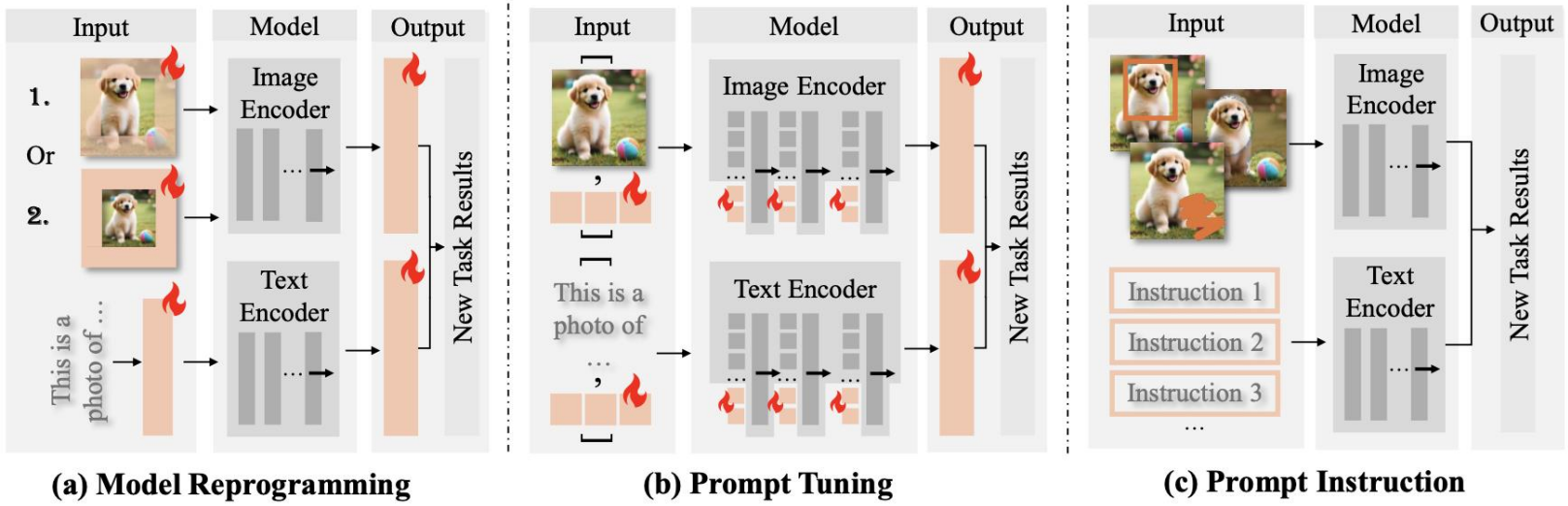


Fig. NNR manifests in different ways across different PEFT methodologies, credits to [1]

[1] Ye et al. Neural Network Reprogrammability: A Unified Theme on Model Reprogramming, Prompt Tuning, and Prompt Instruction. To appear.

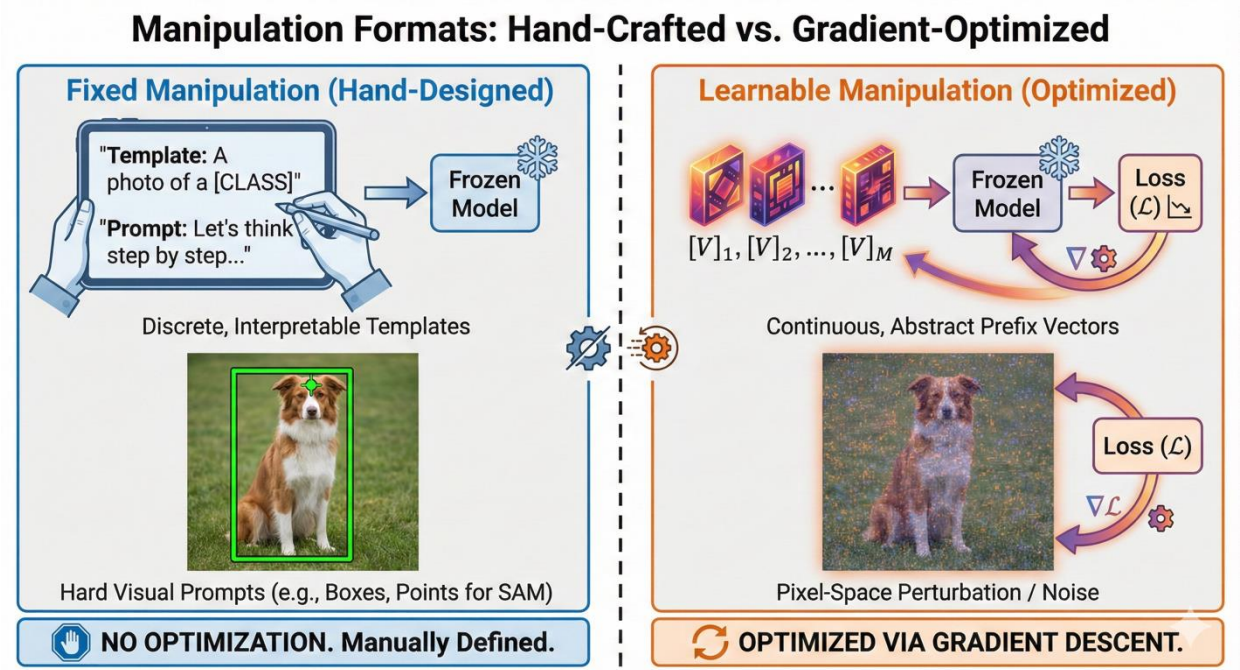
Input Manipulation Overview

TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Manipulation format: *What* is being manipulated?

- Fixed manipulation (hand-designed prompts/programs)
  - few-shot ICL, CoT prompting
  - CLIP zero-shot templates, e.g., ”a photo of a [CLASS]”
  - hard visual prompts, e.g., points/boxes/masks with SAM
- Learnable manipulation (optimized prompts/programs)
  - pixel-space perturbation
  - continuous “prefix” vectors, e.g., ”$[V]_1[V]_2 \dots [V]_M[CLASS]$”
  - node/edge/graph-level perturbation applied to graphs



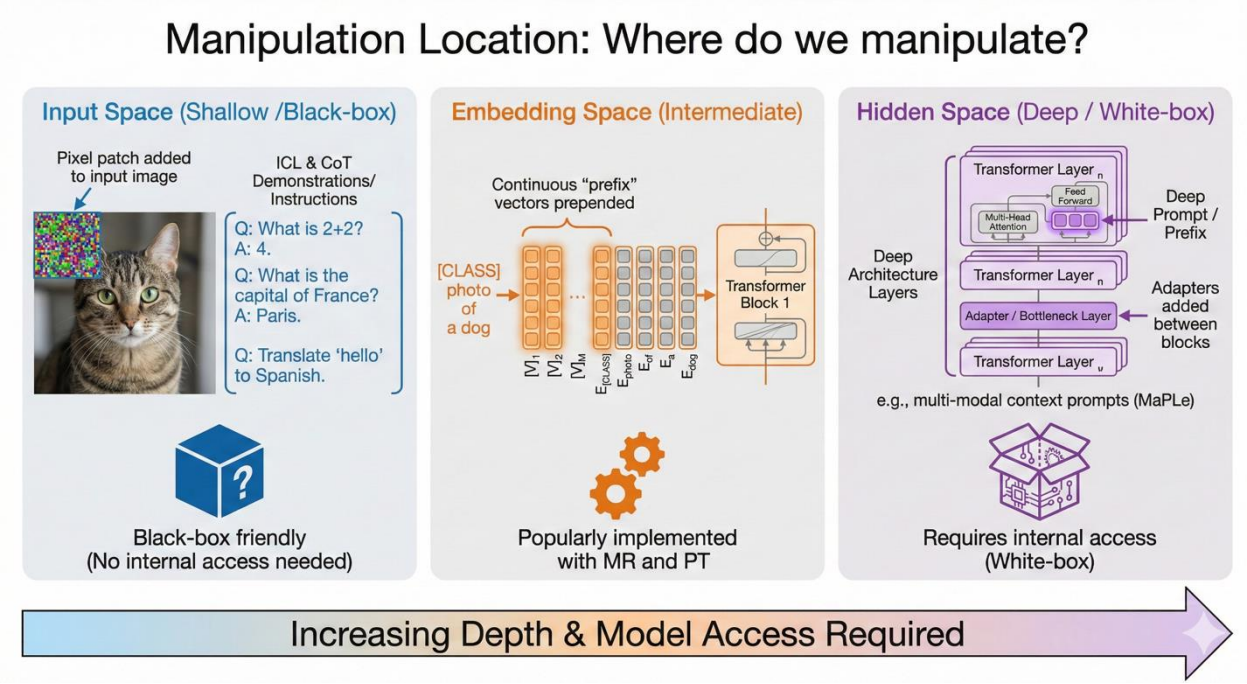
Input Manipulation Overview

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Manipulation location: *Where* do we manipulate?

- Input space $\mathcal{X}^S$
  - pixel patch added to input images
  - ICL & CoT demonstrations/instructions
  - black-box friendly
- Embedding space $\mathcal{E}$
  - prepended/postponed to token embeddings
  - continuous vectors, e.g., "$[V]_1[V]_2 \ldots [V]_M[CLASS]$"
  - most popularly implemented with MR and PT
- Hidden space $\mathcal{H}$
  - prompts/adapters inserted into deeper architectures, e.g.,
  - prompts at many Transformer layers, bottleneck layers (i.e., adapters) added between Transformer blocks, or multi-modal context prompts (i.e., MaPLe)



Input Manipulation Overview

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

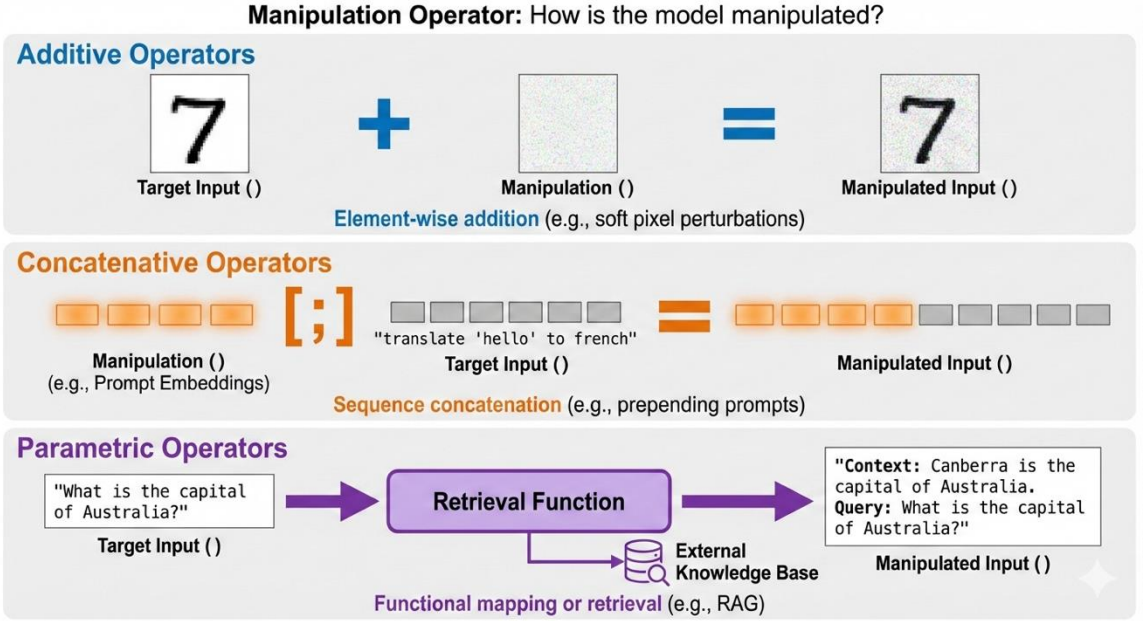
THE UNIVERSITY OF MELBOURNE

Manipulation Operator: *How* is model manipulated by λ?

- Additive operators $x^{\mathrm{S}} = x^{\mathrm{T}} + \lambda$
  - manipulation is <u>added</u> to the target input, e.g.,
  - (soft) pixel perturbations/image masks overlayed on inputs
  - (hard) points/boxes/masks add spatial cues to images

- Concatenative operators $x^{\mathrm{S}} = [x^{\mathrm{T}}; \lambda]$
  - manipulation is <u>concatenated</u> to the target input, e.g.,
  - prepend prompt embeddings before token embeddings
  - ICL/CoT demos and reasoning chains present before query

- Parametric operators $x^{\mathrm{S}} = \lambda(x^{\mathrm{T}})$
  - target input is <u>mapped</u> into pre-trained model's feature space, e.g.,
  - cross-domain reprogramming via learned input transforms
  - RAG where λ is a retrieval function that augment the context



Input Manipulation Overview

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Case studies in the NNR design space

- Example 1 – Adversarial Reprogramming
  - Format: Learnable
  - Location: Input
  - Operator: Additive
  - Output Alignment: label-mapping

→ Turn an ImageNet classifier into digit counter by learning a small pixel pattern + label mapping

- Example 2 – Soft Prompt Tuning on T5
  - Format: Learnable
  - Location: Embedding (and possibly hidden)
  - Operator: Concatenative
  - Output Alignment: identity for generation or classification head for classification

→ Learn a handful of vectors that tell a frozen T5 what task to perform

- Example 3 – GPT-3 In-Context Learning
  - Format: Fixed
  - Location: Input
  - Operator: Concatenative
  - Output Alignment: identity / rule-based parsing

→ Tell GPT-3 what task to perform by changing the prompt

Input Manipulation Overview

TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Design guidelines?

- Format
  - No gradients / API-only => Fixed prompts & instructions
  - Have labelled data / gradients? => Learnable prompts often win
- Location
  - Only black-box access? => input space
  - Need strong performance? => embedding/hidden manipulations if white-box
- Operator
  - start with concatenative
  - reach for additive when you need pixel-space or low-level control
  - parametric when you can afford extra modules

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

PART 2: Input Manipulation

(b) Representative Works in (learnable) IM

Input Manipulation

Existing IM – differ in prompting strategies!

- watermarking [1]

$$f_{\text{in}}(\boldsymbol{x}^{\text{T}}; \lambda) = \text{resize}(\boldsymbol{x}^{\text{T}}) \oplus \lambda$$

$$\text{resize} : \mathbb{R}^{H^{\text{T}} \times W^{\text{T}} \times C^{\text{T}}} \rightarrow \mathbb{R}^{H^{\text{S}} \times W^{\text{S}} \times C^{\text{S}}}$$

$$\lambda \in \mathbb{R}^{H^{\text{S}} \times W^{\text{S}} \times C^{\text{S}}}$$

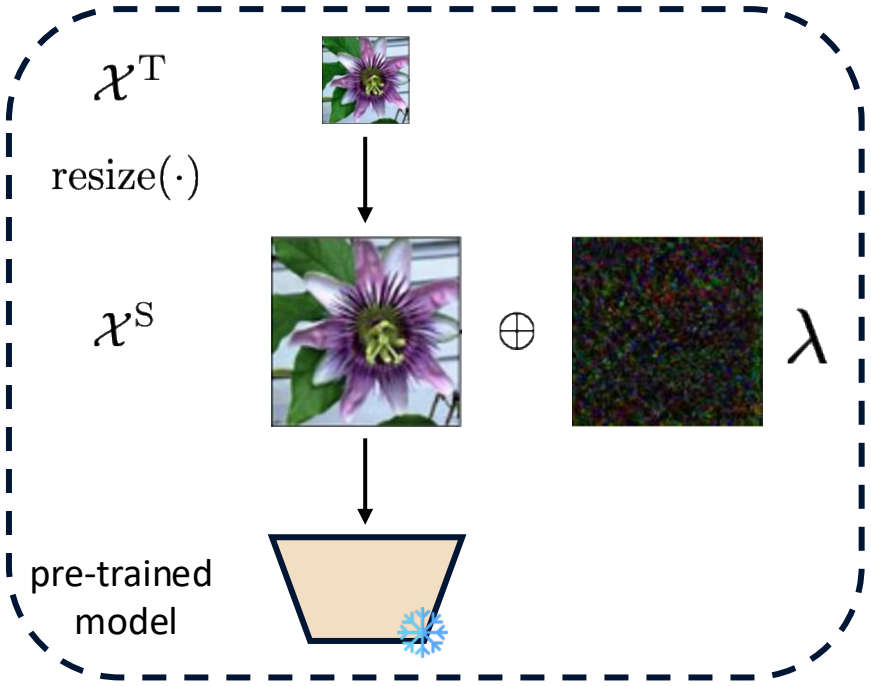


Fig. watermarking-based IM for visual prompting

[1] Bahng et al. Exploring Visual Prompts for Adapting Large-Scale Models. In ArXiV 2022

Input Manipulation

Existing IM – differ in prompting strategies!

- watermarking [1]
  - manipulation at input space
- prediction depends on the pipeline
  - unimodal / multi-modal architecture

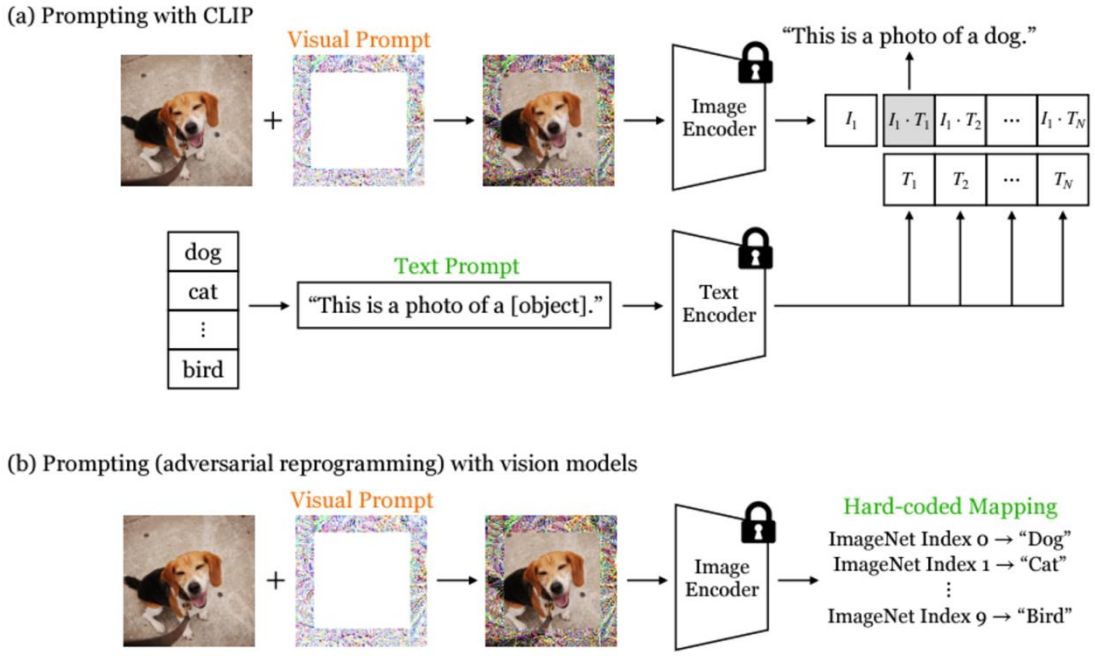


Fig. watermarking IM for VLM vs. vision model

[1] Bahng et al. Exploring Visual Prompts for Adapting Large-Scale Models. In ArXiV 2022

Input Manipulation

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Existing IM – differ in prompting strategies!

- watermarking [1]
  - manipulation at input space
- prediction depends on the pipeline
  - unimodal / multi-modal architecture
- performance depends on the target task
  - how much the target task diverges from source task matters

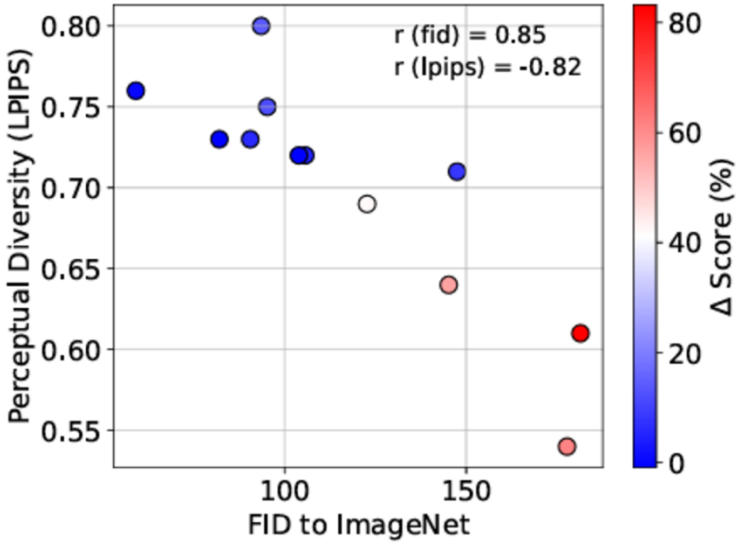


Fig. properties of downstream task that affect performance

[1] Bahng et al. Exploring Visual Prompts for Adapting Large-Scale Models. In ArXiV 2022

Input Manipulation

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Existing IM – differ in prompting strategies!

- watermarking [1]
  - manipulation at input space
- prediction depends on the pipeline
  - unimodal / multi-modal architecture
- performance also depends on the pipeline
  - multi-modal architecture can be better elevated

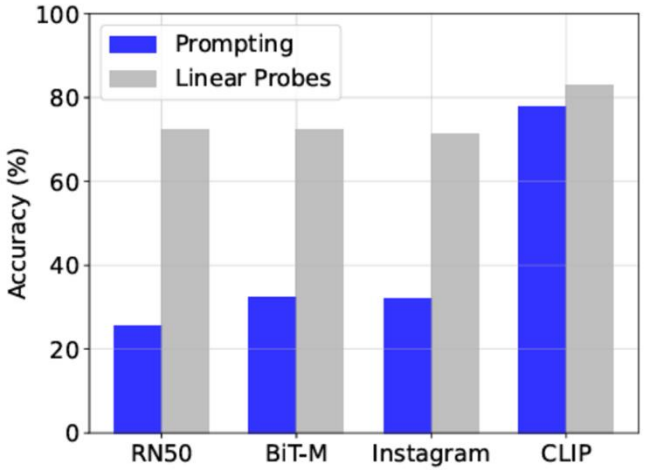


Fig. perf. comparison of prompting with VLM / vision model

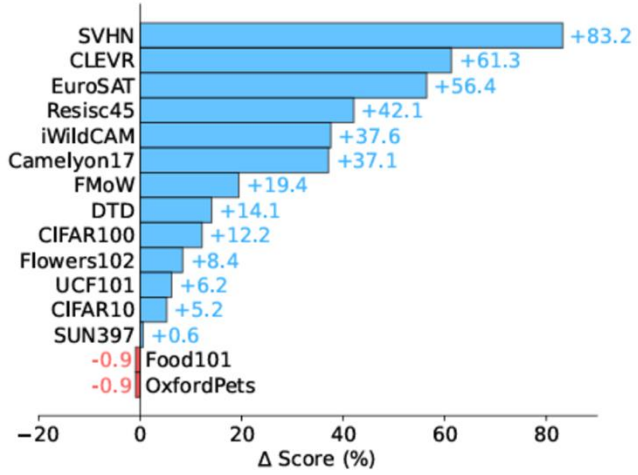


Fig. perf. comparison of prompting with zero-shot VLM for different tasks

[1] Bahng et al. Exploring Visual Prompts for Adapting Large-Scale Models. In ArXiV 2022

Input Manipulation

Existing IM – differ in prompting strategies!

- padding [1]

$$f_{\text{in}}(\boldsymbol{x}^{\text{T}};\lambda) = M \odot \text{pad}(\boldsymbol{x}^{\text{T}}) \oplus (1-M) \odot \lambda$$

$$\text{resize} : \mathbb{R}^{H^{\text{T}} \times W^{\text{T}} \times C^{\text{T}}} \rightarrow \mathbb{R}^{H^{\text{S}} \times W^{\text{S}} \times C^{\text{S}}}$$

$$\lambda \in \mathbb{R}^{(H^{\text{S}}-H^{\text{T}}) \times (W^{\text{S}}-W^{\text{T}}) \times C^{\text{S}}}$$

$$M_{i,j} = \begin{cases} 1 & \text{if } (i,j) \in \text{region of resize}(x) \\ 0 & \text{otherwise} \end{cases}$$

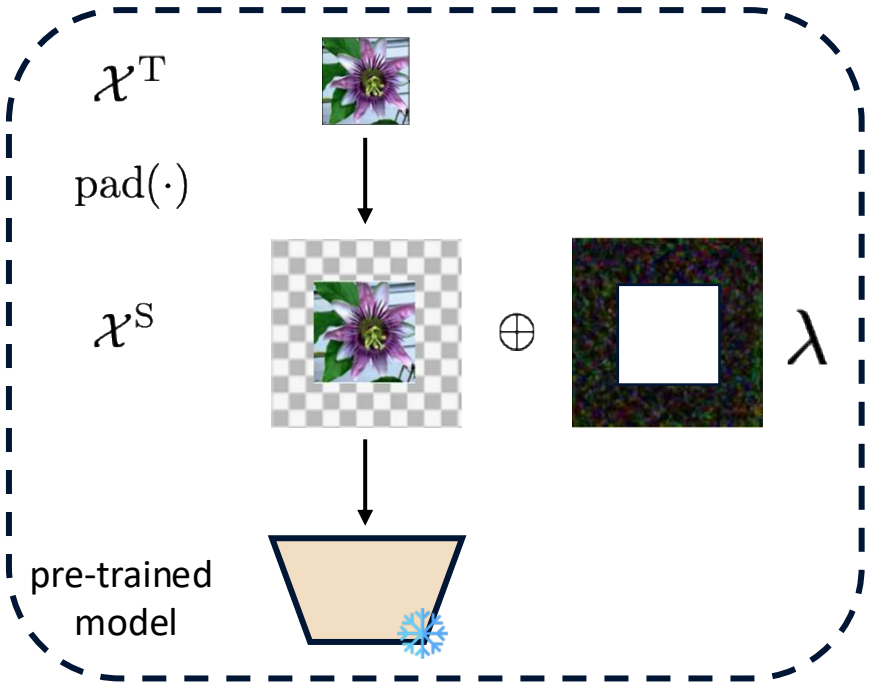


Fig. padding-based IR for visual prompting

[1] Wu et al. Unleashing the Power of Visual Prompting At the Pixel Level. In TMLR 2024

Input Manipulation

Existing IM – differ in prompting strategies!

- VPT [1]
  - manipulation at embedding (VPT-Shallow) or hidden space (VPT-Deep)
- leverage the property of pre-trained model architecture
  - patch-wise manipulation compared to pixel-level manipulation

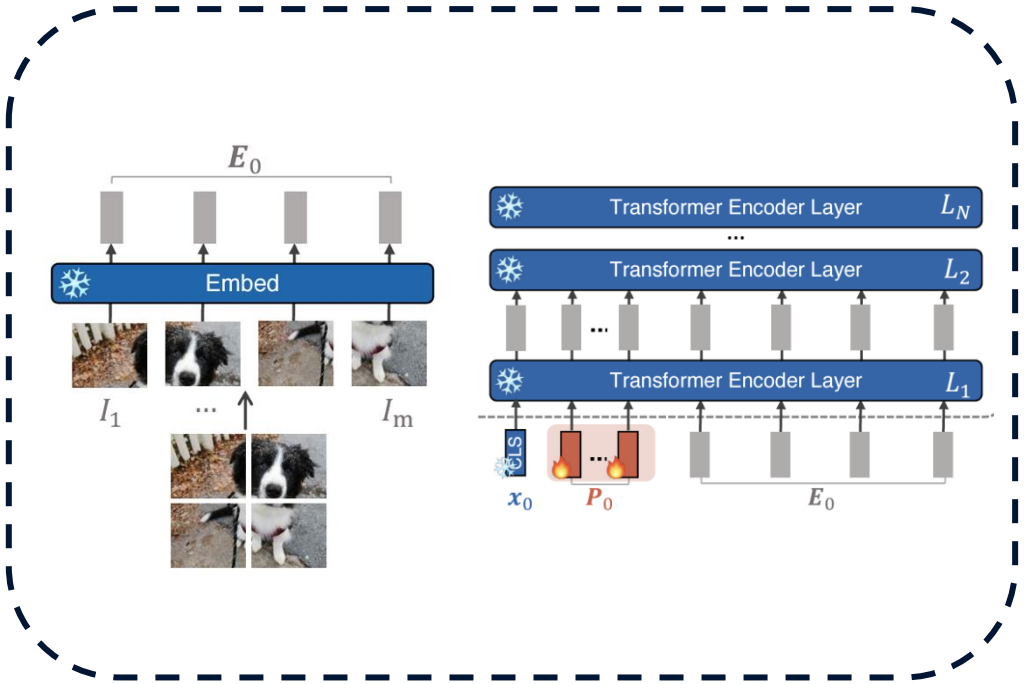


Fig. patch-wise IR for visual prompting (*with ViT only*)

[1] Jia et al. Visual Prompt Tuning. In ECCV 2022

Input Manipulation

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Existing IM – differ in prompting strategies!

- patch-wise IM for Vision Transformer (ViT) [1]

	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)

Fig. perf. comparison of patch-wise IR for visual prompting (*with ViT only*)

[1] Jia et al. Visual Prompt Tuning. In ECCV 2022

Input Manipulation

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Existing IM – differ in prompting strategies!

- patch-wise IM for Vision Transformer (ViT) [1]

	Swin-B (86.7M)	Total params	VTAB-1k Natural	VTAB-1k Specialized	VTAB-1k 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×	**79.85** (**6**)	82.45 (0)	37.75 (0)
	VPT-DEEP	1.05×	76.78 (**6**)	**84.53** (0)	**53.35** (0)

Fig. perf. comparison of patch-wise IR for visual prompting (*with ViT only*)

[1] Jia et al. Visual Prompt Tuning. In ECCV 2022

Input Manipulation

TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Recent research progress of IM

- more effective prompting **placement** [1]
- more **automated** prompting selection [2]
- more challenging scenarios
  - multi-task [3], meta-learning [4]
  - prompting with multi-modal LLM [5]

[1] Cai et al. Sample-specific Masks for Visual Reprogramming-based Prompting. In ICML 2024
[2] Tsao et al. AutoVP: An Automated Visual Prompting Framework and Benchmark. In ICLR 2024
[3] Shen et al. Multitask Vision-Language Prompt Tuning. In WACV 2024
[4] Huang et al. Diversity-Aware Meta Visual Prompting. In CVPR 2023
[5] Zhang et al. Exploring the Transferability of Visual Prompting for Multimodal Large Language Models. In CVPR 2024

Input Manipulation w/ Sample specificity

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Sample-specific masks for IM

- Existing: a pre-determined mask is **shared** across all images
  - which specifies placement of noises

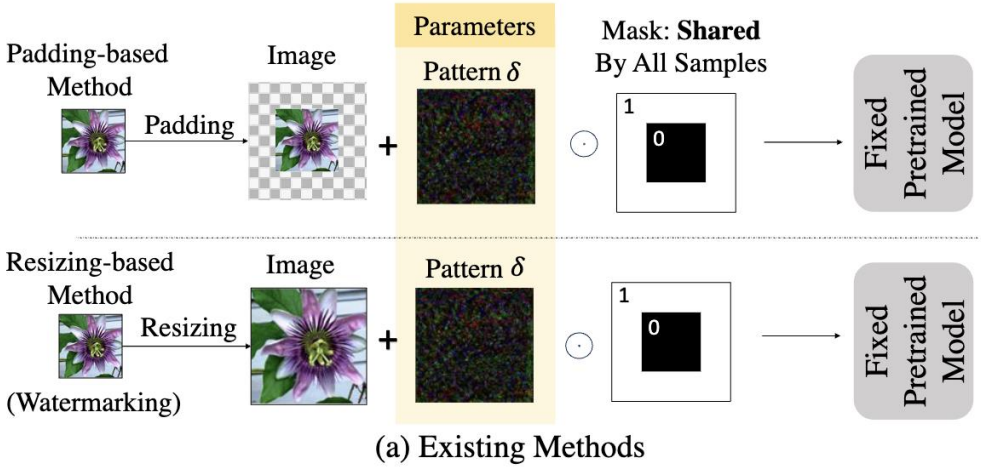


Fig. a shared mask is used across downstream samples [1]

[1] Cai et al. Sample-specific Masks for Visual Reprogramming-based Prompting. In ICML 2024

Input Manipulation w/ Sample specificity

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Sample-specific masks for IM

- a pre-determined mask is **shared** across all images
  - which specifies placement of noises
- failures observed empirically
  - on the sample level

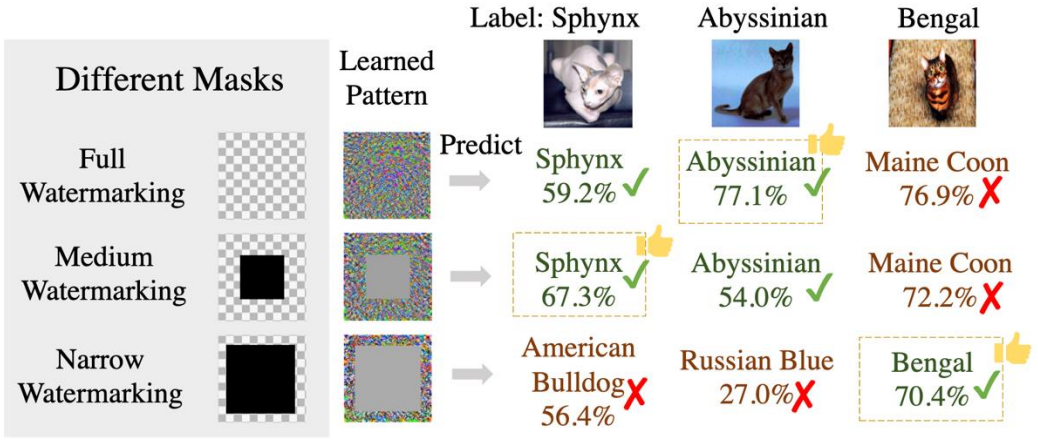


Fig. different samples benefit from different masks [1]

[1] Cai et al. Sample-specific Masks for Visual Reprogramming-based Prompting. In ICML 2024

Input Manipulation w/ Sample specificity

TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Sample-specific masks for IM

- a pre-determined mask is **shared** across all images
  - which specifies placement of noises
- failures observed empirically
  - on the dataset level

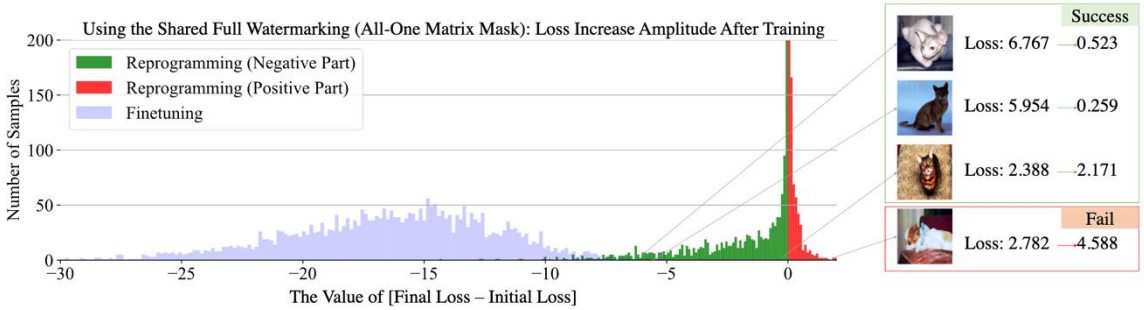


Fig. shared mask leads to loss increase for certain samples [1]

[1] Cai et al. Sample-specific Masks for Visual Reprogramming-based Prompting. In ICML 2024

Input Manipulation w/ Sample specificity

TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Sample-specific masks for IM

- a pre-determined mask is **shared** across all images
  - which specifies placement of noises
- **sub-optimality** of shared mask ☹
- SMM: sample-specific mask generation
  - CNN-based mask generator $f_{\text{mask}} : \mathbb{R}^{H_S \times W_S \times C_S} \to \mathbb{R}^{H_S \times W_S \times C_S}$
  - IM now in the form as

$$f_{\text{in}}(\boldsymbol{x}_i, \theta; \theta, \phi) = \text{resize}(\boldsymbol{x}_i) + \theta \odot f_{\text{mask}}(\text{resize}(\boldsymbol{x}_i); \phi)$$

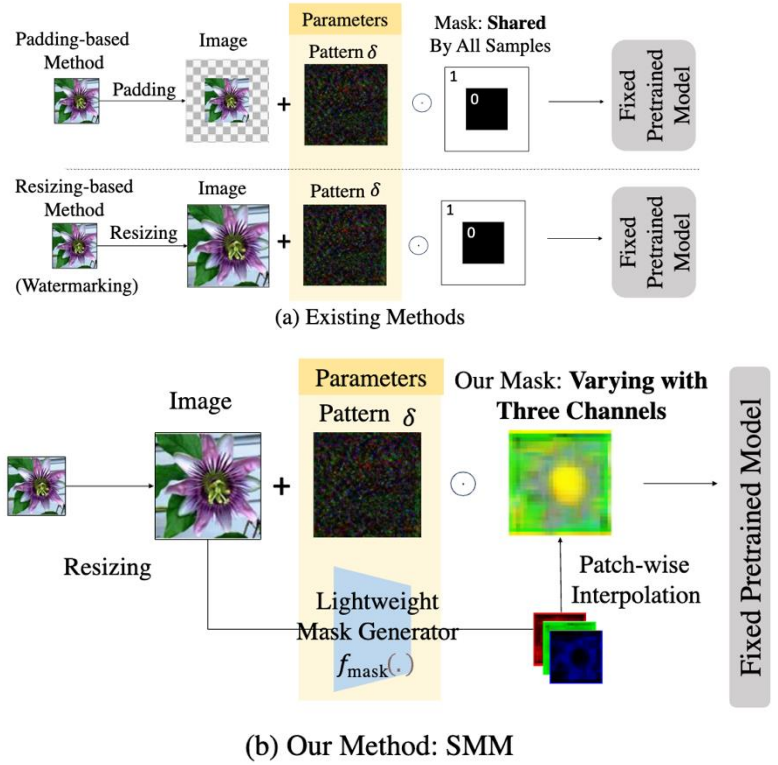


Fig. SMM: generate a mask for each image [1]

[1] Cai et al. Sample-specific Masks for Visual Reprogramming-based Prompting. In ICML 2024

Input Manipulation w/ Sample specificity

TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Sample-specific masks for IM (what's good?)

- **performance gain** across pre-trained classifier

Pre-trained	ResNet-18 (ImageNet-1k)					ResNet-50 (ImageNet-1k)				
Methods	Pad	Narrow	Medium	Full	Ours	Pad	Narrow	Medium	Full	Ours
CIFAR10	65.5 ±0.1	68.6 ±2.8	68.8 ±1.1	68.9 ±0.4	**72.8** ±0.7	76.6±0.3	77.4±0.5	77.8±0.2	79.3±0.3	**81.4**±0.6
CIFAR100	24.8±0.1	36.9±0.6	34.9±0.2	33.8±0.2	**39.4**±0.6	38.9±0.3	42.5±0.2	43.8±0.2	47.2±0.1	**49.0**±0.2
SVHN	75.2±0.2	58.5±1.1	71.1±1.0	78.3±0.3	**84.4**±2.0	75.8±0.4	59.1±1.3	71.5±0.8	79.5±0.5	**82.6**±2.0
GTSRB	52.0±1.2	46.1±1.5	56.4±1.0	76.8±0.9	**80.4**±1.2	52.5±1.4	38.9±1.3	52.6±1.3	76.5±1.3	**78.2**±1.1
Flowers102	27.9±0.7	22.1±0.1	22.6±0.5	23.2±0.5	**38.7**±0.7	24.6±0.6	19.9±0.6	20.9±0.6	22.6±0.1	**35.9**±0.5
DTD	**35.3**±0.9	33.1±1.3	31.7±0.5	29.0±0.7	33.6±0.4	40.5±0.5	37.8±0.7	38.4±0.2	34.7±1.3	**41.1**±1.1
UCF101	23.9±0.5	27.2±0.9	26.1±0.3	24.4±0.9	**28.7**±0.8	34.6±0.2	38.4±0.2	37.2±0.2	35.2±0.2	**38.9**±0.5
Food101	14.8±0.2	14.0±0.1	14.4±0.3	13.2±0.1	**17.5**±0.1	17.0±0.3	18.3±0.2	18.3±0.2	16.7±0.2	**19.8**±0.0
SUN397	13.0±0.2	15.3±0.1	14.2±0.1	13.4±0.2	**16.0**±0.3	20.3±0.2	22.0±0.1	21.5±0.1	21.1±0.1	**22.9**±0.0
EuroSAT	85.2±0.6	82.8±0.4	83.8±0.5	84.3±0.5	**92.2**±0.2	83.6±0.7	83.7±0.4	85.8±0.1	86.9±0.3	**92.0**±0.6
OxfordPets	65.4±0.7	73.7±0.2	71.4±0.2	70.0±0.6	**74.1**±0.4	76.2±0.6	76.4±0.3	75.6±0.3	73.4±0.3	**78.1**±0.2
Average	43.91	43.48	45.04	46.85	**52.53**	49.15	46.76	49.39	52.10	**56.35**

Fig. SMM performance with ResNet-18 and ResNet-50

Pre-trained	ViT-B32 (ImageNet-1k)				
Method	Pad	Narrow	Medium	Full	Ours
CIFAR10	62.4	96.6	96.5	95.8	**97.4**
CIFAR100	31.6	74.4	75.3	75.0	**82.6**
SVHN	80.2	85.0	87.4	87.8	**89.7**
GTSRB	62.3	57.8	68.6	75.5	**80.5**
Flowers102	57.3	55.3	56.6	55.9	**79.1**
DTD	43.7	37.3	38.5	37.7	**45.6**
UCF101	33.6	44.5	**44.8**	40.9	42.6
Food101	37.4	47.3	48.6	49.4	**64.8**
SUN397	21.8	29.0	29.4	28.8	**36.7**
EuroSAT	**95.9**	90.9	90.9	89.1	93.5
OxfordPets	57.6	82.5	81.0	75.3	**83.8**
Average	53.1	63.7	65.2	64.7	**72.4**

Fig. SMM performance with ViT-B32

[1] Cai et al. Sample-specific Masks for Visual Reprogramming-based Prompting. In ICML 2024

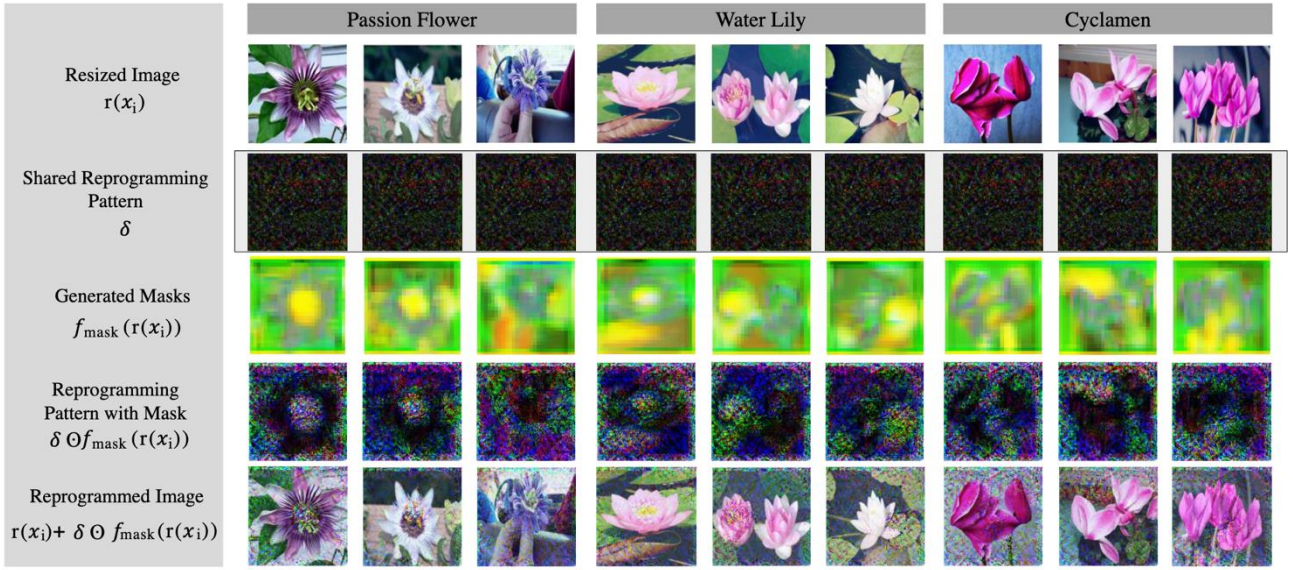
Input Manipulation w/ Sample specificity

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Sample-specific masks for IM (what's good?)

- masks ***tailored to*** each specific image, better aligning with the semantics



[1] Cai et al. Sample-specific Masks for Visual Reprogramming-based Prompting. In ICML 2024

Input Manipulation w/ Sample specificity

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Sample-specific masks for IM (what's good?)

- **negligible additional parameters** compared with pre-trained classifier

PRE-TRAINED	INPUT IMAGE SIZE	f_{mask} CNN LAYERS	EXTRA PARAMETERS OF OUR f_{mask}	OUR EXTRA PARAMETERS ÷ REPROGRAMMING PARAMETERS	OUR EXTRA PARAMETERS ÷ PRE-TRAINED MODEL PARAMETERS
RESNET-18	224×224×2	5	26,499	17.60%	0.23%
RESNET-50	224×224×3	5	26,499	17.60%	0.10%
VIT-B32	384×384×3	6	102,339	23.13%	0.12%

- **competitive performance:** SMM [1] outperforms LoRA [2] in the case of *fewer* trainable parameters
  - LoRA: # params increases when pre-trained classifier grows in scale

	EXTRA PARAMETERS	CIFAR10	CIFAR100	SVHN	GTSRB	AVERAGE (32×32)	AVERAGE (128×128)
FINETUNING-LORA	0.60M	95.9	83.6	65.3	66.6	77.9	83.4
OUR SMM	0.54M	**97.4**	**87.3**	**91.0**	**84.2**	**90.0**	**83.5**

[1] Cai et al. Sample-specific Masks for Visual Reprogramming-based Prompting. In ICML 2024
[2] Hu et al. Lora: Low-rank adaptation of large language models. In ICLR 2022

Input Manipulation w/ Sample specificity

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Sample-specific masks for IM (what's good?)

- **negligible additional parameters** compared with pre-trained classifier

PRE-TRAINED	INPUT IMAGE SIZE	f_{mask} CNN LAYERS	EXTRA PARAMETERS OF OUR f_{mask}	OUR EXTRA PARAMETERS ÷ REPROGRAMMING PARAMETERS	OUR EXTRA PARAMETERS ÷ PRE-TRAINED MODEL PARAMETERS
RESNET-18	224×224×2	5	26,499	17.60%	0.23%
RESNET-50	224×224×3	5	26,499	17.60%	0.10%
VIT-B32	384×384×3	6	102,339	23.13%	0.12%

- **orthogonal role**: SMM is *not a competitor* of fine-tuning, but can further boost its performance when combined

	CIFAR10	CIFAR100	SVHN	GTSRB	FLOWERS102	DTD
FINETUNING-FC	90.1	70.7	63.5	77.8	**90.9**	67.6
FINETUNING-FC + OUR SMM	**91.2**	**72.4**	**86.9**	**85.2**	**90.9**	**68.2**

	UCF101	FOOD101	SUN397	EUROSAT	OXFORDPETS	AVERAGE
FINETUNING-FC	70.8	57.6	53.5	95.7	90.4	75.3
FINETUNING-FC + OUR SMM	**72.0**	**59.6**	**57.9**	**95.8**	**90.6**	**79.2**

[1] Cai et al. Sample-specific Masks for Visual Reprogramming-based Prompting. In ICML 2024

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

PART 2: Input Manipulation

(c) Conclusions

Conclusion

TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Wrap up

- IM takes inspiration from adversarial attack, but in a beneficial side
- introduce fixed patterns or trainable parameters as task-specific hint
- can take various ways, places, modalities to be applied

Conclusion

TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Future Outlook

- Can IM patterns be transferrable across downstream tasks?
- Can IM pattern be learnable while being in more interpretable forms?
- Can IM pattern be designed with formal theoretical guarantee?

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

PART 2: Output Alignment

(a) Fundamentals

Recap NNR Framework

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Adapt pre-trained model w/ NNR

compared with fully fine-tuning

- Why: **freeze** pre-trained model's parameter space
  - preserve encoded knowledge
  - keep efficient when model scales

- How: **modify** input/context and output spaces
  1) input manipulation
  2) output alignment

Fig. what we need to adapt in NNR – input and output

[1] Elsayed et al. Adversarial Reprogramming of Neural Networks. In ICLR 2019

Output Alignment (OA) across methodology

NNR needs not only IM, but also OA

- Manipulation formats
- Manipulation location
- Manipulation operator
- Output Alignment
  - identity mapping
  - rule-based
  - learning-based
  - statistics-based

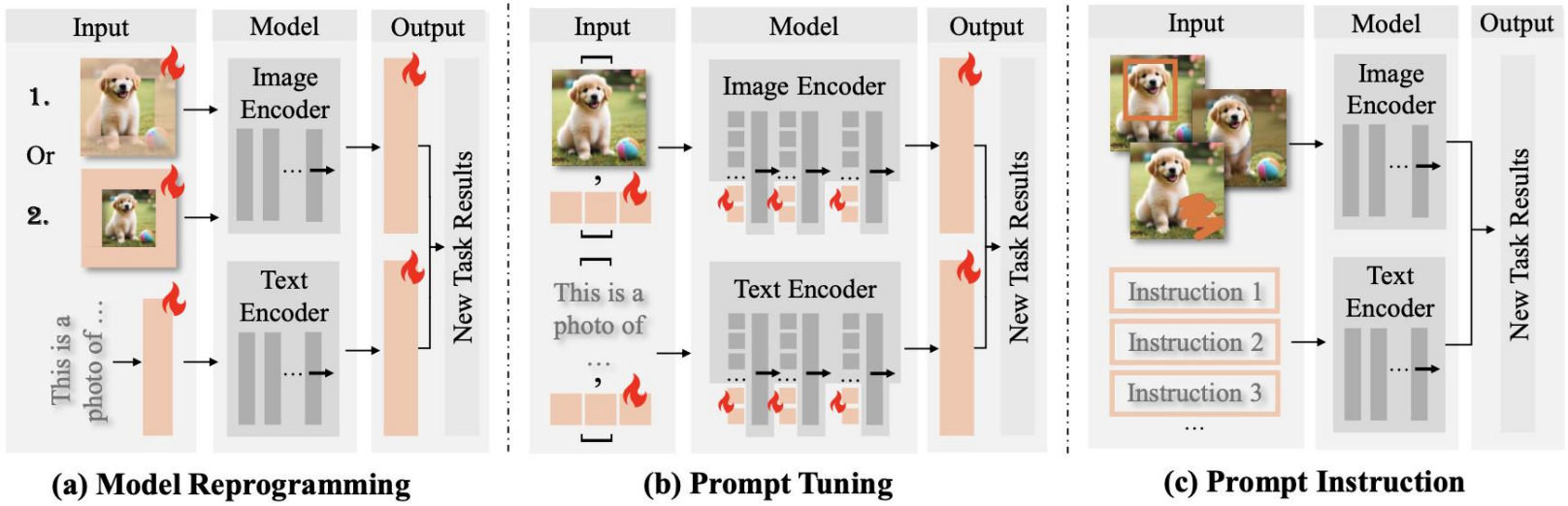


Fig. OA manifests in different ways across different PEFT methodologies, credits to [1]

[1] Ye et al. Neural Network Reprogrammability: A Unified Theme on Model Reprogramming, Prompt Tuning, and Prompt Instruction. To appear.

Output Alignment (OA) - implicit

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

OA depends on the task

- identity mapping
  - when $Y^S = Y^T$
- rule-based
- learning-based
- statistics-based

} implicit

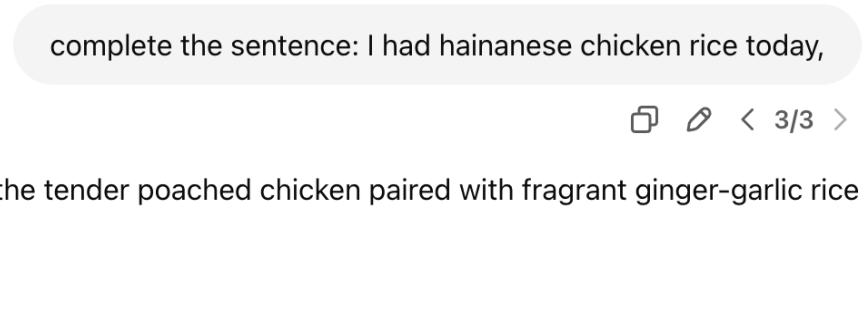


Fig. OA in the form of identity mapping

Output Alignment (OA) - implicit

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

OA depends on the task

- identity mapping
  - when $Y^S = Y^T$
- rule-based alignment
  - when structured interpretation is needed

} implicit

- learning-based
- statistics-based

very briefly summarize what is fine-tuning. if mathematical concepts are involved, they should be rendered using LaTeX, in bullet list.

Fine-tuning is the process of taking a pre-trained model and further training it on a smaller, task-specific dataset to adapt it to a particular application.

- A pre-trained model f_θ with parameters θ is updated using new data.
- The objective is to minimize a loss function $\mathcal{L}$ over the new dataset $D = \{(x_i, y_i)\}$:

$$\theta^* = \arg\min_\theta \sum_{(x_i,y_i)\in D} \mathcal{L}(f_\theta(x_i), y_i)$$

- Typically uses a lower learning rate to retain useful knowledge from the original training.

Fig. OA in the form of rule-based mapping

Output Alignment (OA) - explicit

When OA cannot be implemented internally, image classification as an example

- frozen pre-trained classifier $f : \mathcal{X}^{\mathrm{S}} \to \mathcal{Y}^{\mathrm{S}}$
  - e.g., ResNet, ResNeXt, ViT
- Input Manipulation (IM) $I : \mathcal{X}^{\mathrm{T}} \to \mathcal{X}^{\mathrm{S}}$
  - e.g., learnable noise patterns
- Output Alignment (OA) $O : \mathcal{Y}^{\mathrm{S}} \to \mathcal{Y}^{\mathrm{T}}$
  - equivalently label mapping
  - establish meaningful mappings between label spaces

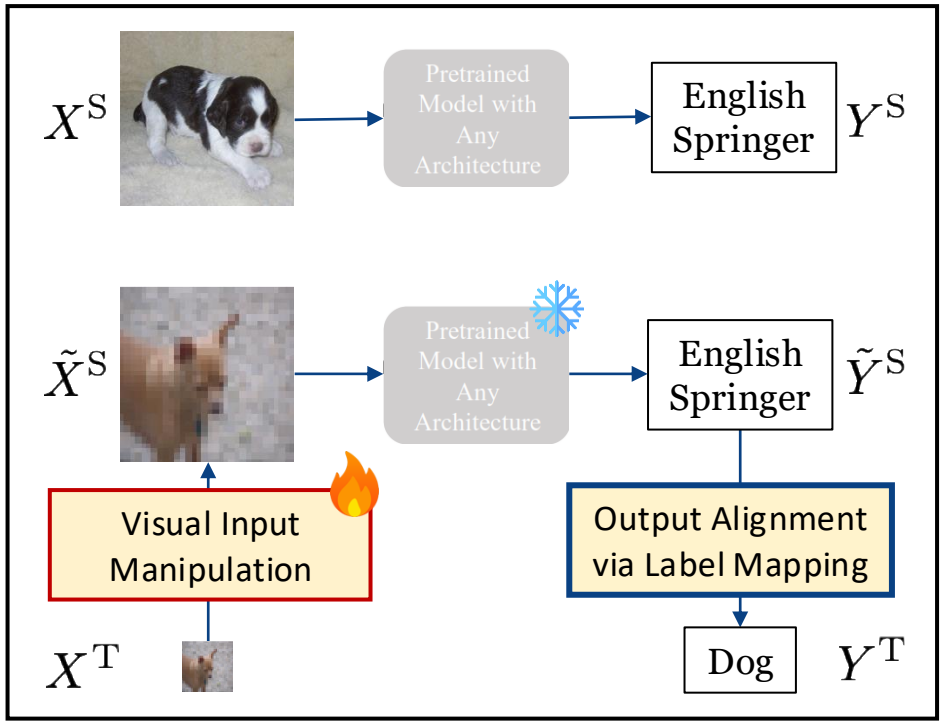


Fig. Example VR: ImageNet 1K -> CIFAR10

Output Alignment (OA) - explicit

Together with IM, forms a forward pass for each x^T

- frozen pre-trained classifier $f : \mathcal{X}^{\mathrm{S}} \to \mathcal{Y}^{\mathrm{S}}$
- Input Manipulation (IM) $I : \mathcal{X}^{\mathrm{T}} \to \mathcal{X}^{\mathrm{S}}$
- Output Alignment (OA) $O : \mathcal{Y}^{\mathrm{S}} \to \mathcal{Y}^{\mathrm{T}}$

Supervised Training

$\mathcal{D}^{\mathrm{T}} = \{(x_i^{\mathrm{T}}, y_i^{\mathrm{T}})\}_{i=1}^{n}$

$$\min_{\Theta} \frac{1}{n} \sum_{i=1}^{n} \ell\left(y_i^{\mathrm{T}}, (O \circ f \circ I)\right)(x_i^{\mathrm{T}}; \Theta)$$

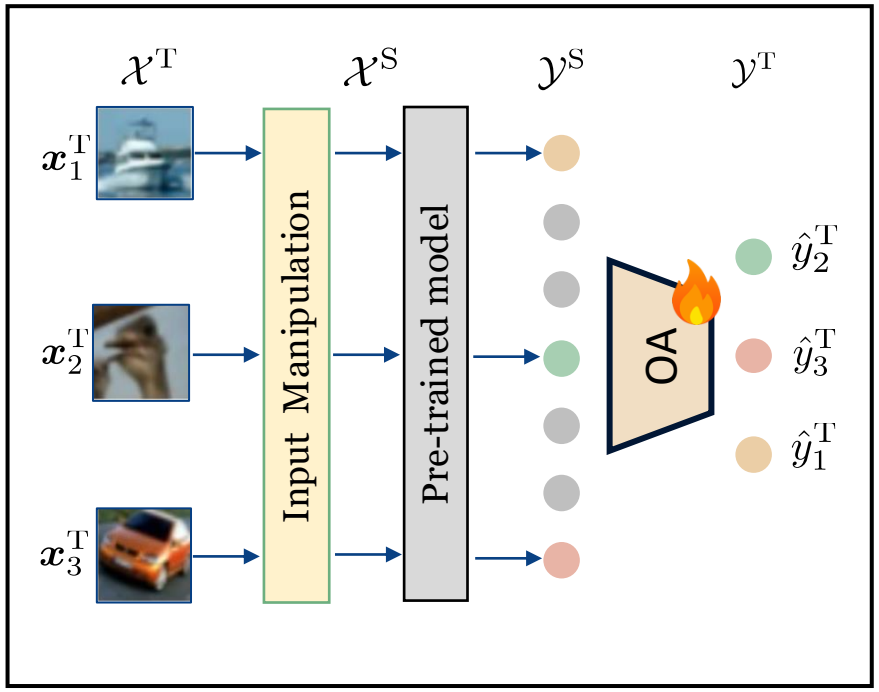


Fig. OA in the form of explicit mapping

Output Alignment (OA) - explicit

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Explicit OA by label mapping

- Learning-based
  - optimize trainable and **parametric** fully-connected layer ω.
  - increased complexity as $|\mathcal{Y}^{\mathrm{S}}| \times |\mathcal{Y}^{\mathrm{T}}|$ grows
  - potential over-fitting risks and lower efficiency

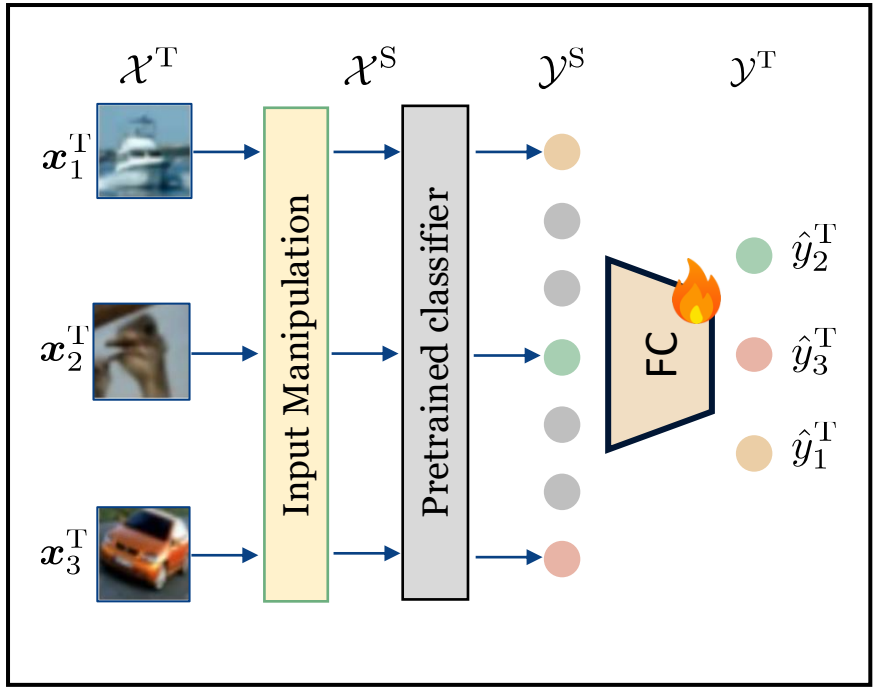


Fig. learning-based Output Mapping w/ trainable ω

Output Alignment (OA) - explicit

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Explicit OA by label mapping

- Learning-based
  - optimize a learnable and **parametric** fully-connected layer ω.
  - increased complexity as $|\mathcal{Y}^{\mathrm{S}}| \times |\mathcal{Y}^{\mathrm{T}}|$ grows
  - potential over-fitting risks and lower efficiency

- Statistics-based
  - 1-to-1 mapping for $y^{\mathrm{S}} \in \mathcal{Y}^{\mathrm{S}} \rightarrow y^{\mathrm{T}} \in \mathcal{Y}^{\mathrm{T}}, \ \ \forall y^{\mathrm{T}}$
  - based on the statistical inference results

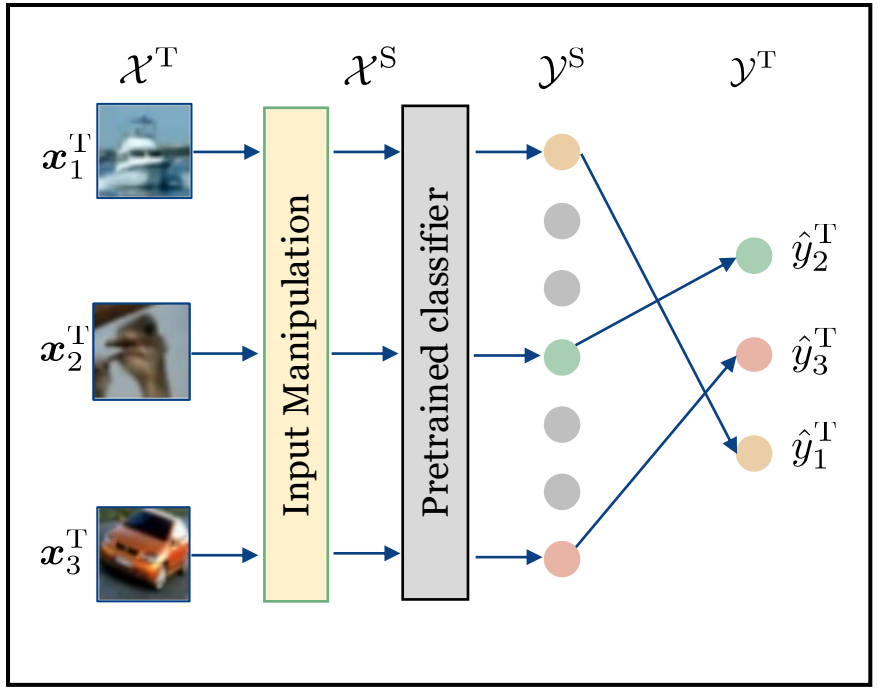


Fig. 1-to-1 statistical inference-based output mapping w/o trainable ω

Output Alignment (OA) - explicit

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Explicit OA by label mapping

- Inference-based
  - 1-to-1 mapping for $y^{\mathrm{S}} \in \mathcal{Y}^{\mathrm{S}} \rightarrow y^{\mathrm{T}} \in \mathcal{Y}^{\mathrm{T}}, \ \ \forall y^{\mathrm{T}}$
  - based on the inference results
- Established strategies
  - random label mapping (RLM) [1]
  - frequent label mapping (FLM) [2]
  - iterative label mapping (ILM) [3]

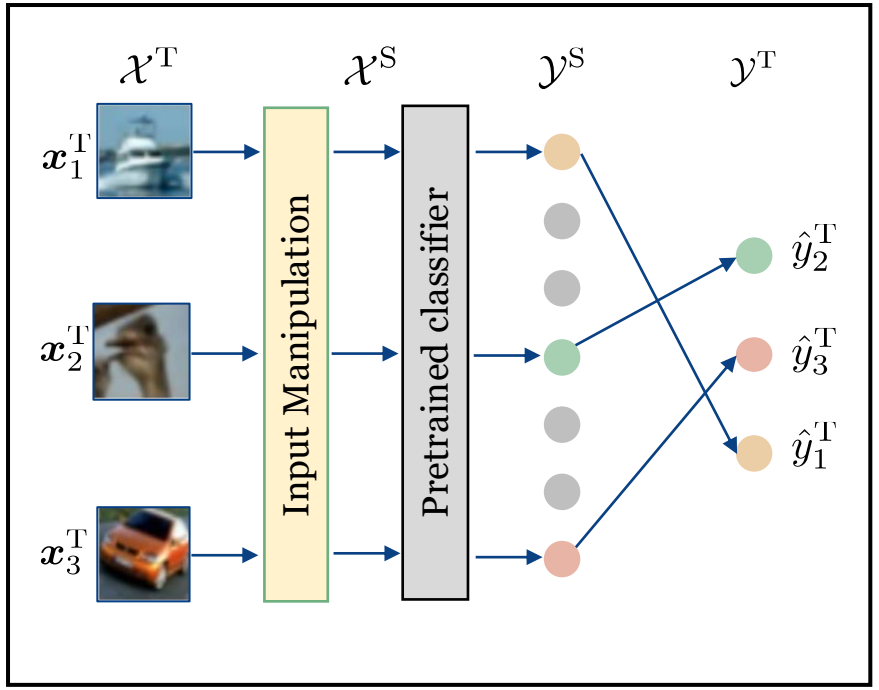


Fig. 1-to-1 Output Mapping

[1] Elsayed et al. Adversarial Reprogramming of Neural Networks. In ICLR 2019
[2] Tsai et al. Transfer Learning without knowing: Reprogramming black-box machine learning models with scarce data and limited resources. In ICML 2021
[3] Chen et al. Understanding and improving visual prompting: A label-mapping perspective. In CVPR 2023

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

PART 2: Output Alignment

(b) Representative Works

Revisit Explicit OA

(Explicit) OA via statistical inference

- Random label mapping [1]
  - 1-to-1 mapping for $y^{\mathrm{S}} \in \mathcal{Y}^{\mathrm{S}} \rightarrow y^{\mathrm{T}} \in \mathcal{Y}^{\mathrm{T}}, \ \ \forall y^{\mathrm{T}}$
  - based on the inference results
- How to establish the mapping?
  - *random* hard-coded mapping from pre-trained label set, e.g.,
  - first 10 classes from $\mathcal{Y}^{\mathrm{S}}$, if $|\mathcal{Y}^{\mathrm{T}}| = 10$

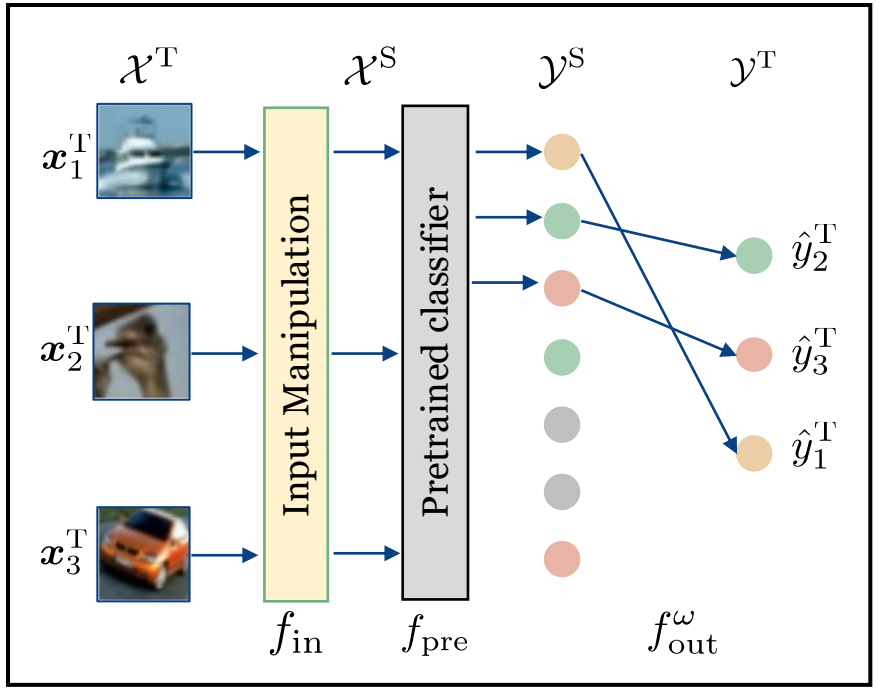


Fig. 1-to-1 RLM w/o trainable ω

[1] Elsayed et al. Adversarial Reprogramming of Neural Networks. In ICLR 2019

Revisit Explicit OA

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

(Explicit) OA via statistical inference

- Frequent label mapping [1]
  - 1-to-1 mapping for $y^{\mathrm{S}} \in \mathcal{Y}^{\mathrm{S}} \rightarrow y^{\mathrm{T}} \in \mathcal{Y}^{\mathrm{T}}, \ \ \forall y^{\mathrm{T}}$
  - based on the inference results

- How to establish the mapping?
  - sequentially assign *the most frequent* source label to
  - corresponding dominating target label, until
  - each target label is assigned

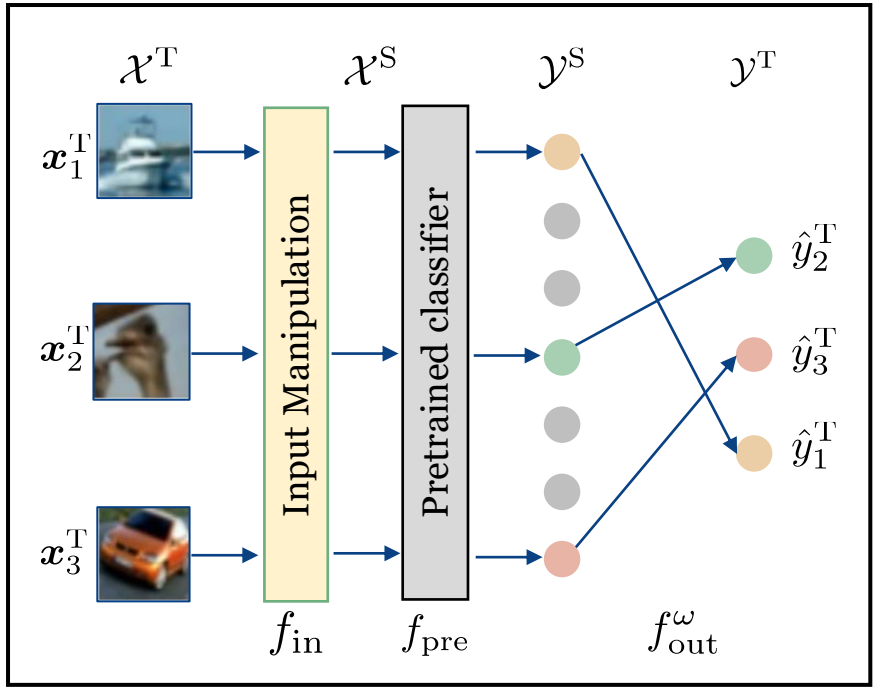


Fig. 1-to-1 FLM w/o trainable ω

[1] Tsai et al. Transfer Learning without knowing: Reprogramming black-box machine learning models with scarce data and limited resources. In ICML 2021

Output Mapping

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

(Explicit) OA via statistical inference

- Even RLM yields meaningful feature spaces (before classification layer)

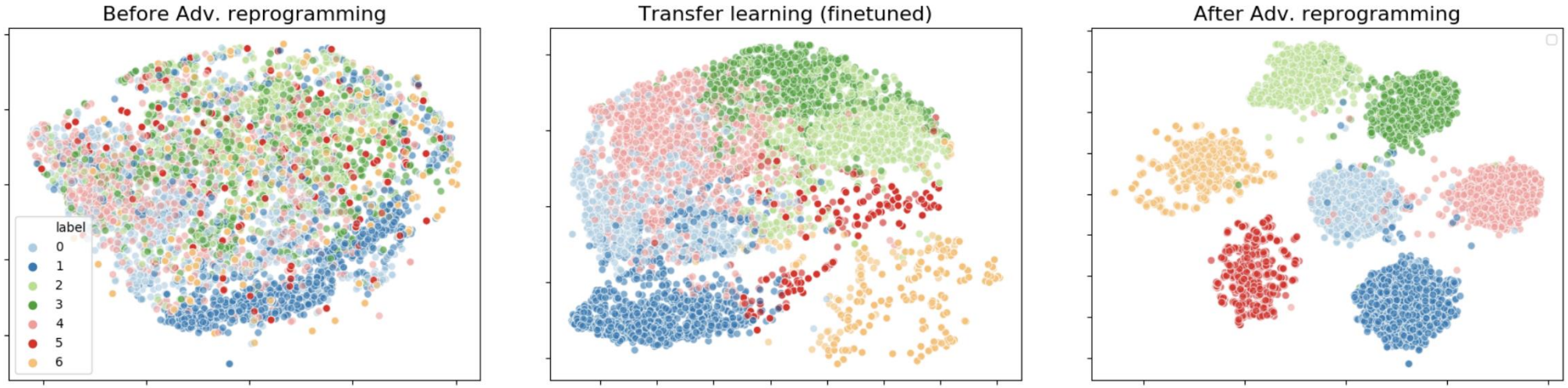


Fig. 1-to-1 t-SNE embedding using ResNet50

[1] Tsai et al. Transfer Learning without knowing: Reprogramming black-box machine learning models with scarce data and limited resources. In ICML 2021

Revisit Explicit OA

(Explicit) OA via statistical inference

- Moreover, FLM does lead to meaningful performance gain over RLM

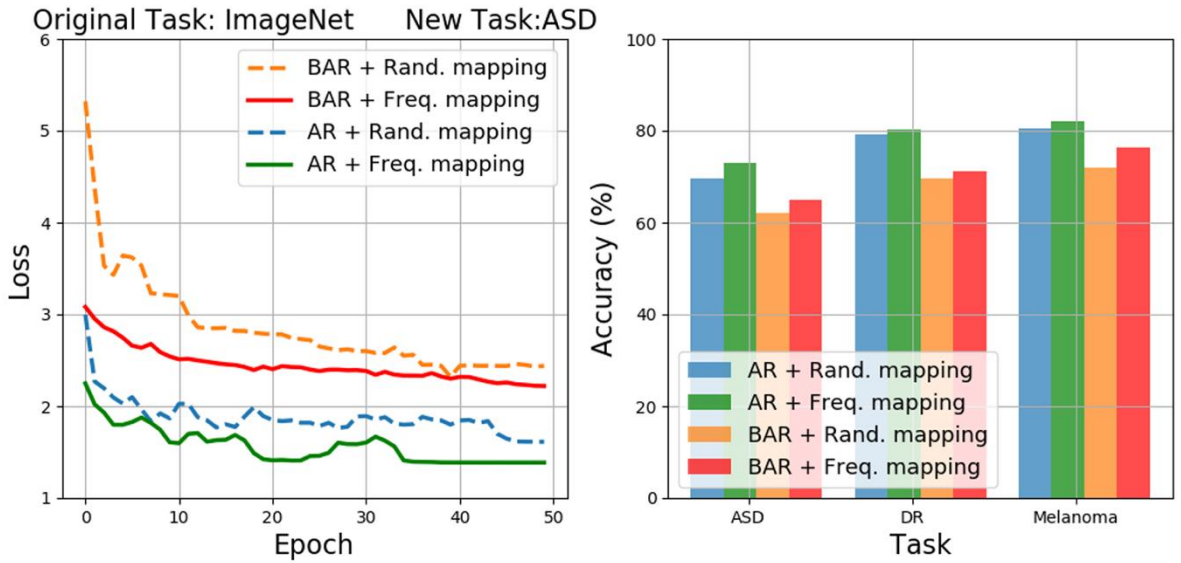


Fig. 1-to-1 Ablation study on RLM and FLM

[1] Tsai et al. Transfer Learning without knowing: Reprogramming black-box machine learning models with scarce data and limited resources. In ICML 2021

Revisit Explicit OA

(Explicit) OA via statistical inference

- Iterative label mapping [1]
  - 1-to-1 mapping for $y^{\mathrm{S}} \in \mathcal{Y}^{\mathrm{S}} \rightarrow y^{\mathrm{T}} \in \mathcal{Y}^{\mathrm{T}}, \ \ \forall y^{\mathrm{T}}$
  - based on the inference results
- How to establish the mapping?
  - follow the *principle of FLM*
  - refine the correspondence iteratively during optimization

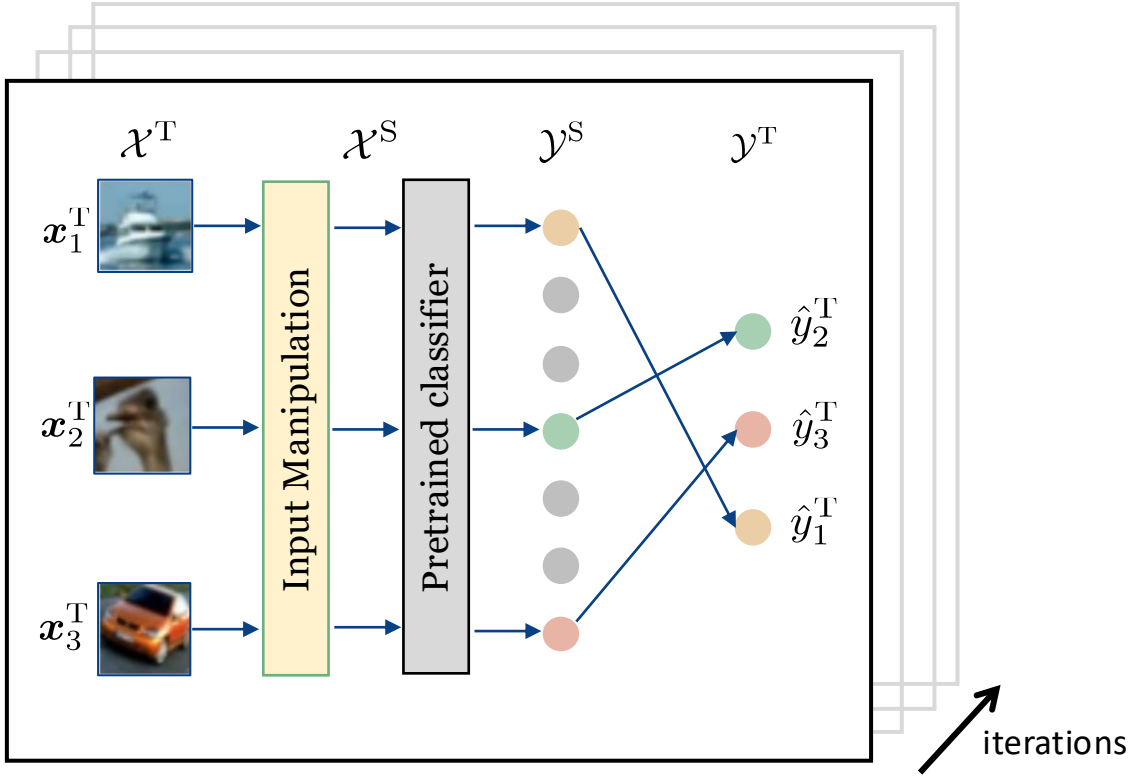


Fig. 1-to-1 ILM w/o trainable ω

[1] Chen et al. Understanding and improving visual prompting: A label-mapping perspective. In CVPR 2023

Summarizing Explicit OA

(Explicit) OA via statistical inference

- Existing statistical-based OA
  - **1-to-1** mapping for $y^{\mathrm{S}} \in \mathcal{Y}^{\mathrm{S}} \rightarrow y^{\mathrm{T}} \in \mathcal{Y}^{\mathrm{T}}, \ \forall y^{\mathrm{T}}$
  - based on the inference results

- Established statistical-based OA strategies
  - random label mapping (RLM)
  - frequent label mapping (FLM)
  - iterative label mapping (ILM)

	I^{θ}	f_{pre}	$f_{\mathrm{out}}^{\omega}$
Learning	Updated by BP	Frozen	Updated by BP
Statistical	Updated by BP	Frozen	Determined by FP

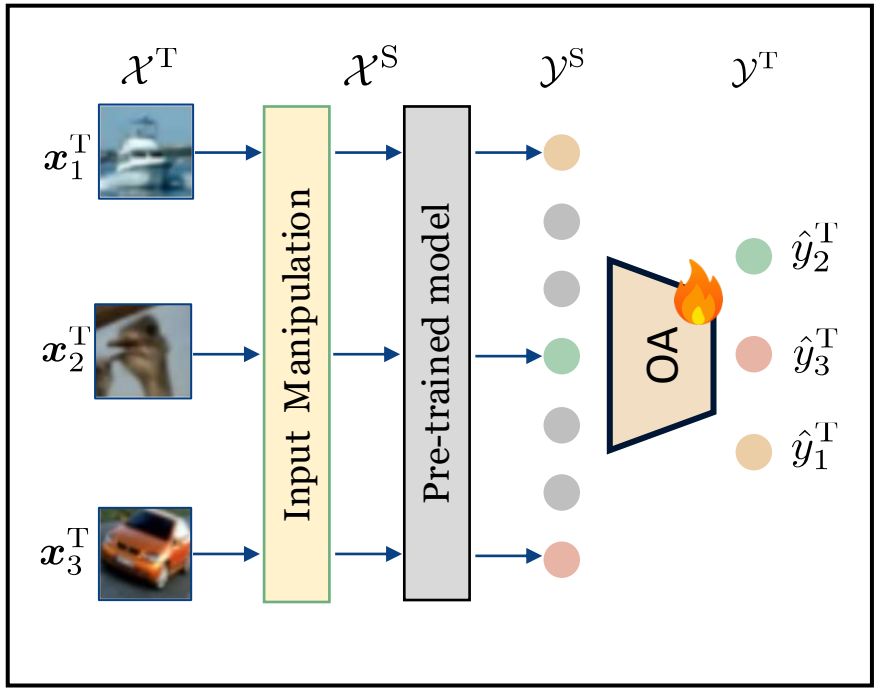


Fig. OA in the form of explicit mapping

Bayesian-guided (statistical) Label Mapping

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Are 1-to-1 Mappings all we need?

- Anything wrong? (sample level)
  - Each downstream label takes only <u>one</u> pre-trained label with the <u>highest</u> logits

Ignoring other **high-ranking** logits (and *semantics similarities*)

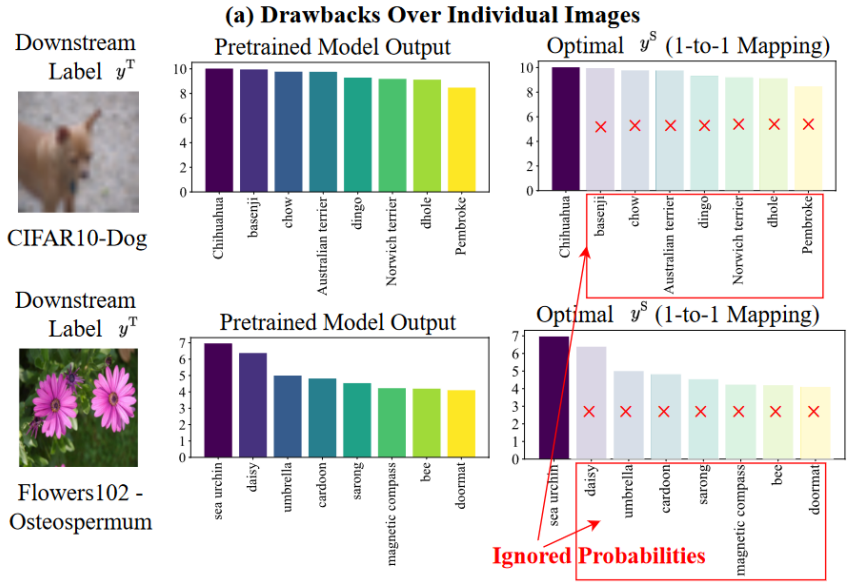


Fig. limitations of 1-to-1 mapping

[1] Cai et al. Bayesian-guided Label Mapping for Visual Reprogramming. In NeurIPS 2024

Bayesian-guided (statistical) Label Mapping

TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Are 1-to-1 Mappings all we need?

- Anything wrong? (dataset level)
  - greedy `top-1` matching deprives optimal matching
    - [moving van] has already been paired with [truck]
    - [moving van] can no longer be paired with [automobile]

sub-optimal *label assignment* for certain classes

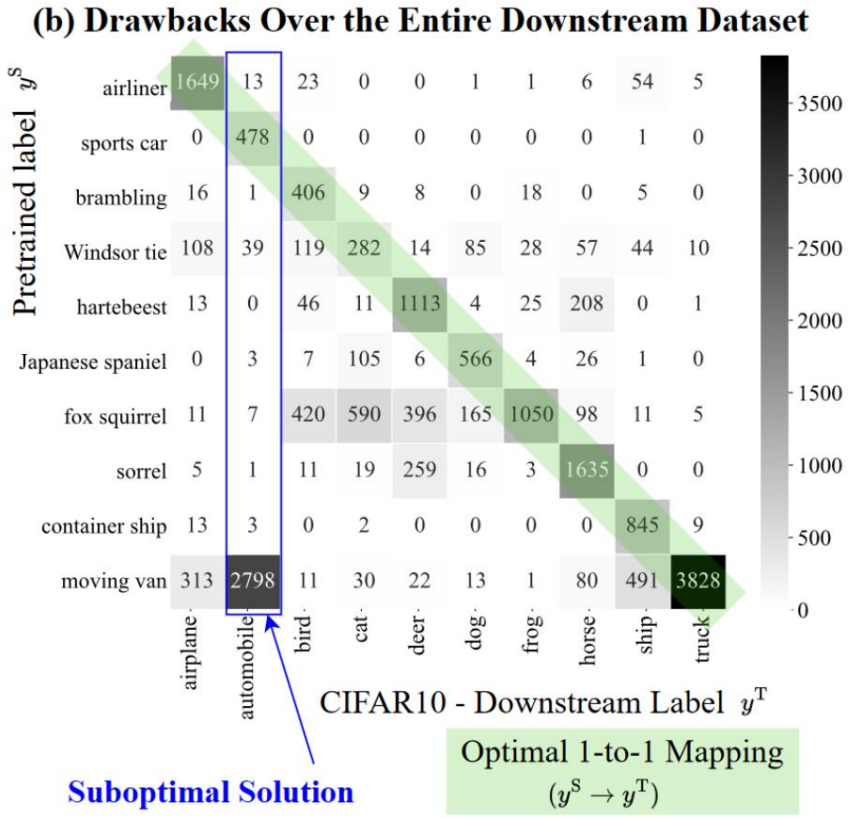


Fig. Predictive frequency distribution

[predicted pretrained labels, ground-truth downstream labels]

[1] Cai et al. Bayesian-guided Label Mapping for Visual Reprogramming. In NeurIPS 2024

Bayesian-guided Label Mapping

TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Revisit Label Mapping

- Generalize $f_{\text{out}} : \mathcal{Y}^{\text{S}} \to \mathcal{Y}^{\text{T}} \Leftrightarrow \omega \in \mathbb{R}^{k_{\text{S}} \times k_{\text{T}}}$ to a linear transformation

$$k_{\text{S}} = |\mathcal{Y}^{\text{S}}|$$
$$k_{\text{T}} = |\mathcal{Y}^{\text{T}}|$$

- Then, downstream label prediction by vector-matrix multiplication

$$\tilde{y}_i^{\text{T}} \equiv \begin{bmatrix} \tilde{y}_i^1 \\ \vdots \\ \tilde{y}_i^{k_{\text{T}}} \end{bmatrix} = f(x_i^{\text{T}})^\top \cdot \omega = \begin{bmatrix} f(x_i^{\text{T}})_1 & \dots & f(x_i^{\text{T}})_{k_{\text{S}}} \end{bmatrix} \begin{bmatrix} \omega_{1,1} & \dots & \omega_{1,k_{\text{T}}} \\ \vdots & \ddots & \vdots \\ \omega_{k_{\text{S}},1} & \dots & \omega_{k_{\text{S}},k_{\text{T}}} \end{bmatrix} \quad \text{with } f \triangleq (f_{\text{pre}} \circ f_{\text{in}})(x_i^{\text{T}}; \theta)$$

Deterministic 1-to-1 OM	→	Probabilistic many-to-many LM
$\omega \in \{0,1\}^{k_{\text{S}} \times k_{\text{T}}}$ s.t. $\sum_{j=1}^{k_{\text{S}}} \omega_{j,\cdot} = 1$		$\omega \in [0,1]^{k_{\text{S}} \times k_{\text{T}}}$ s.t. $\sum_{j=1}^{k_{\text{S}}} \omega_{j,\cdot} = 1$

[1] Cai et al. Bayesian-guided Label Mapping for Visual Reprogramming. In NeurIPS 2024

Bayesian-guided Label Mapping

From *1-to-1* mapping to probabilistic *many-to-many* mapping

$$\tilde{y}_i^{\mathrm{T}} \equiv \begin{bmatrix} \tilde{y}_i^1 \\ \vdots \\ \tilde{y}_i^{k_{\mathrm{T}}} \end{bmatrix} = f(x_i^{\mathrm{T}})^\top \cdot \omega = \begin{bmatrix} f(x_i^{\mathrm{T}})_1 & \dots & f(x_i^{\mathrm{T}})_{k_{\mathrm{S}}} \end{bmatrix} \begin{bmatrix} \omega_{1,1} & \dots & \omega_{1,k_{\mathrm{T}}} \\ \vdots & \ddots & \vdots \\ \omega_{k_{\mathrm{S}},1} & \dots & \omega_{k_{\mathrm{S}},k_{\mathrm{T}}} \end{bmatrix} \quad \text{with } f \triangleq (f_{\mathrm{pre}} \circ f_{\mathrm{in}})(x_i^{\mathrm{T}}; \theta)$$

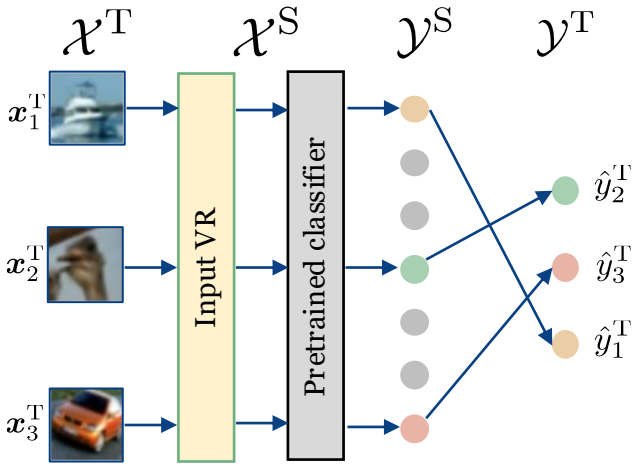


Fig. binary 1-to-1 mapping

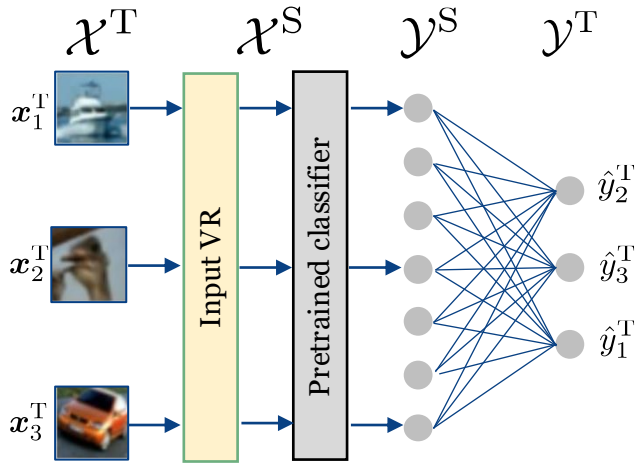


Fig. probabilistic many-to-many mapping

[1] Cai et al. Bayesian-guided Label Mapping for Visual Reprogramming. In NeurIPS 2024

Bayesian-guided Label Mapping

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Forward pass by marginalization

$$p(Y^{\mathrm{T}}|X^{\mathrm{T}}) = \sum_{y^{\mathrm{S}} \in \mathcal{Y}^{\mathrm{S}}} p(Y^{\mathrm{S}} = y^{\mathrm{S}}|X^{\mathrm{T}}) p(Y^{\mathrm{T}}|Y^{\mathrm{S}} = y^{\mathrm{S}}, X^{\mathrm{T}})$$

[1] Cai et al. Bayesian-guided Label Mapping for Visual Reprogramming. In NeurIPS 2024

Bayesian-guided Label Mapping

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Forward pass by marginalization

$$p(Y^{\mathrm{T}}|X^{\mathrm{T}}) = \sum_{y^{\mathrm{S}} \in \mathcal{Y}^{\mathrm{S}}} p(Y^{\mathrm{S}} = y^{\mathrm{S}}|X^{\mathrm{T}}) p(Y^{\mathrm{T}}|Y^{\mathrm{S}} = y^{\mathrm{S}}, X^{\mathrm{T}})$$

$$\approx \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{y^{\mathrm{S}} \in \mathcal{Y}^{\mathrm{S}}} \underbrace{p(Y^{\mathrm{S}} = y^{\mathrm{S}}|X^{\mathrm{T}} = x_i^{\mathrm{T}})}_{\text{input VR:}(f_{\mathrm{pre}} \circ f_{\mathrm{in}})(x_i^{\mathrm{T}};\theta)} \underbrace{p(Y^{\mathrm{T}} = y_i^{\mathrm{T}}|Y^{\mathrm{S}} = y^{\mathrm{S}}, X^{\mathrm{T}} = x_i^{\mathrm{T}})}_{\text{output LM:}f_{\mathrm{out}}^{\omega, y^{\mathrm{S}}}} \right)$$

[1] Cai et al. Bayesian-guided Label Mapping for Visual Reprogramming. In NeurIPS 2024

Bayesian-guided Label Mapping

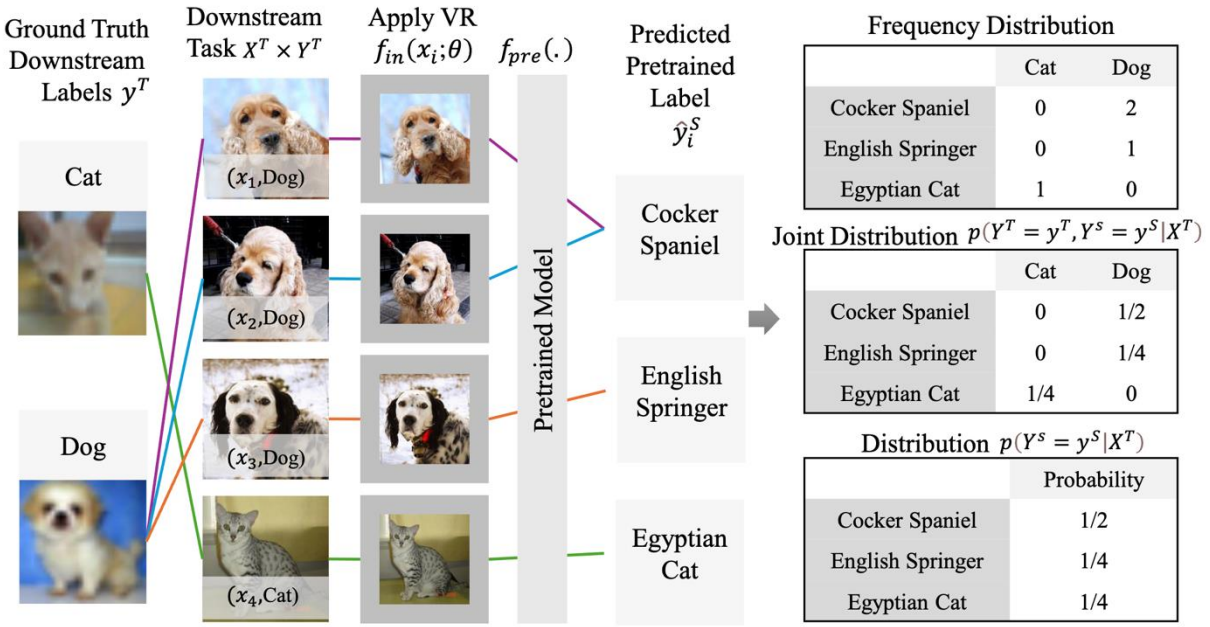
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Probabilistic mapping by *empirical approximation*

$$p(Y^{\mathrm{T}} = y^{\mathrm{T}} | Y^{\mathrm{S}} = y^{\mathrm{S}}, X^{\mathrm{T}}) = \frac{p(Y^{\mathrm{T}} = y^{\mathrm{T}}, Y^{\mathrm{S}} = y^{\mathrm{S}} | X^{\mathrm{T}})}{p(Y^{\mathrm{S}} = y^{\mathrm{S}} | X^{\mathrm{T}})}$$

- to estimate the conditional probability of interest
- illustration with a simple example



	Cat	Dog
Cocker Spaniel	0	2
English Springer	0	1
Egyptian Cat	1	0

	Cat	Dog
Cocker Spaniel	0	1/2
English Springer	0	1/4
Egyptian Cat	1/4	0

	Probability
Cocker Spaniel	1/2
English Springer	1/4
Egyptian Cat	1/4

Fig. empirical estimation of numerator and denominator

[1] Cai et al. Bayesian-guided Label Mapping for Visual Reprogramming. In NeurIPS 2024

Bayesian-guided Label Mapping

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Probabilistic mapping by empirical approximation

$$p(Y^{\mathrm{T}} = y^{\mathrm{T}} | Y^{\mathrm{S}} = y^{\mathrm{S}}, X^{\mathrm{T}}) = \frac{p(Y^{\mathrm{T}} = y^{\mathrm{T}}, Y^{\mathrm{S}} = y^{\mathrm{S}} | X^{\mathrm{T}})}{p(Y^{\mathrm{S}} = y^{\mathrm{S}} | X^{\mathrm{T}})}$$

- BLM: frequency count of joint event

BLM

$$\hat{p}_{\mathrm{BLM}}(Y^{\mathrm{T}} = y^{\mathrm{T}}, Y^{\mathrm{S}} = y^{\mathrm{S}} | X^{\mathrm{T}}) = \frac{\sum_{i=1}^{n} \mathbb{1}\{y_i^{\mathrm{T}} = y^{\mathrm{T}}\} \cdot \mathbb{1}\{\hat{y}_i^{\mathrm{S}} = y^{\mathrm{S}}\}}{n}$$

$$\hat{p}_{\mathrm{BLM}}(Y^{\mathrm{S}} = y^{\mathrm{S}} | X^{\mathrm{T}}) = \frac{\sum_{y^{\mathrm{T}} \in \mathcal{Y}^{\mathrm{T}}} \sum_{i=1}^{n} \mathbb{1}\{y_i^{\mathrm{T}} = y^{\mathrm{T}}\} \cdot \mathbb{1}\{\hat{y}_i^{\mathrm{S}} = y^{\mathrm{S}}\} + \lambda}{n + k_{\mathrm{S}} \cdot \lambda}$$

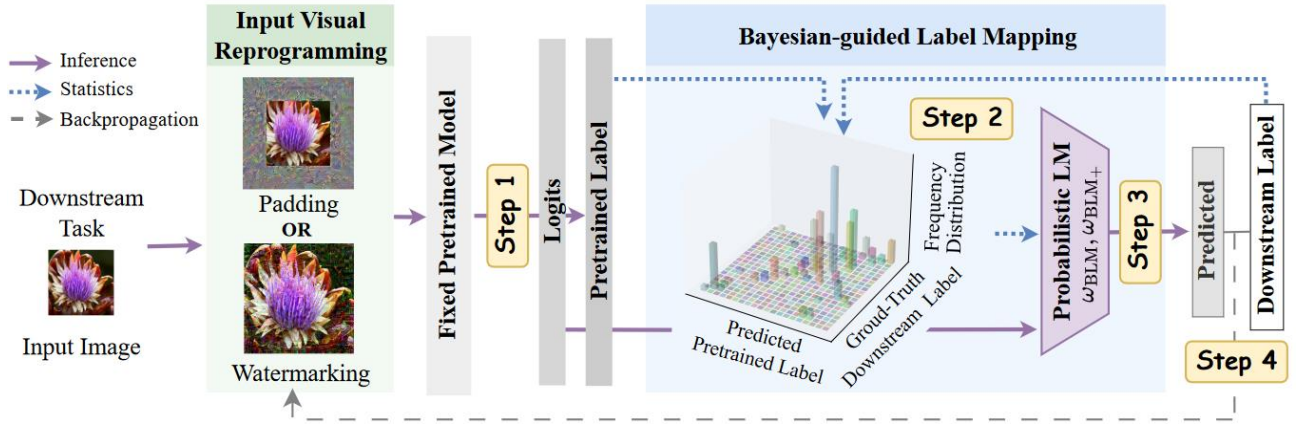


Fig. BLM (BLM+) algorithmic pipeline

[1] Cai et al. Bayesian-guided Label Mapping for Visual Reprogramming. In NeurIPS 2024

Bayesian-guided Label Mapping

TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Probabilistic mapping by empirical approximation

$$p(Y^{\mathrm{T}} = y^{\mathrm{T}} | Y^{\mathrm{S}} = y^{\mathrm{S}}, X^{\mathrm{T}}) = \frac{p(Y^{\mathrm{T}} = y^{\mathrm{T}}, Y^{\mathrm{S}} = y^{\mathrm{S}} | X^{\mathrm{T}})}{p(Y^{\mathrm{S}} = y^{\mathrm{S}} | X^{\mathrm{T}})}$$

- BLM+: truncated (i.e., top-k) full prediction

BLM+

$$\hat{p}_{\mathrm{BLM}_+}(Y^{\mathrm{T}} = y^{\mathrm{T}}, Y^{\mathrm{S}} = y^{\mathrm{S}} | X^{\mathrm{T}}) = \frac{\sum_{i=1}^{n} \mathbb{1}\{y_i^{\mathrm{T}} = y^{\mathrm{T}}\} \cdot \hat{p}(y^{\mathrm{S}} | x_i^{\mathrm{T}}) \cdot \mathbb{1}\{y^{\mathrm{S}} \in \mathcal{Y}_{K,i}^{\mathrm{S}}\}}{n}$$

$$\hat{p}_{\mathrm{BLM}_+}(Y^{\mathrm{S}} = y^{\mathrm{S}} | X^{\mathrm{T}}) = \frac{\sum_{i=1}^{n} \hat{p}(y^{\mathrm{S}} | x_i^{\mathrm{T}}) \cdot \mathbb{1}\{y^{\mathrm{S}} \in \mathcal{Y}_{K,i}^{\mathrm{S}}\} + \lambda}{n + k^{\mathrm{S}} \cdot \lambda}$$

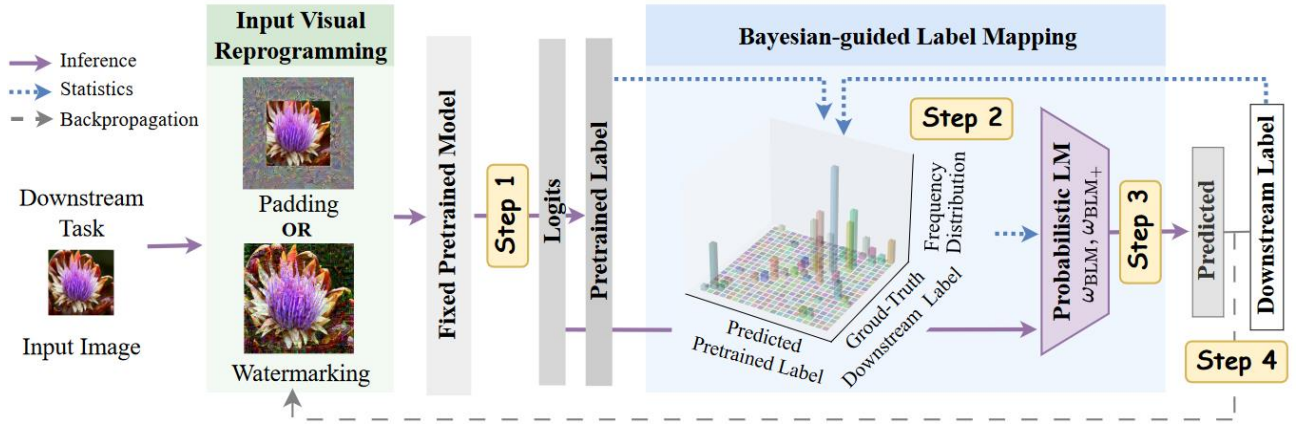


Fig. BLM (BLM+) algorithmic pipeline

[1] Cai et al. Bayesian-guided Label Mapping for Visual Reprogramming. In NeurIPS 2024

Bayesian-guided Label Mapping

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Why probabilistic many-to-many mapping?

- **consistent** performance improvement except for SVHN
- **agnostic** to different pre-trained model and IR strategies

	ResNet-18 (ImageNet-1K)						ResNeXt-101-32x8d (Instagram)				
Padding	Gradient-free					Deep	Gradient-free				Deep
Methods	RLM	FLM	ILM	BLM	BLM+	-	FLM	ILM	BLM	BLM+	-
Flowers102	11.0±0.5	20.0±0.3	27.9±0.7	44.4±1.1	**50.1**±0.6	76.7±0.2	22.5±0.5	27.9±0.3	**31.5**±0.4	30.1±0.7	85.2±1.3
DTD	16.3±0.7	32.4±0.5	35.3±0.9	42.0±0.5	**43.9**±0.4	49.1±0.3	40.3±0.5	41.4±0.7	47.8±0.4	**49.4**±0.4	64.0±0.2
UCF101	6.6±0.4	18.9±0.5	23.9±0.5	30.9±1.1	**32.0**±0.4	46.0±0.6	41.9±0.6	43.1±0.8	48.3±0.1	**50.1**±0.6	68.3±0.1
Food101	3.8±0.3	12.8±0.1	14.8±0.2	23.2±0.1	**25.1**±0.3	34.1±0.1	20.5±0.5	23.0±0.4	29.6±0.6	**31.4**±0.2	58.7±0.3
GTSRB	46.1±1.3	45.5±1.0	52.0±1.2	**54.8**±0.7	54.3±0.7	63.1±0.5	56.2±0.6	59.9±1.0	62.9±0.5	**63.0**±0.8	74.4±0.5
EuroSAT	82.4±0.4	83.8±0.2	85.2±0.6	**86.7**±0.2	**86.7**±0.1	92.4±0.1	87.8±0.4	86.2±0.8	87.6±0.3	**88.3**±0.3	93.2±0.1
OxfordPets	9.3±0.4	62.9±0.1	65.4±0.7	69.8±0.3	**70.6**±0.2	73.0±0.3	76.8±0.6	78.9±0.8	82.4±0.4	**83.0**±0.6	91.8±0.1
StanfordCars	0.9±0.1	2.7±0.1	4.5±0.1	5.4±0.1	**7.7**±0.1	14.3±0.1	4.6±0.1	7.0±0.2	8.3±0.1	**9.3**±0.3	50.5±0.5
SUN397	1.0±0.1	10.4±0.1	13.0±0.2	16.2±0.1	**18.7**±0.3	26.3±0.6	21.6±0.3	23.7±0.2	30.1±0.1	**32.0**±0.3	51.5±0.8
CIFAR10	63±0.1	65.7±0.6	65.5±0.1	66.7±0.2	**66.8**±0.2	72.1±0.8	80.3±0.3	81.7±0.3	**82.2**±0.3	**82.2**±0.1	83.4±0.1
CIFAR100	12.9±0.1	18.1±0.2	24.8±0.1	29.6±0.6	**30.6**±0.4	46.7±0.2	39.7±0.2	45.9±0.2	**47.8**±0.3	**47.8**±0.3	56.2±0.4
SVHN	73.5±0.3	73.1±0.2	**75.2**±0.2	74.5±0.7	74.2±0.3	82.1±0.2	79.0±0.5	**81.4**±0.1	79.8±0.3	79.3±0.4	85.7±0.2
Average	27.2	37.2	40.6	45.3	**46.7**	56.3	47.6	50.0	53.2	**53.8**	71.9

Fig. BLM/BLM+ performance with padding

Watermarking	Gradient-free			Deep
Methods	ILM	BLM	BLM+	-
Flowers102	23.2±0.5	39.2±0.6	**44.1**±0.9	82.4±0.4
DTD	29.0±0.7	40.1±0.2	**43.0**±0.2	48.9±0.5
UCF101	24.4±0.9	32.9±0.8	**35.4**±0.5	53.1±0.2
Food101	13.2±0.1	21.5±0.4	**22.9**±0.1	30.4±0.9
GTSRB	76.8±0.9	82.1±0.7	**82.0**±0.8	89.5±0.3
EuroSAT	84.3±0.5	84.4±0.5	**84.8**±0.2	89.2±0.2
OxfordPets	70.0±0.6	72.4±0.6	**73.3**±0.1	77.6±0.8
StanfordCars	3.4±0.1	5.5±0.1	**7.4**±0.1	30.7±0.3
SUN397	13.4±0.2	18.4±0.1	**19.4**±0.2	32.9±0.3
CIFAR10	68.9±0.4	74.9±0.2	**75.7**±0.1	71.7±0.6
CIFAR100	33.8±0.2	41.2±0.3	**41.6**±0.3	39.9±0.5
SVHN	78.3±0.3	**79.2**±0.1	78.8±0.2	83.7±0.2
Average	43.2	49.3	**50.7**	60.8

Fig. BLM/BLM+ performance with watermarking

[1] Cai et al. Bayesian-guided Label Mapping for Visual Reprogramming. In NeurIPS 2024

Bayesian-guided Label Mapping

TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Why probabilistic many-to-many mapping?

- handle semantics **similarity** and **inclusion** relationship
- with reweighting coefficients

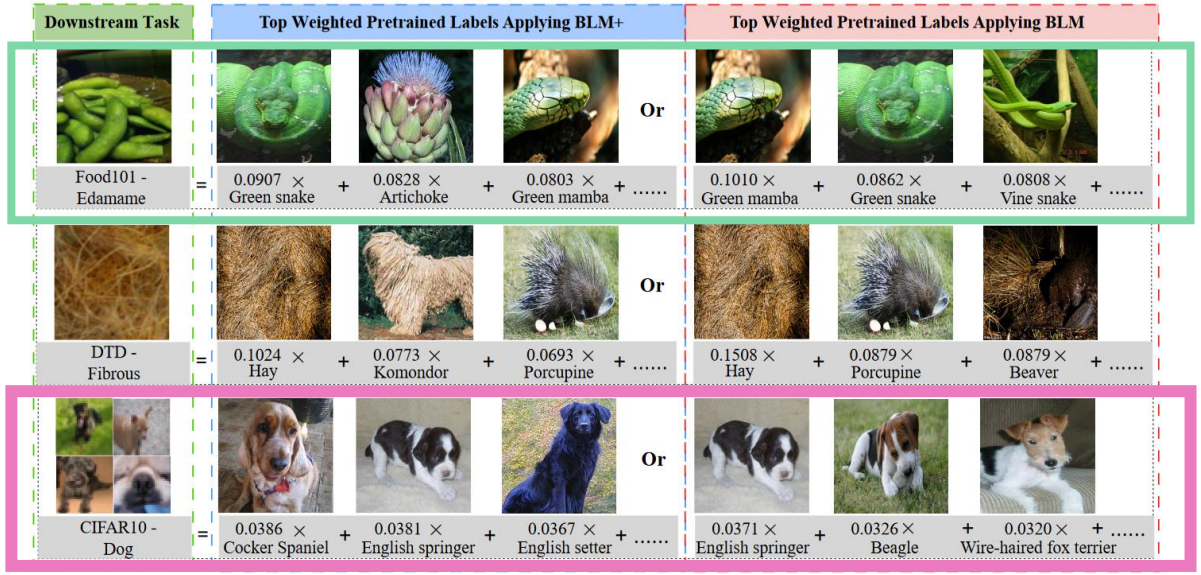


Fig. a visualization of many-to-many weights to constitute downstream label prediction

[1] Cai et al. Bayesian-guided Label Mapping for Visual Reprogramming. In NeurIPS 2024

Bayesian-guided Label Mapping

TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Why probabilistic many-to-many mapping?

- weighted aggregation of multiple pre-trained labels

  diverse semantics is considered

- *gradual refinement* of involved similar pre-trained labels from $\mathcal{Y}^S$
  - e.g., from [teddy bear] to [pineapples] (for [Marigold])

  more *aligned* color, shape, etc.

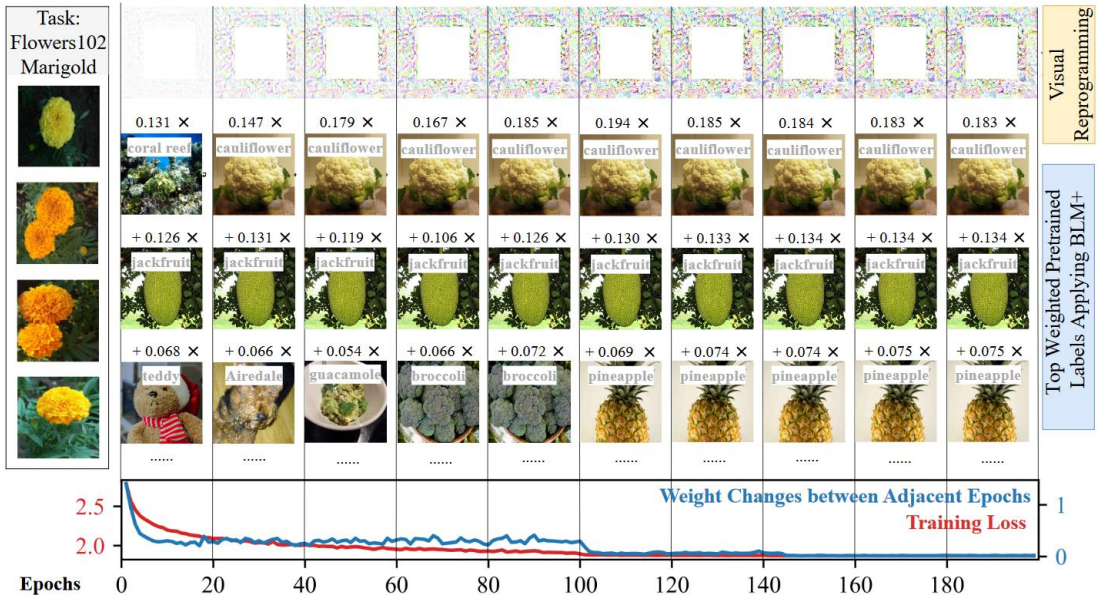
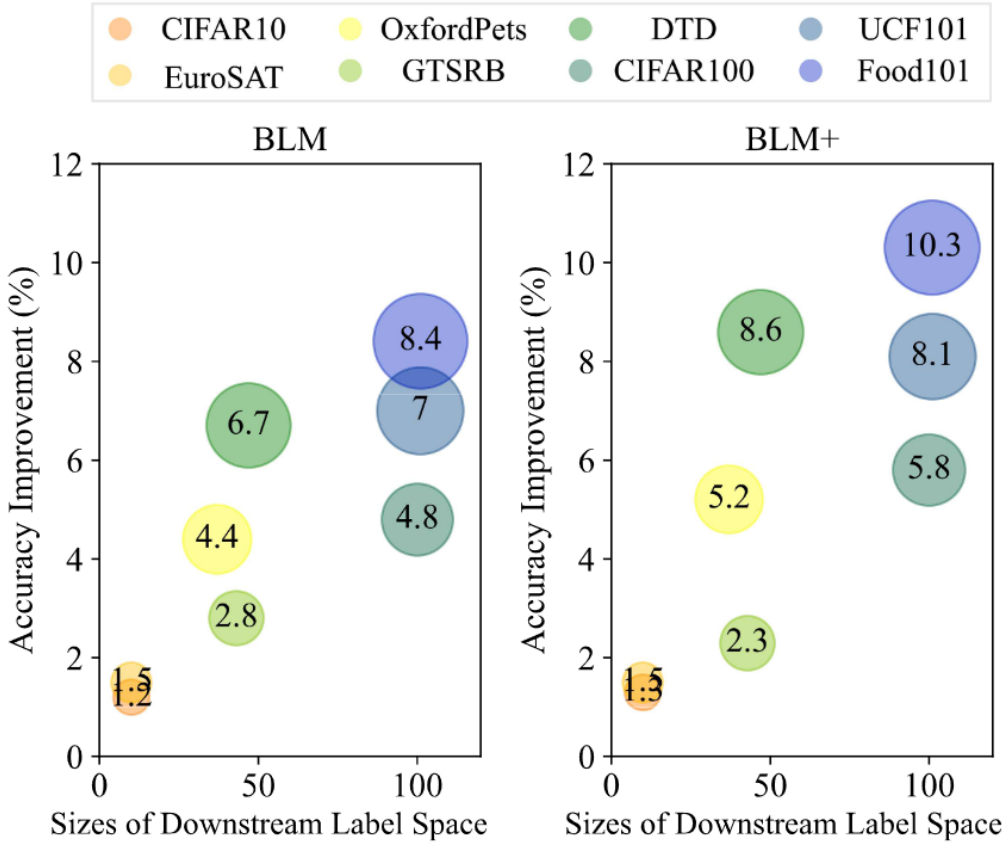
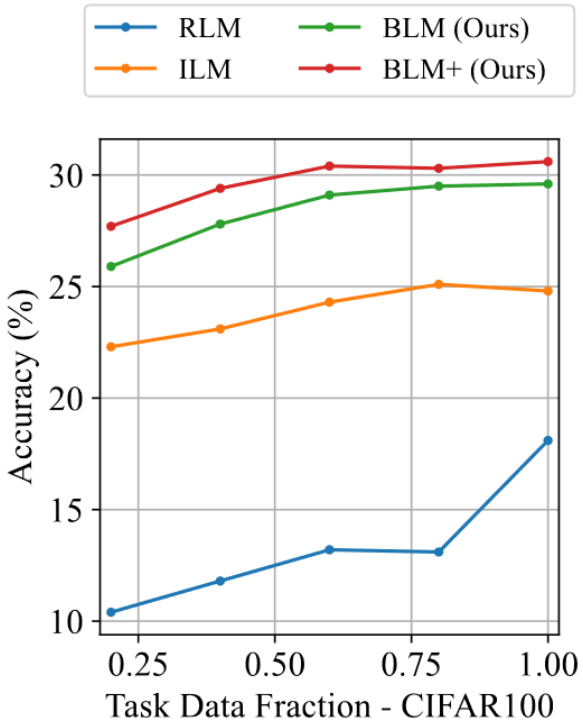


Fig. a visualization of many-to-many OM weights refinement during optimization of IR

[1] Cai et al. Bayesian-guided Label Mapping for Visual Reprogramming. In NeurIPS 2024

Bayesian-guided Label Mapping





Why probabilistic LM?

- Capacity: full potential when the **label space is larger**
- Efficiency: comparable accuracy even **with less training data**

[1] Cai et al. Bayesian-guided Label Mapping for Visual Reprogramming. In NeurIPS 2024

Bayesian-guided Label Mapping

TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Why probabilistic many-to-many mapping?

- BLM/BLM+: *sparse* reweighting matrix
- potentially more interpretable than linear probe

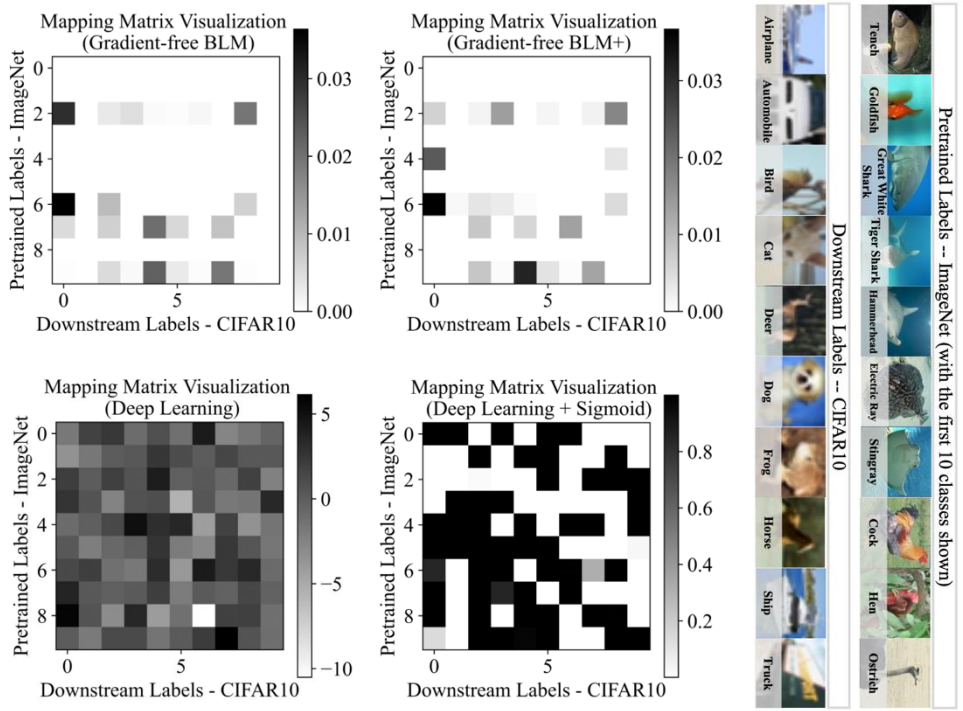


Fig. established OM weights of the first 10 rows/cols
top: BLM/BLM+; bottom: DL (linear probe)

[1] Cai et al. Bayesian-guided Label Mapping for Visual Reprogramming. In NeurIPS 2024

Beyond Label Mapping

Now, let's go back to see what we did in BLM.

First, what we used in BLM?

A pretrained model

A fully supervised data set (target/downstream task)

output

Then, what is the key in BLM?

$$p(Y^{\mathrm{T}}|X^{\mathrm{T}}) = \sum_{y^{\mathrm{S}} \in \mathcal{Y}^{\mathrm{S}}} p(Y^{\mathrm{S}} = y^{\mathrm{S}}|X^{\mathrm{T}}) p(Y^{\mathrm{T}}|Y^{\mathrm{S}} = y^{\mathrm{S}}, X^{\mathrm{T}})$$

Compute the discrepancy between both (e.g., log likelihood)

$\{(\boldsymbol{x}_i^{\boldsymbol{T}}, \boldsymbol{y}_i^{\boldsymbol{T}})\}_{i=1}^{\boldsymbol{n}}$

The meaning of this loss?

How the pretrained model performs on the target data set.

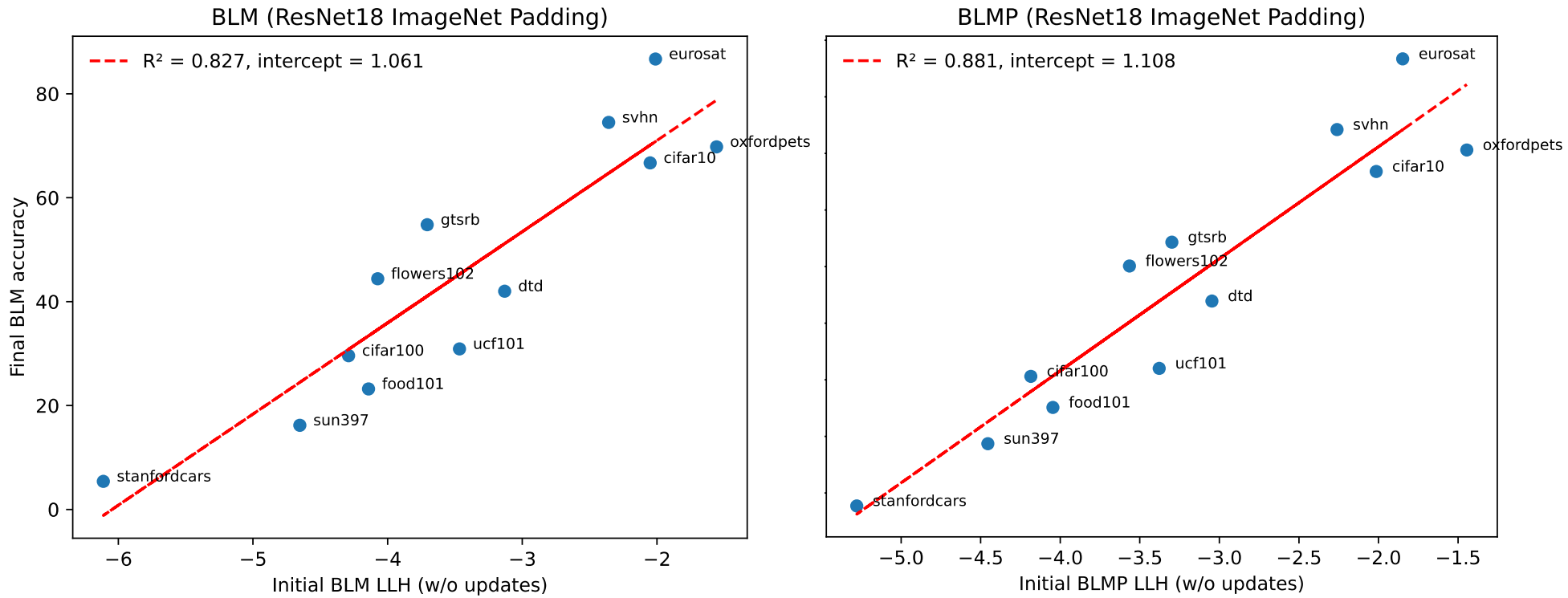
Training free!

BLM for Pre-trained Model Selection

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Correlation between BLM/BLM+ initial log likelihood (LLH) and final classification accuracy:



- ***Observation***

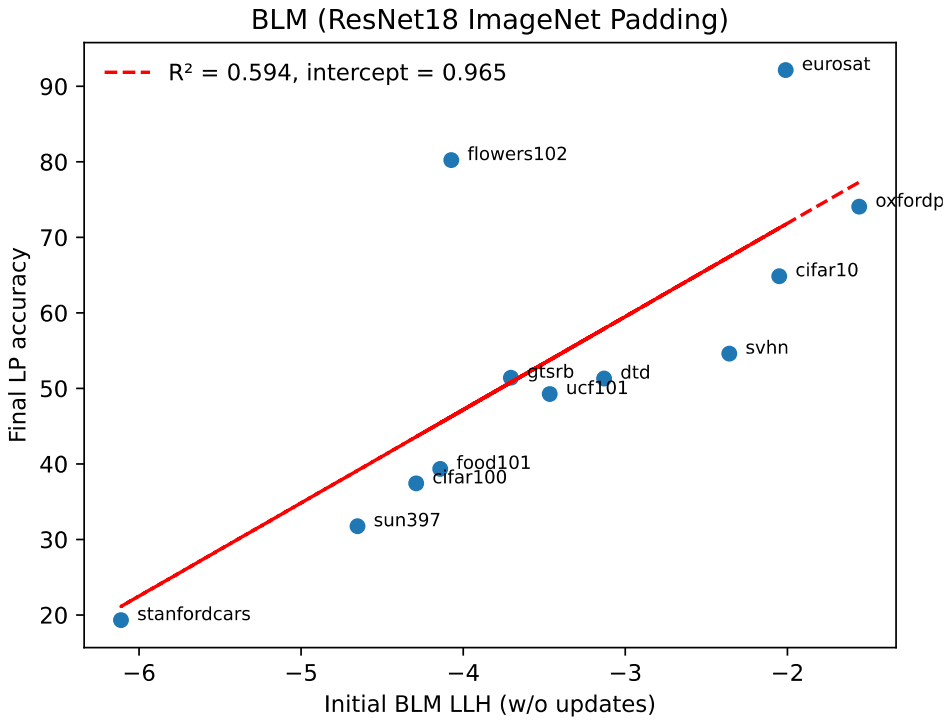
lower initial BLM likelihood => (generally) lower model VR performance

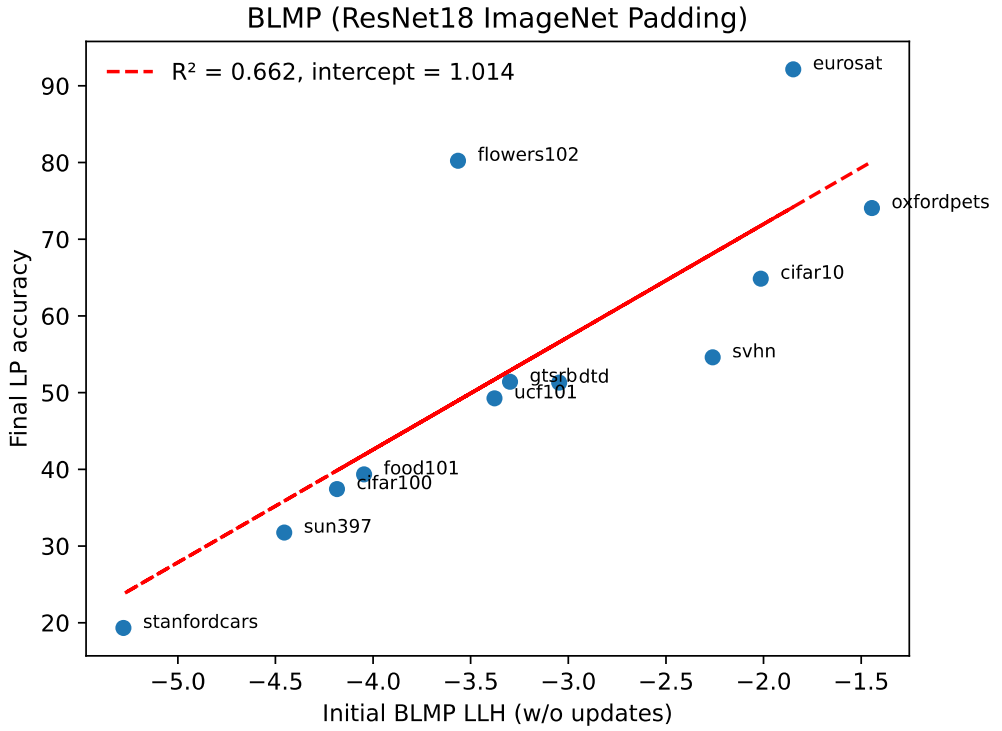
BLM for Pre-trained Model Selection

TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Correlation between BLM/BLM+ initial log likelihood (LLH) and linear probe performance:





- ***Observation*** relationship extends to linear probe (i.e., non-BLM/BLM+) performance

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

PART 2: Output Alignment

(c) Conclusions

Conclusion

TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Wrap up

- Why OA: make pre-trained model's prediction meaningful on the downstream task
- can be implemented internally (implicit) or externally (explicit)
- 1-to-1 mappings are straightforward, but lead to sub-optimality,
- probabilistic many-to-many mappings has higher interpretability with negligible overhead

Conclusion

TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Future outlook

- ❑ Can statistical-based output mapping be estimated reliably with very limited instances?
- ❑ Can other prediction tasks (e.g., segmentation) benefit from probabilistic label (output) mapping?

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

PART 3: Input Manipulation under Multi-modal Architectures

Revisit Visual Input Manipulation in VLMs

TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

For VLMs, Visual Prompt Tuning (or Visual Reprogramming)

- apply learnable components (i.e., prompts/programs) to the input on the visual branch
- "search" for optimal prompts with gradient-descent
  - *learnable* perturbation added to the image, e.g.,

- supervision signals are provided by the language branch

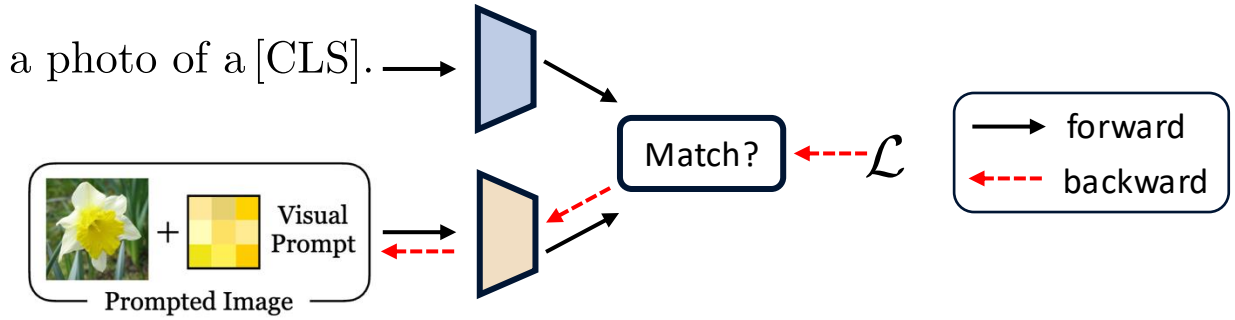


Fig. prompt tuning on image with vision-language models

Revisit Visual Input Manipulation in VLMs

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

For VLMs, Visual Prompt Tuning (or Visual Reprogramming)

- apply learnable components to the input on the visual branch
- "search" for optimal prompts with gradient-descent
  - *learnable* perturbation added to the image, e.g.,
- <u>supervision signals</u> are provided by the language branch

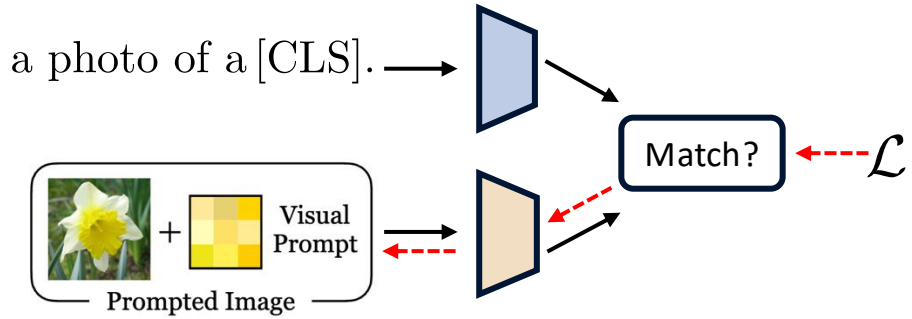
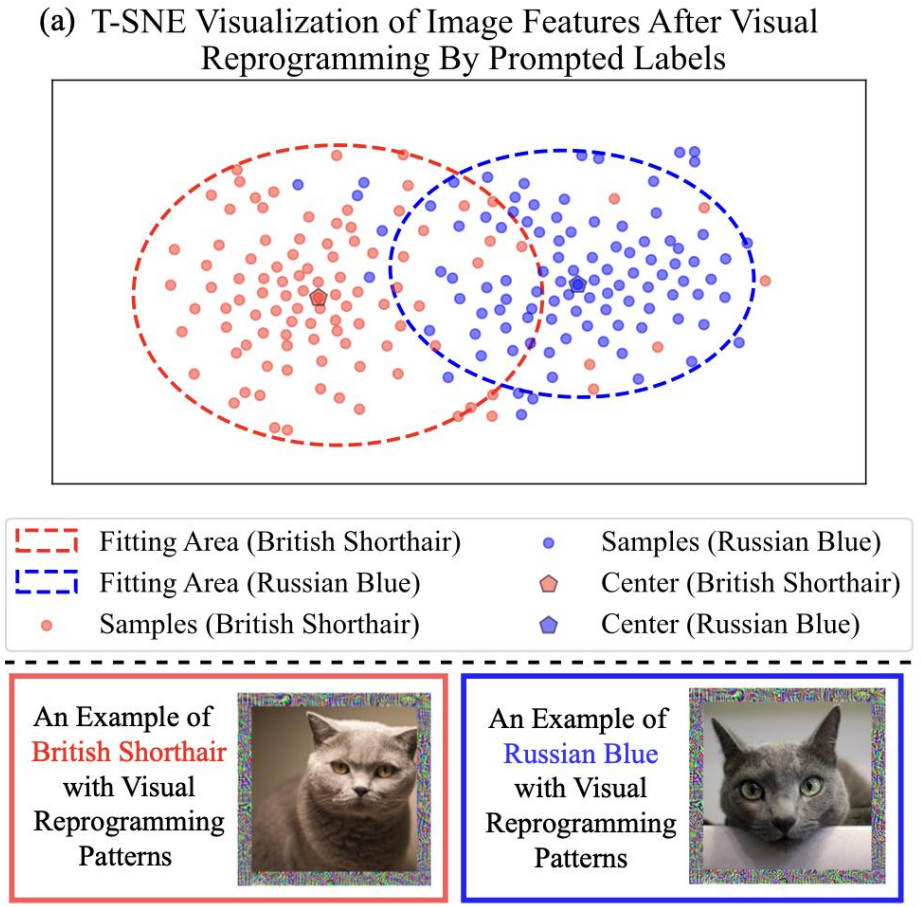


Fig. prompt tuning on image with vision-language models



[1] Cai et al. Attribute-based Visual Reprogramming for Vision-Language Models. In ICLR 2025

Revisit Visual Input Manipulation in VLMs

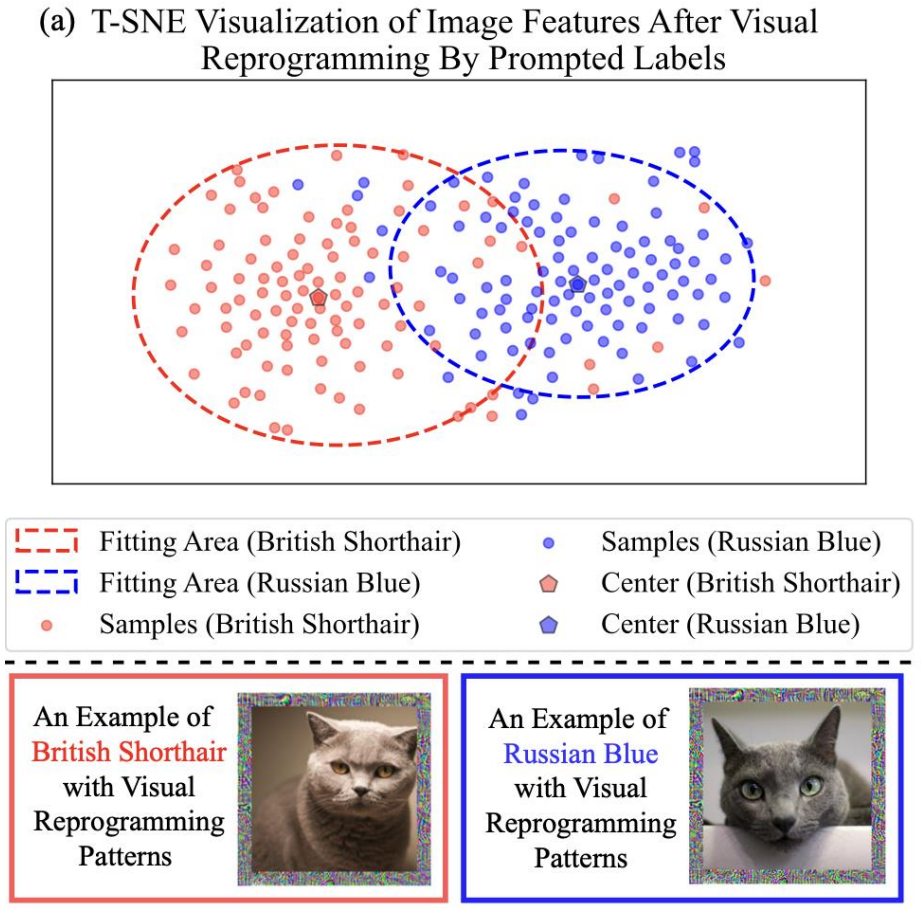
TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

For VLMs, Visual Prompt Tuning (or Visual Reprogramming)

- apply learnable components to the input on the visual branch
- "search" for optimal prompts with gradient-descent
  - *learnable* perturbation added to the image, e.g.,
- <u>supervision signals</u> are provided by the language branch
  - Class A: a photo of British Shorthair
  - Class B: a photo of Russian Blue

Similar *syntactic structure* between prompted class A & class B ==> **less separated** image features



[1] Cai et al. Attribute-based Visual Reprogramming for Vision-Language Models. In ICLR 2025

Attribute-based Visual Input Manipulation

For VLMs, Visual Prompt Tuning (or Visual Reprogramming)

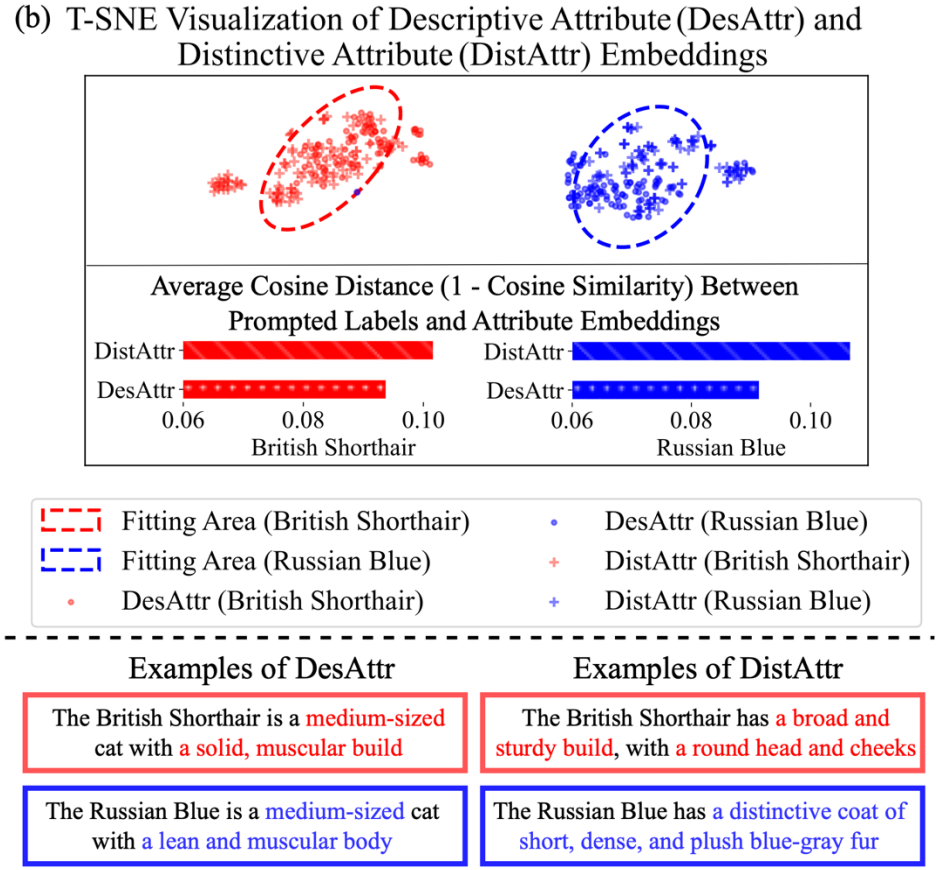
- apply learnable components to the input on the visual branch
- "search" for optimal prompts with gradient-descent
  - *learnable* perturbation added to the image, e.g.,
- supervision signals are provided by the language branch
- enrich and diversify language supervision

$$\tilde{\mathcal{A}}_{\text{des}}(y^{\text{T}}) = f_{\text{LLM}}(y^{\text{T}} \mid [\texttt{des\_prompt}])$$

  - where des_prompt = Describe the attributes of [Task] [CLASS]

$$\tilde{\mathcal{A}}_{\text{dist}}(y^{\text{T}}) = f_{\text{LLM}}(y^{\text{T}} \mid [\texttt{dist\_prompt}])$$

  - where dist_prompt = Describe the *unique* appearance of a [Class] from the other [Task]



[1] Cai et al. Attribute-based Visual Reprogramming for Vision-Language Models. In ICLR 2025

Attribute-based Visual Input Manipulation

For VLMs, Visual Prompt Tuning (or Visual Reprogramming)

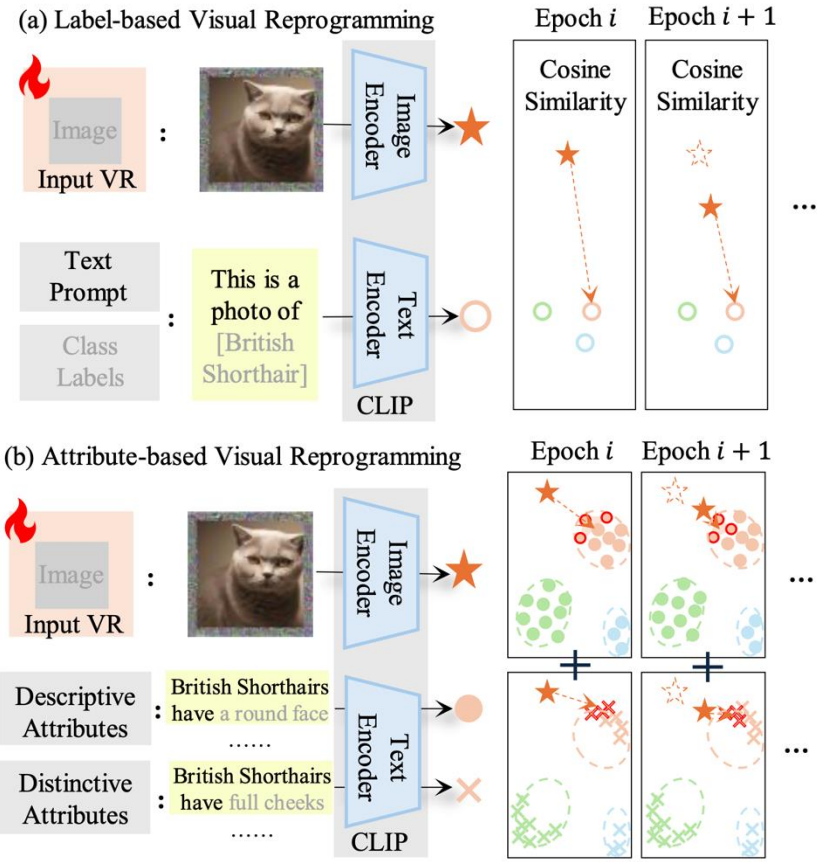
- apply learnable components to the input on the visual branch
- "search" for optimal prompts with gradient-descent
  - *learnable* perturbation added to the image, e.g.,
- supervision signals are provided by the language branch
- enrich and diversify language supervision

$$\tilde{\mathcal{A}}_{\text{des}}(y^{\text{T}}) = f_{\text{LLM}}(y^{\text{T}} \mid [\texttt{des\_prompt}])$$

  - where des_prompt = Describe the attributes of [Task] [CLASS]

$$\tilde{\mathcal{A}}_{\text{dist}}(y^{\text{T}}) = f_{\text{LLM}}(y^{\text{T}} \mid [\texttt{dist\_prompt}])$$

  - where dist_prompt = Describe the *unique* appearance of a [Class] from the other [Task]



[1] Cai et al. Attribute-based Visual Reprogramming for Vision-Language Models. In ICLR 2025

Attribute-based Visual Input Manipulation

For VLMs, Visual Prompt Tuning (or Visual Reprogramming)

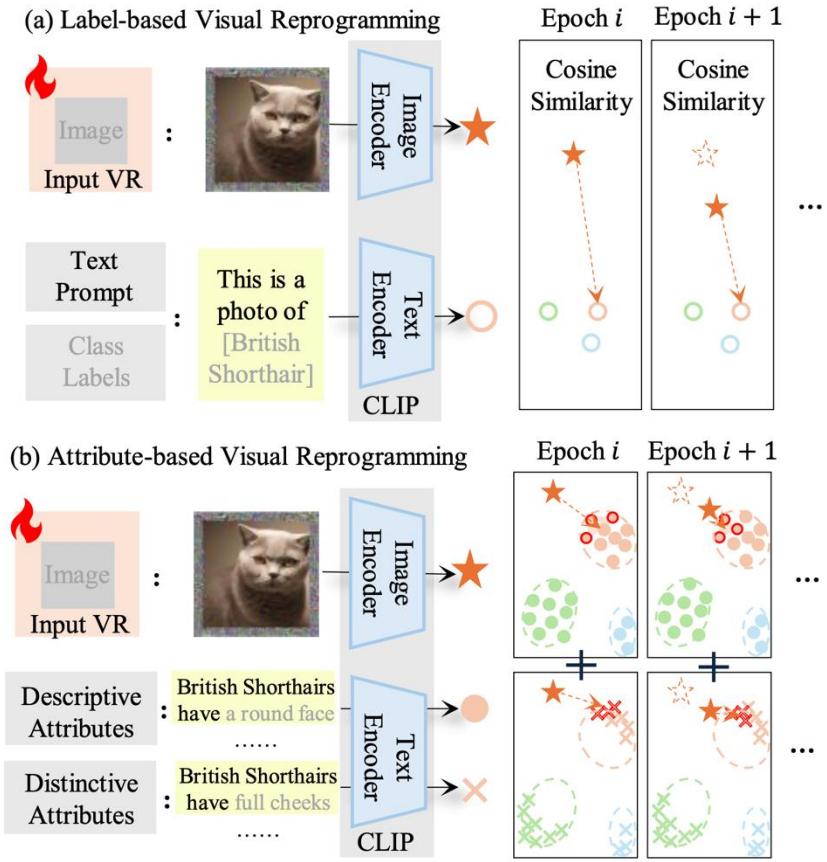
- apply learnable components to the input on the visual branch
- enrich and diversify language supervision

$$\tilde{\mathcal{A}}_{\text{des}}(y^{\text{T}}) = f_{\text{LLM}}(y^{\text{T}} \mid [\texttt{des\_prompt}])$$

  - where des_prompt = Describe the attributes of [Task] [CLASS]

$$\tilde{\mathcal{A}}_{\text{dist}}(y^{\text{T}}) = f_{\text{LLM}}(y^{\text{T}} \mid [\texttt{dist\_prompt}])$$

  - where dist_prompt = Describe the *unique* appearance of a [Class] from the other [



k-nearest iterative updating

- **why**: different [British shorthair] images have different appearance
- **what**: similar attribute embeddings vary across different samples
- **how**: query top-k nearest attribute embeddings for each sample

[1] Cai et al. Attribute-based Visual Reprogramming for Vision-Language Models. In ICLR 2025

Attribute-based Visual Input Manipulation

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Empirically, extending language supervision with semantically-rich attributes leads to …

- **Performance gain** on all fine-grained classification tasks, regardless of encoder backbones

Table 1: Accuracy comparison of different methods trained on 16-shot downstream classification tasks, using ViT-B16-based CLIP as the pre-trained model (Mean % ± Std %, ours are highlighted and the highest is in **bold**).

Method	Aircraft	Caltech	Cars	DTD	ESAT	Flowers	Food	Pets	SUN	UCF	IN	Resisc	Avg.
ZS	22.4	89.0	65.2	41.1	38.7	65.5	84.4	86.1	61.7	66.7	64.2	55.9	61.7
AttrZS	28.5	94.1	65.1	54.3	50.8	81.6	**86.5**	91.6	65.6	69.3	69.3	62.2	68.2
VP	32.1	93.5	65.5	61.4	91.2	82.5	82.3	91.0	65.8	73.8	64.2	79.1	73.5
	±0.6	±0.1	±0.3	±0.5	±0.3	±0.4	±0.1	±0.3	±0.2	±0.5	±0.1	±0.3	
AR	31.7	95.5	68.0	62.0	93.4	85.9	85.2	92.7	67.9	78.1	66.0	81.6	75.7
	±0.3	±0.2	±0.3	±0.1	±0.1	±0.7	±0.1	±0.1	±0.3	±0.2	±0.0	±0.3	
AttrVR	**36.6**	**95.7**	**68.3**	**65.6**	**93.8**	**92.9**	85.9	**93.3**	**69.6**	**79.0**	**69.4**	**82.6**	**77.7**
	±0.3	±0.1	±0.3	±0.8	±0.3	±0.4	±0.1	±0.0	±0.1	±0.6	±0.0	±0.4	

Table 2: Average accuracy of different VR methods on 12 datasets, using different backbones as CLIP visual encoders (Mean Accuracy %, ours are highlighted and the highest is in **bold**, RN stands for ResNet).

	RN50	RN101	ViT-B32	ViT-B16	ViT-L14
ZS	53.4	56.1	58.2	61.7	68.7
AttrZS	59.9	62.4	63.8	68.2	73.2
VP	53.2	57.1	67.5	73.5	61.1
AR	59.9	62.3	65.5	75.7	71.9
AttrVR	**64.2**	**66.8**	**69.1**	**77.7**	**75.5**

[1] Cai et al. Attribute-based Visual Reprogramming for Vision-Language Models. In ICLR 2025

Attribute-based Visual Input Manipulation

Empirically, extending language supervision with semantically-rich attributes leads to …

- **Performance gain** on all fine-grained classification tasks under few-shot training, regardless of # shots

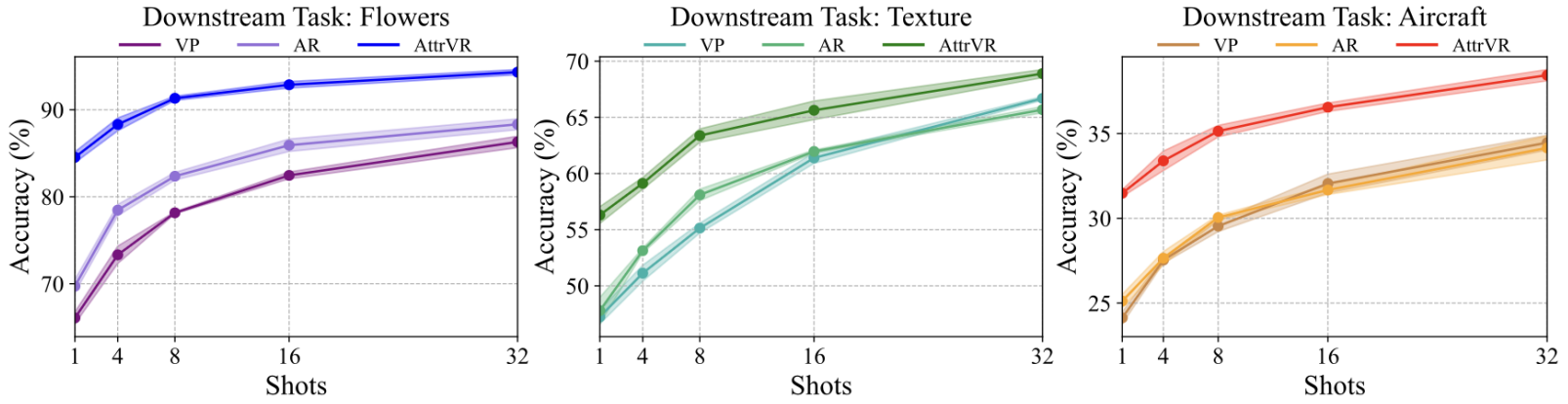


Figure 3: Accuracy comparison of different VR methods trained on different shots from [1, 4, 8, 16, 32]. Pre-trained ViT-B16-based CLIP is used. The striped area indicates the error bars.

[1] Cai et al. Attribute-based Visual Reprogramming for Vision-Language Models. In ICLR 2025

Attribute-based Visual Input Manipulation

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Empirically, extending language supervision with semantically-rich attributes leads to …

- **Well-separated** class-wise image features in the joint embedding space

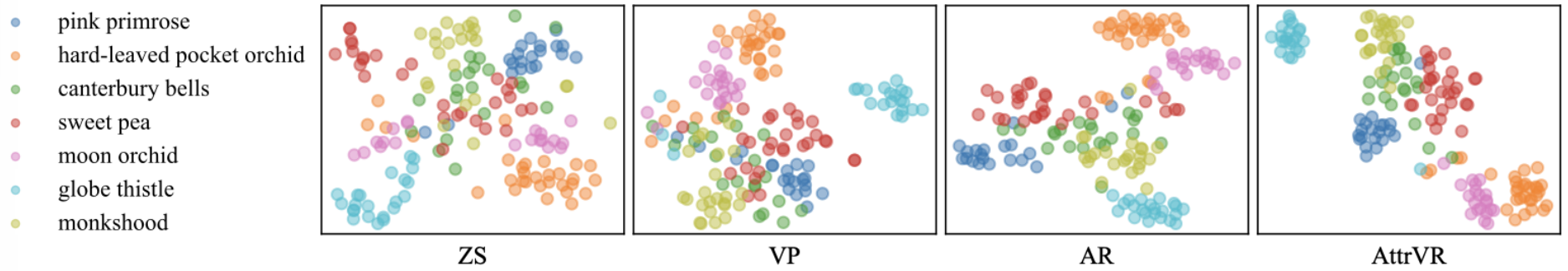


Figure 5: T-SNE visualization results of image embeddings from seven classes in the Flowers task, utilizing the ViT-B16-based CLIP as the pre-trained model. In the first plot, embeddings of zero-shot images are indicated with *ZS*. The following three plots display embeddings of images with VR patterns, categorized by different training methods and marked as *VP*, *AR*, and *AttrVR*, respectively.

[1] Cai et al. Attribute-based Visual Reprogramming for Vision-Language Models. In ICLR 2025

Attribute-based Visual Input Manipulation

TRUSTWORTHY MACHINE LEARNING AND REASONING

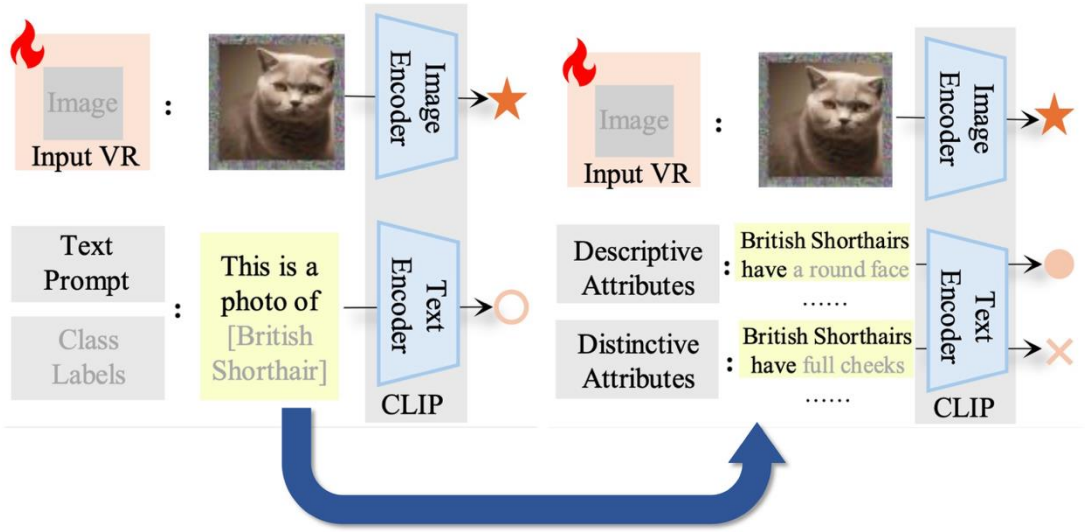
THE UNIVERSITY OF MELBOURNE

Wrap up

(Visual) IM for VLMs

- "search" for optimal visual prompts with gradient-descent
  - *learnable* perturbation added to the image, e.g.,
  - supervision signals are provided by language
  - **align** prompted *image* embedding with *textual* embedding of [CLASS]

- *Attribute set* as semantically-meaningful language supervision
  - powerful LLMs can describe visual features w/ details
  - improve *learnable visual* IM but also zero-shot fixed language IM [2-4]



Similar *syntactic structure* between class A & B => **less separated** image features

Diverse *syntactic structure & semantics* between classes => **more separated** image features

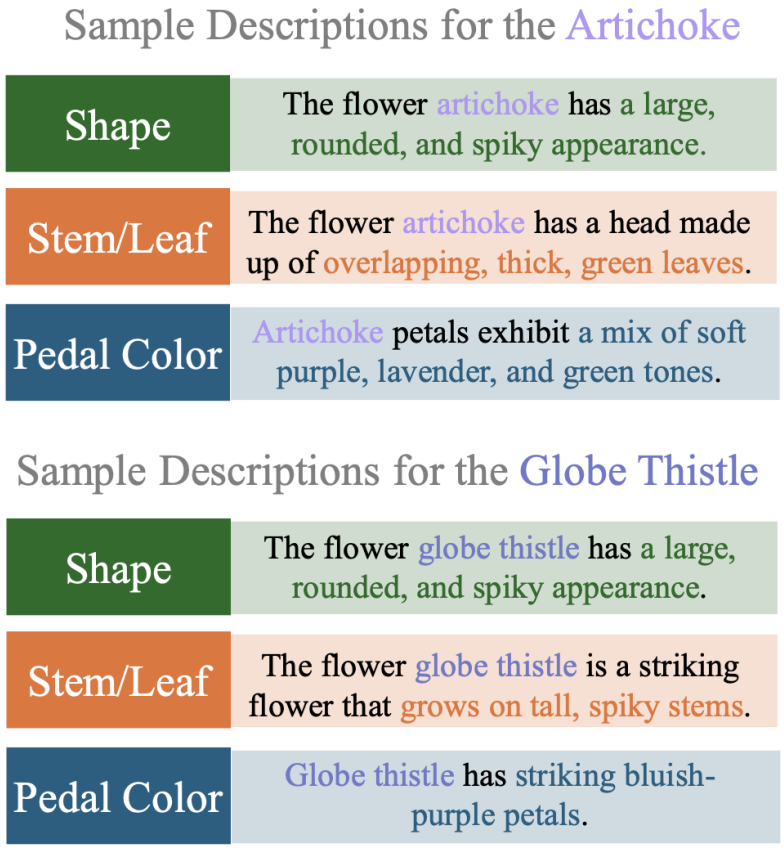
[1] Cai et al. Attribute-based Visual Reprogramming for Vision-Language Models. In ICLR 2025
[2] Pratt et al. What does a platypus look like? generating customized prompts for zero-shot image classification. In ICCV 2023
[3] Yang et al. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In CVPR 2023
[4] Li et al. Visual-text cross alignment: Refining the similarity score in vision-language models. In ICML 2024

Attribute-based Visual Input Manipulation

However

(Visual) IM for VLMs

- "search" for optimal visual prompts with gradient-descent
  - *learnable* perturbation added to the image, e.g.,
  - supervision signals are provided by language
  - **align** prompted *image* embedding with *textual* embedding of [CLASS]
- *Attribute set* as semantically-meaningful language supervision
  - powerful LLMs can describe visual features w/ details
  - improve *learnable visual* IM but also zero-shot fixed language IM [2-4]
- Optimizing a single VP/IM weight for a downstream task



[1] Cai et al. Attribute-based Visual Reprogramming for Vision-Language Models. In ICLR 2025
[2] Pratt et al. What does a platypus look like? generating customized prompts for zero-shot image classification. In ICCV 2023
[3] Yang et al. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In CVPR 2023
[4] Li et al. Visual-text cross alignment: Refining the similarity score in vision-language models. In ICML 2024

Attribute-based Visual Input Manipulation

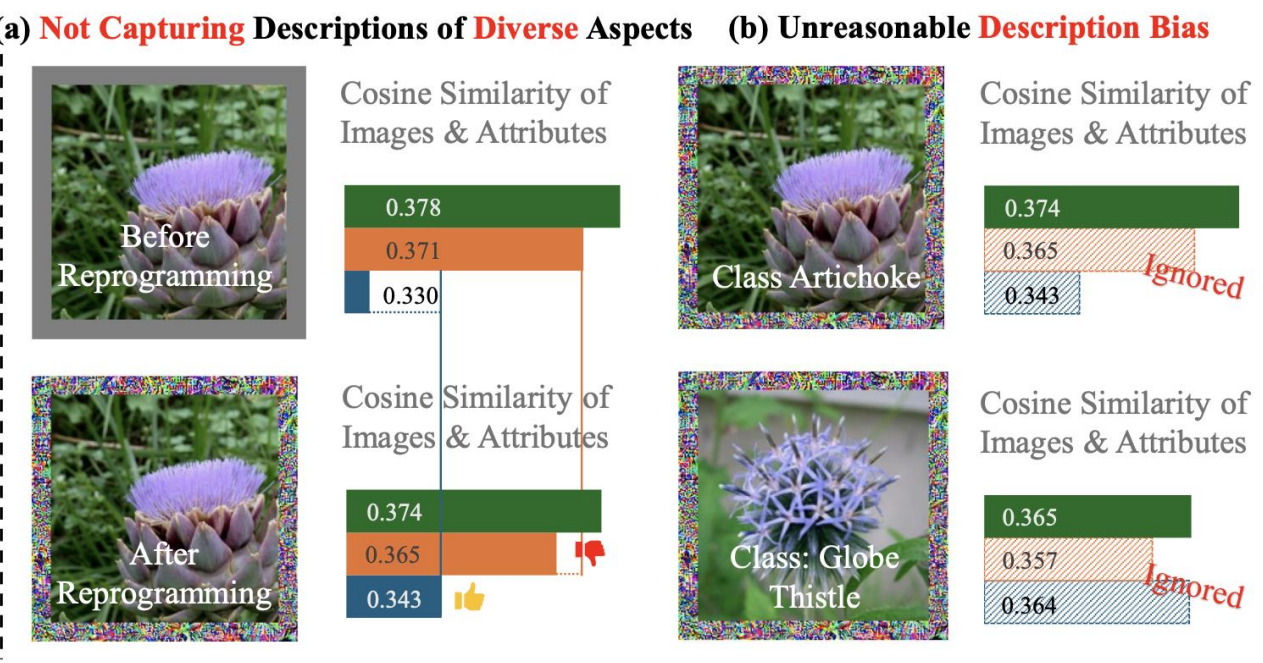
Can single VP align with all descriptions? – No ☹

Sample Descriptions for the Artichoke

Shape	The flower artichoke has a large, rounded, and spiky appearance.
Stem/Leaf	The flower artichoke has a head made up of overlapping, thick, green leaves.
Pedal Color	Artichoke petals exhibit a mix of soft purple, lavender, and green tones.

Sample Descriptions for the Globe Thistle

Shape	The flower globe thistle has a large, rounded, and spiky appearance.
Stem/Leaf	The flower globe thistle is a striking flower that grows on tall, spiky stems.
Pedal Color	Globe thistle has striking bluish-purple petals.



[1] Cai et al. Understanding Model Reprogramming for CLIP via Decoupling Visual Prompts. In ICML 2025

Attribute-based Visual Input Manipulation

TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

(Visual) IM for VLMs

- "search" for optimal visual prompts with gradient-descent
  - **align** prompted *image* embedding with *textual* embedding of [CLASS]

- *Attribute set* as semantically-meaningful language supervision
  - single VP **cannot align** image embeddings with all aspects of [CLASS] descritions

➔ Need ***multiple*** VPs to handle ***multiple*** diverse aspects ➔

TL;DR – should and how we separately optimize each VP against one aspect?

[1] Cai et al. Understanding Model Reprogramming for CLIP via Decoupling Visual Prompts. In ICML 2025

Attribute-based Visual Input Manipulation

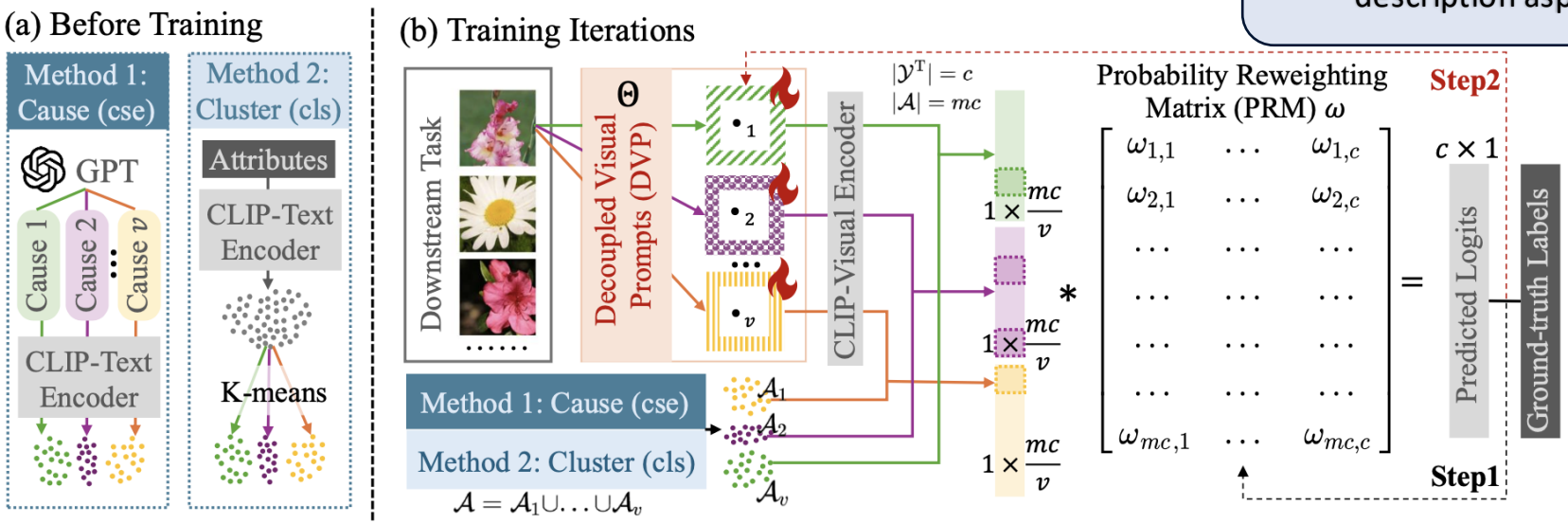
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

(Visual) IM for VLMs

- "search" for optimal visual prompts with gradient-descent
  - **align** prompted *image* embedding with *textual* embedding of [CLASS]
  - single VP **cannot align** image embeddings with all aspects of [CLASS] descriptions

Decouple the training of VPs: each VP focuses on its own description aspect



[1] Cai et al. Understanding Model Reprogramming for CLIP via Decoupling Visual Prompts. In ICML 2025

Attribute-based Visual Input Manipulation

TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Decoupled Visual IM for VLMs

- Generally, this further leads to performance gain in terms of downstream tasks over AttrVR [2]

Table 1. Accuracy comparison of different methods trained on 16-shot downstream classification tasks, using ViT-B16-based CLIP as the pretrained model (Mean % ± Std %, ours are highlighted and the highest is in **bold**). See Appendix C.7 for parameter numbers.

METHOD	AIRCRAFT	CALTECH	CARS	DTD	ESAT	FLOWERS	FOOD	PETS	SUN	UCF	RESISC	AVG.
VP	32.1±0.6	93.5±0.1	65.5±0.3	61.4±0.5	91.2±0.3	82.5±0.4	82.3±0.1	91.0±0.3	65.8±0.2	73.8±0.5	79.1±0.3	74.4
AR	31.7±0.3	95.5±0.2	68.0±0.3	62.0±0.1	93.4±0.1	85.9±0.7	85.2±0.1	92.7±0.1	67.9±0.3	78.1±0.2	81.6±0.3	76.5
ATTRVR	36.6±0.3	95.7±0.1	68.3±0.3	65.6±0.8	93.8±0.3	92.9±0.4	**85.9**±0.1	**93.3**±0.0	69.6±0.1	79.0±0.6	82.6±0.4	78.5
DVP-CSE	**40.3**±0.2	**96.2**±0.1	**72.5**±0.2	**66.7**±0.4	93.9±0.1	**95.4**±0.1	85.6±0.1	93.1±0.0	**71.1**±0.2	81.7±0.3	**84.6**±0.4	**80.1**
DVP-CLS	38.7±0.4	96.0±0.0	70.8±0.2	65.5±0.7	**94.1**±0.3	95.0±0.2	85.7±0.0	**93.3**±0.1	**71.1**±0.2	**82.0**±0.1	84.4±0.0	79.7

[1] Cai et al. Understanding Model Reprogramming for CLIP via Decoupling Visual Prompts. In ICML 2025
[2] Cai et al. Attribute-based Visual Reprogramming for Vision-Language Models. In ICLR 2025

Attribute-based Visual Input Manipulation

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Decoupled Visual IM for VLMs

- Generally, this further leads to performance gain in terms of downstream tasks over AttrVR [2]
- without necessarily using more trainable parameters

Table 14. Accuracy comparison of our DVPlite and the best-performing baseline method AttrVR trained on 16-shot downstream classification task, using ViT-B16-based CLIP as the pretrained model (Mean %, ours is highlighted and the highest is in **bold**).

	Aircraft	Caltech	Cars	DTD	ESAT	Flowers	Food	Pets	SUN	UCF	Resisc	Avg.
AttrVR (Baseline)	36.6	95.7	68.3	65.6	**93.8**	92.9	**85.9**	93.3	69.6	79	82.6	78.5
DVPlite	**39.3**	**95.9**	**71.4**	**66.5**	**93.8**	**95.2**	85.8	**93.4**	**71.6**	**81**	**83.6**	**79.8**

Table 15. A comparison of parameter numbers for different methods and their average accuracy across 11 datasets, using ViT-B16-based CLIP as the pretrained model (Mean %, ours are highlighted .

	VP	AR	AttrVR	DVP-cse	DVP-cls	DVPlite
Parameter Numbers	0.07M	0.04M	0.04M	0.12M	0.04-0.12M	0.04M
Average accuracy over 11 tasks	74.4	76.5	78.5	80.1	79.7	79.8

[1] Cai et al. Understanding Model Reprogramming for CLIP via Decoupling Visual Prompts. In ICML 2025
[2] Cai et al. Attribute-based Visual Reprogramming for Vision-Language Models. In ICLR 2025

Attribute-based Visual Input Manipulation

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Decoupled Visual IM for VLMs

- Simply using more VPs does **NOT** necessarily benefit the performance

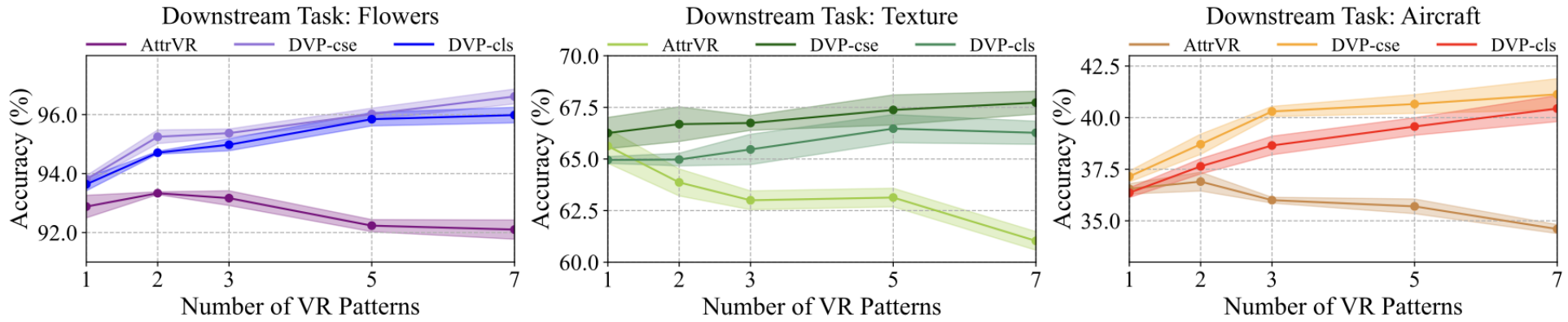


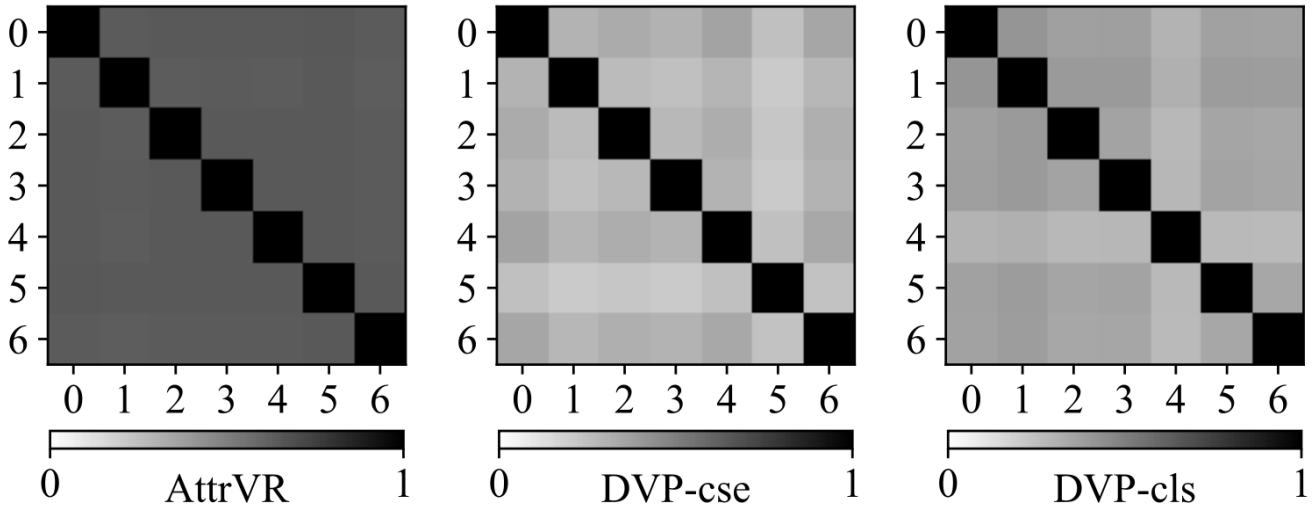
Figure 4. Accuracy comparison of different VR methods with the number of VR patterns (i.e., VPs) $v \in \{1, 2, 3, 5, 7\}$. Pre-trained ViT-B16-based CLIP is used. The striped area indicates the error bars.

[1] Cai et al. Understanding Model Reprogramming for CLIP via Decoupling Visual Prompts. In ICML 2025
[2] Cai et al. Attribute-based Visual Reprogramming for Vision-Language Models. In ICLR 2025

Attribute-based Visual Input Manipulation

Decoupled Visual IM for VLMs

- Lower pairwise dependency between VPs, using the same training objective as of AttrVR [2]



[1] Cai et al. Understanding Model Reprogramming for CLIP via Decoupling Visual Prompts. In ICML 2025
[2] Cai et al. Attribute-based Visual Reprogramming for Vision-Language Models. In ICLR 2025

Attribute-based Visual Input Manipulation

TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Decoupled Visual IM for VLMs

- Towards better understanding of the role of VPs in reprogramming

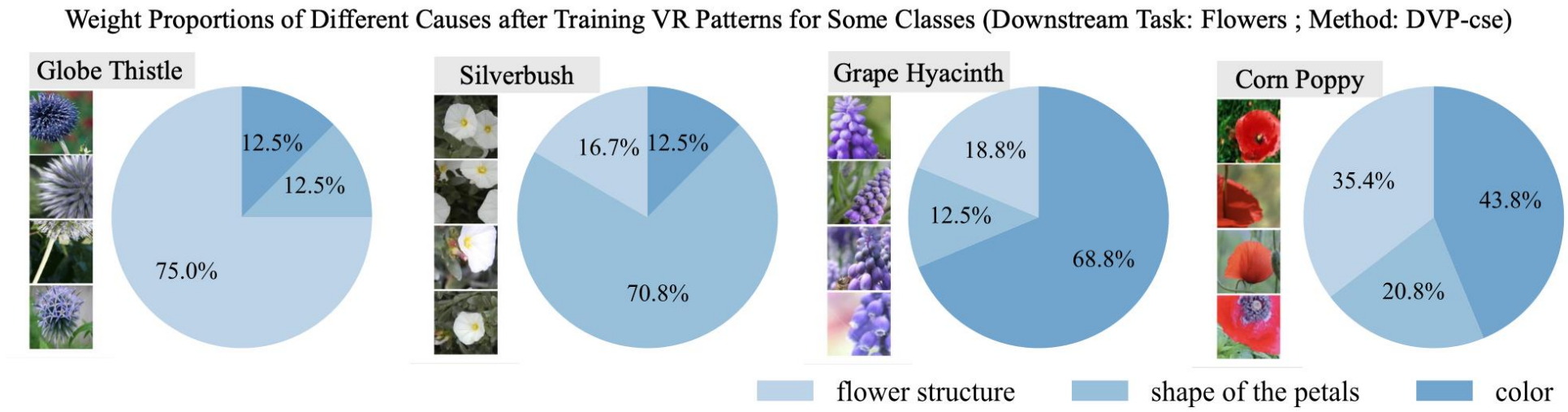


Figure 5. Visualization results of DVP-cse, showing weights of causes for classifying certain classes, using ViT-B16-based CLIP.

[1] Cai et al. Understanding Model Reprogramming for CLIP via Decoupling Visual Prompts. In ICML 2025

Attribute-based Visual Input Manipulation

TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Decoupled Visual IM for VLMs

- Towards better understanding of the role of VPs in reprogramming

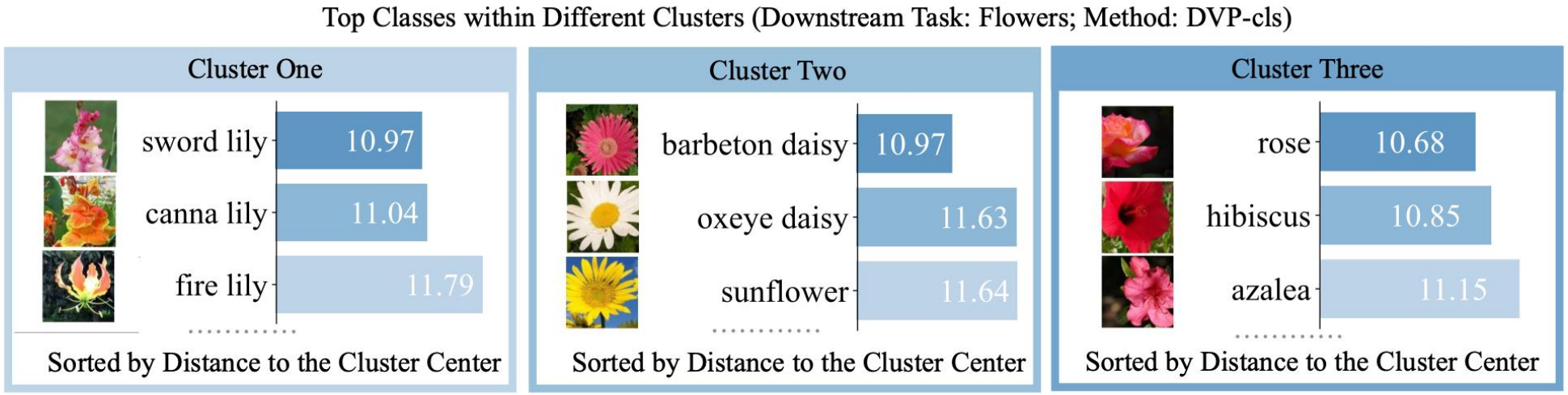


Figure 6. Visualization results of DVP-cls, showing the top 3 classes closest to the description cluster centers, using ViT-B16-based CLIP.

[1] Cai et al. Understanding Model Reprogramming for CLIP via Decoupling Visual Prompts. In ICML 2025

Attribute-based Visual Input Manipulation

TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Wrap up

(Visual) IM for VLMs relies on semantically-aligned language supervision

- *Attribute set* as semantically-meaningful language supervision
  - powerful LLMs can describe visual features w/ details
  - improve *learnable visual* IM but also zero-shot fixed language IM [2-4]

- Visual IM/Visual Prompt itself also needs to be empowered
  - single IM/VP cannot handle all semantics ➔ multiple, decoupled VPs can be a solution
  - lens to understand the role of each VP and inspire better-informed decision making

[1] Cai et al. Attribute-based Visual Reprogramming for Vision-Language Models. In ICLR 2025
[2] Pratt et al. What does a platypus look like? generating customized prompts for zero-shot image classification. In ICCV 2023
[3] Yang et al. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In CVPR 2023
[4] Li et al. Visual-text cross alignment: Refining the similarity score in vision-language models. In ICML 2024

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

PART 4: Useful Resources

Useful Resources

TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Hub for Neural Network Reprogrammability

- What is this?
  - a curated collection of resources for Reprogrammability
  - a go-to place to find related papers, tools, and datasets

- Why this repo?
  - more than just a list; built upon a unified framework

$$\hat{y}^{\mathrm{T}} = O_\omega \circ f \circ I_{\lambda,\tau,\ell}(x^{\mathrm{T}}, c)$$

  - all resources are contextualized within a four-dimensional taxonomy
    - I_λ: configuration (what manipulations)
    - I_ℓ: location (where to place manipulations)
    - I_τ: operator (how manipulations are applied)
    - O_ω: alignment (whether output alignment is needed)

Link to survey paper arXiv

Link to github repo

https://github.com/zyecs/awesome-reprogrammability

Star the repo to stay updated!

★ Star Star

Useful Resources

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Types of assets (under construction … stay tuned!)

- 📚 Resources by Type
  - 📄 Research Papers
  - 🛠️ Tools & Libraries
  - 📊 Datasets & Benchmarks
  - 📖 Educational Resources

Link to survey paper arXiv

Link to github repo

https://github.com/zyecs/awesome-reprogrammability

Star the repo to stay updated!

★ Star Star

Useful Resources

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Beyond a list

- ❑ Explore by Taxonomy
  - Don't just browse; you can filter resources based on your interests
  - Want to find methods that modify *input space*? Or those only using *additive operator*?
  - The repo is structured to answer these questions
- ❑ Applications & Use cases
  - Show how reprogrammability is applied in science
  - Covers CV, NLP, Audio, and Scientific domains (e.g., protein prediction)
- ❑ Structure Learning path
  - beginner: hands-on exercise to get familiar with implementing reprogrammability
  - intermediate: off-the-shelf reprogrammability library to facilitate key results reproduce
  - advanced: identify research gaps and contribute to the field!

Link to survey paper arXiv

Link to github repo

https://github.com/zyecs/awesome-reprogrammability

Star the repo to stay updated!

★ Star Star

We welcome thoughts and feedback!

If you have a valuable resource or suggestions, please submit a pull request or raise an issue.

THE UNIVERSITY OF
MELBOURNE

Thank you

zeshengye.ml@gmail.com
https://zyecs.github.io/