THE UNIVERSITY OF MELBOURNE

Neural Network Reprogrammability

A Unified View of Model Reprogramming, Prompt Tuning, and In-Context Learning

Feng Liu

School of Computing and Information Systems

The University of Melbourne

Date: 16/July/2026 (FLINS-ISKE 2026 Tutorial)

About the Lecturer

TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

- **Name:** Feng Liu
- **Position:** Senior Lecturer in Machine Learning, Director of Melbourne TMLR Group
- **Major Awards:** NeurIPS Outstanding Paper Award, ARC Early Career Researcher Award
- **Research Interests:** Statistical Hypothesis Testing, Distribution Shift Detection, Learning under Distribution Shift

Distribution Shift

Distribution Shift	Adversarial Shift	**Defense: Detection/Robust to/Purified**
	Out-of-distribution Shift	**Detection/Generalization**
	Natural Shift	**Adaptation/Transfer Learning**

Foundation models are emerging

Foundation models are getting powerful

- large-scale pre-training on massive and diverse datasets

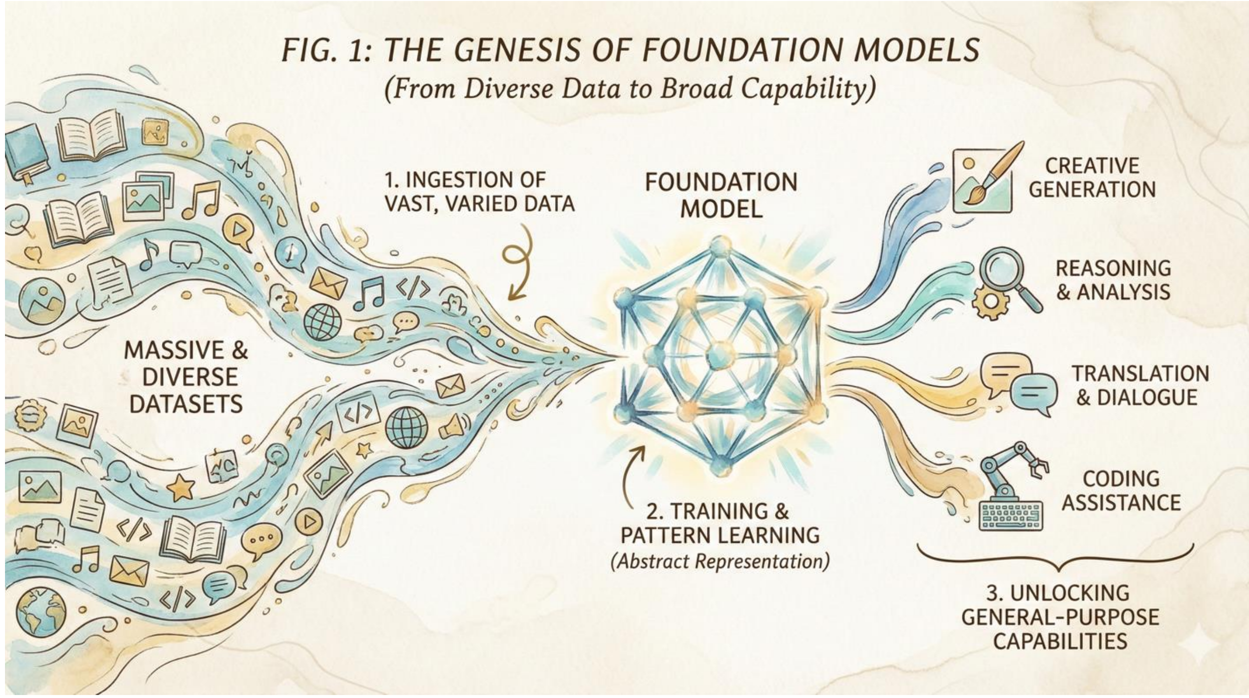


Fig. large-scale pre-training of Foundation models;
generated by Foundation model itself ☺ (credits to Nano Banana Pro)

Foundation models are emerging

Foundation models are getting powerful

- large-scale pre-training on massive and diverse datasets
- achieve remarkable performance (e.g., ChatGPT, Stable-Diffusion)

help me plan a 1-day trip in Adelaide, summarized in two sentences.

Spend the morning exploring **Glenelg Beach** and its coastal walk, then head into the city for lunch at the **Central Market** followed by a relaxed stroll through the **Botanic Garden** and nearby **Art Gallery of South Australia**. End the day at **Mount Lofty Summit** for sunset views over Adelaide and dinner in the Adelaide Hills.

Fig. Examples of Foundation models usage

Fine-tuning Foundation models?

Foundation models are getting powerful

- general-purpose representation that can be adapted to downstream tasks

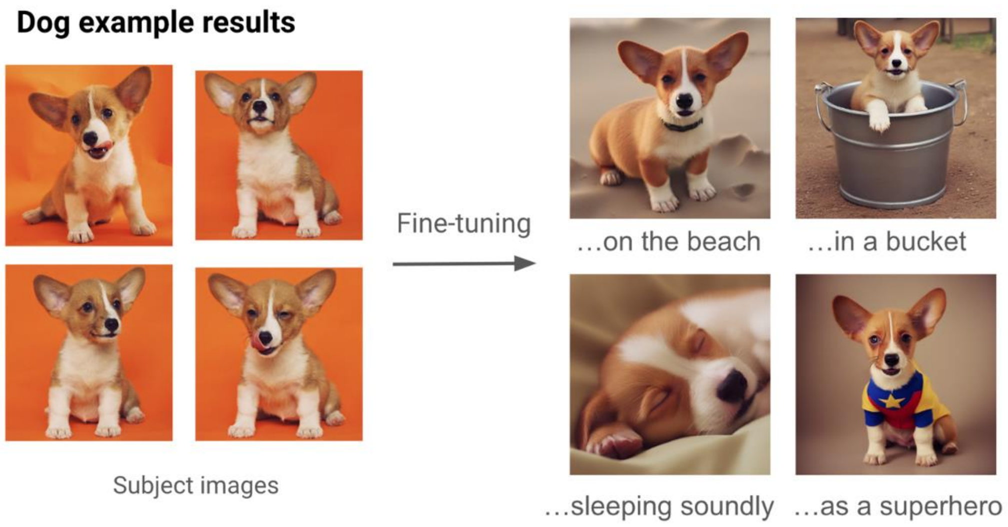


Fig. Examples of Foundation models usage, credits to [1]

[1] https://docs.anyscale.com/examples/fine-tune-stable-diffusion/

Fine-tuning Foundation models?

Foundation models are getting powerful

- general-purpose representation that can be adapted to downstream tasks
- updating the pre-trained model's parameters

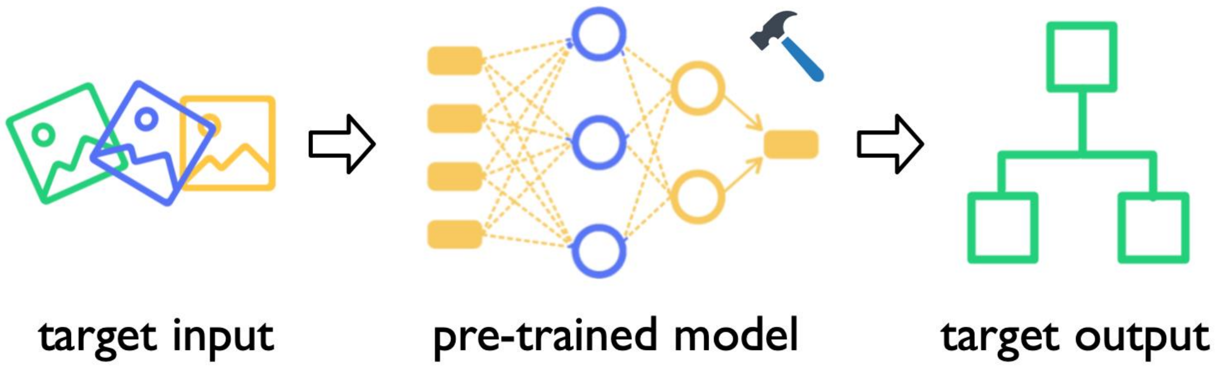


Fig. retrain or fine-tune pre-trained models

Parameter-Efficient Fine-tuning (PEFT)

Not as easy as we expected …

- # parameters reaching billion-level
- Fully fine-tuning a Foundation model is
  - costly, data hungry, and time consuming
  - A complexity perspective
    - fully fine tuning $\mathcal{O}\left(\sum_{l}^{L} d_l^2\right)$
    - fine-tune just last layer $\mathcal{O}\left(d_{L-1} \times d_L\right)$

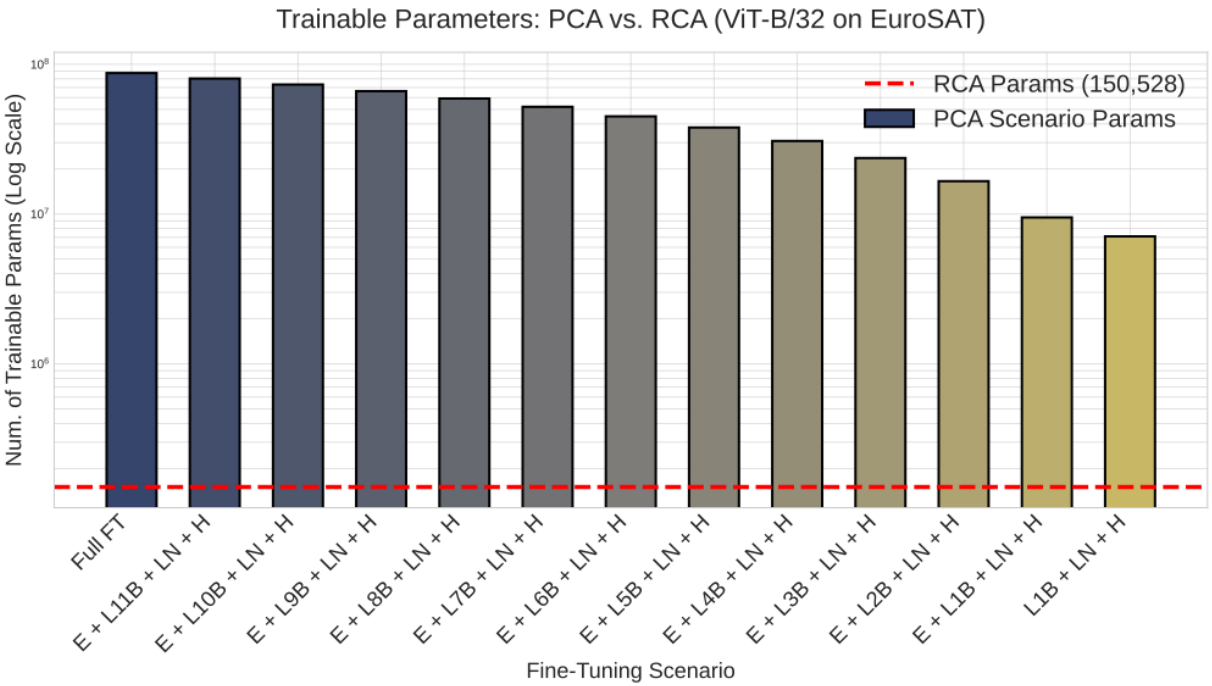


Fig. trainable parameters of fine-tuning a ViT-B/32 [1]

[1] Ye et al. Neural Network Reprogrammability: A Unified Theme on Model Reprogramming, Prompt Tuning, and Prompt Instruction. In ArXiv 2025

PEFT: Paradigm Shift

Parameter-Centric => *Reprogrammability*-Centric

- What is “reprogrammability”?
  - model can be *repurposed* in multiple scenarios
  - details to be discovered later on
- Thought transformation
  - (a) Fine-tuning: modify model to align with target task
  - (b) Reprogram: modify target task to align with model
- What to discuss today?
  - *not* a new concept, active research topics around VLMs and LLMs
  - yet fragmented themes across communities (ML, CV, NLP):
  - model reprogramming, prompt tuning, in-context learning, etc.

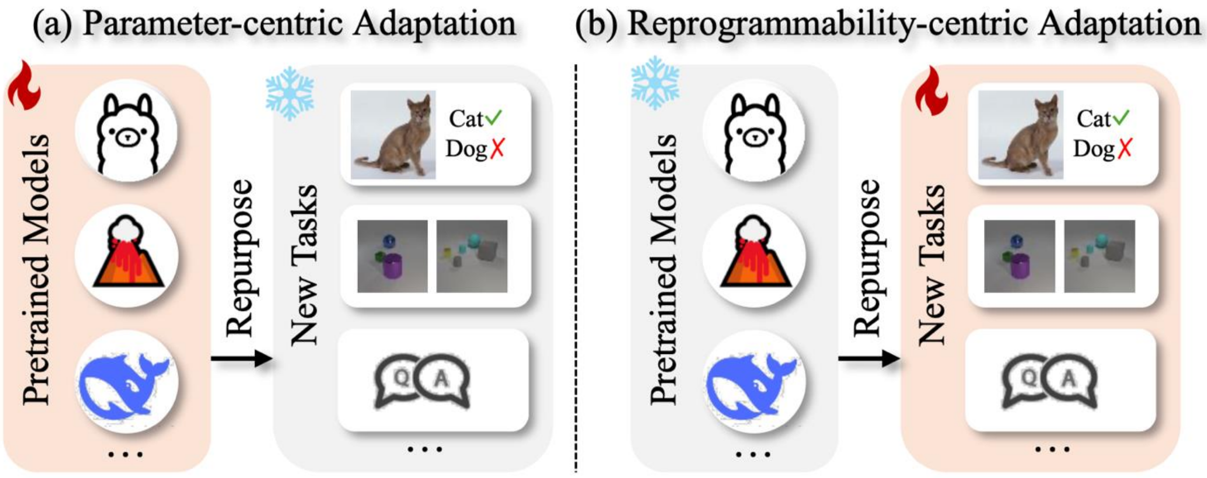


Fig. paradigm shift from adapting model to adapting tasks

***Connect** these concepts together*

Outline

I. Neural Network Reprogrammability (NNR)

II. Interpret ICL, CoT with NNR

III. Useful Resources

PART 1: Neural Network Reprogrammability

Neural Network Reprogrammability: Motivation

Concepts originates from Adversarial Attack [1]

- Neural networks are sensitive
  - overconfident, non-continuous decision boundary, etc.
- Even negligible perturbations can mislead model
- Can be exploited to hinder model performances ☹

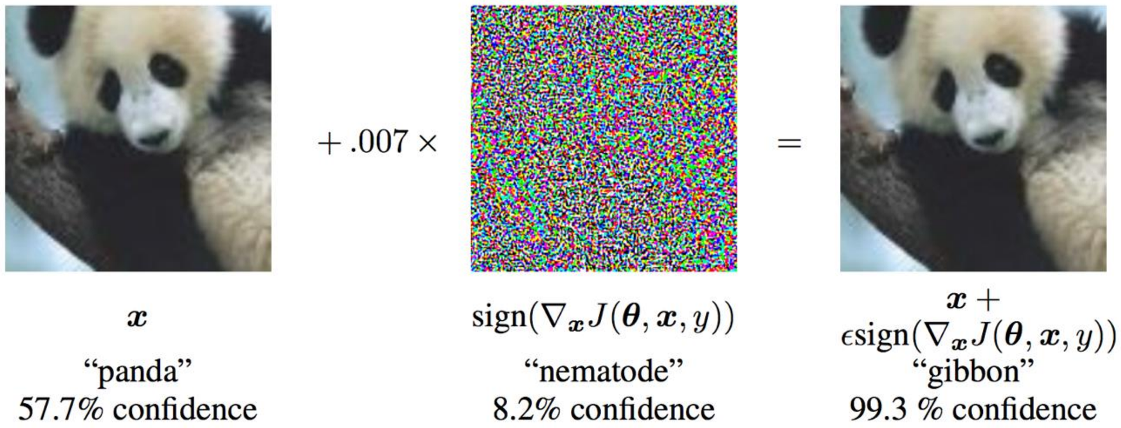


Fig. Adversarial Example and Attack

[1] Szegedy et al. Intriguing properties of neural networks. In ICLR 2014

NNR: Motivation

Can we make use of this sensitivity?

- Perturbation can also *guide* model behavior
- Aim to *perform* new downstream tasks ☺
- Known as Model (Adversarial) Reprogramming [1]

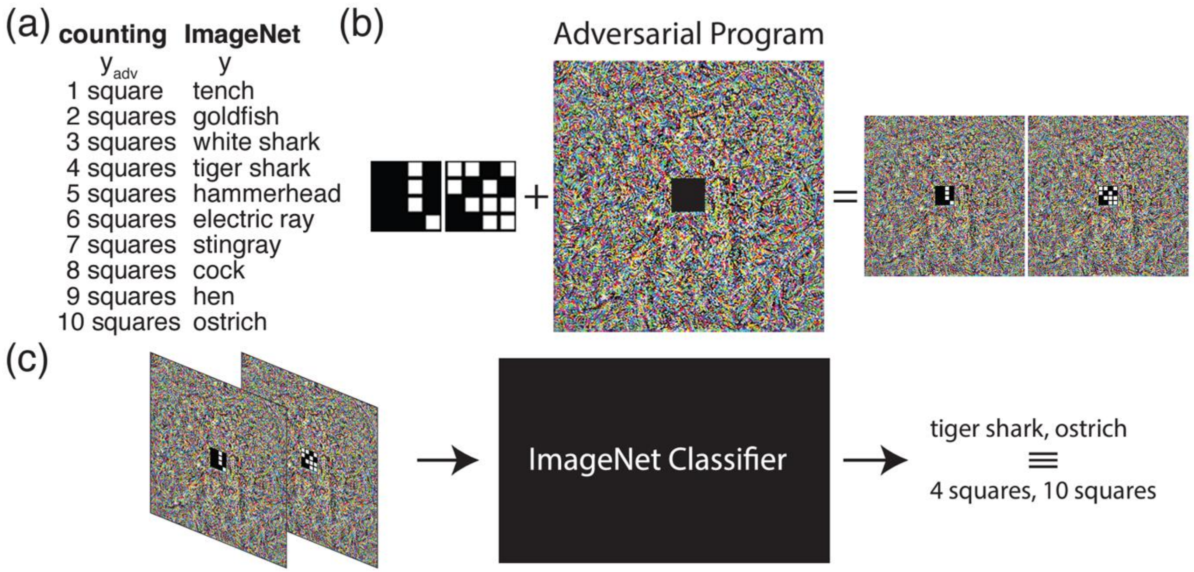
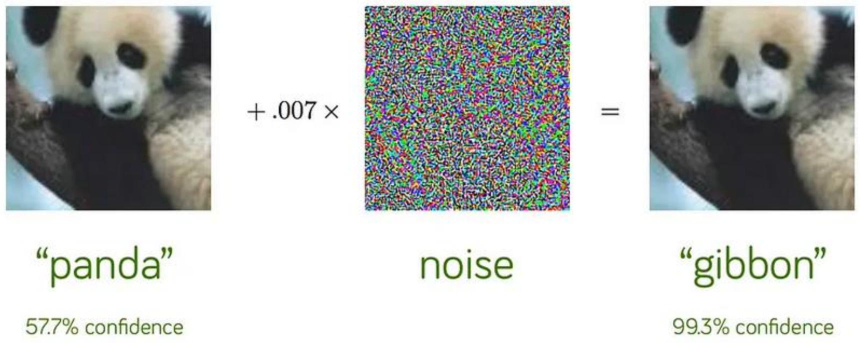


Fig. Illustration of Input (Adversarial) Reprogramming, credits to [1]

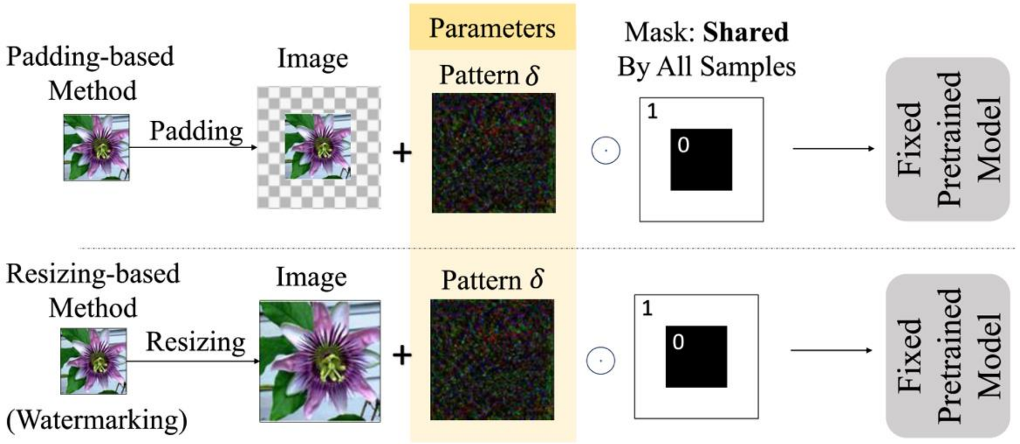
[1] Elsayed et al. Adversarial Reprogramming of Neural Networks. In ICLR 2019

NNR: Motivation

Takeaway



VS



Attack: perturbation to *hinder* a model

Reprogramming: perturbation to *reuse* a model [1, 2]

[1] Elsayed et al. Adversarial Reprogramming of Neural Networks. In ICLR 2019
[2] Cai et al. Sample-specific Masks for Visual Reprogramming-based Prompting. In ICML 2024

NNR: Components

Takeaway

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

NNR: Definition

Formally

Let $f(x^S; \theta)$ be a model pre-trained on a source domain $D^S \subseteq X^S \times Y^S$, where $f: X^S \rightarrow Y^S$.

Say ***N****eural* ***N****etwork* ***R****eprogrammability* (NNR) *as* $f(x^S; \theta)$ that achieves a **target functionality**, defined over $X^T \times Y^T$, with two *configurable* mappings:

- Input manipulation $I: X^T \rightarrow X^S$
- Output alignment $O: Y^S \rightarrow Y^T$

NNR: Definition

Formally

Let $f(x^S;\theta)$ be a model pre-trained on a source domain $D^S \subseteq X^S \times Y^S$, where $f: X^S \to Y^S$.

Say ***N**eural **N**etwork **R**eprogrammability* (NNR) *as* $f(x^S;\theta)$ that achieves a **target functionality**, defined over $X^T \times Y^T$, with two *configurable* mappings:

- Input manipulation $I: X^T \to X^S$
- Output alignment $O: Y^S \to Y^T$

$$\hat{y}^T = O\left(\underbrace{f\left(\underbrace{I\left(x^T\right)}_{\tilde{x}^S}\right)}_{y^S}\right)$$

NNR in textual modality

Revisit Text Prompting under NNR

- Foundation *model* (e.g., VLM) [1] can do zero-shot prediction
- target task: $D^T \neq D^S$ with target data $\{(x^T, y^T)\}$
  - **What?** Text prompting are input manipulations
  - **How?** manipulating *input* by prompting w/ "This is a photo of [y^T]", $\forall\, y^T \in Y^T$
  - **Why?** *task-specific* hint to guide VLM's behavior
- Can input manipulation be other forms?

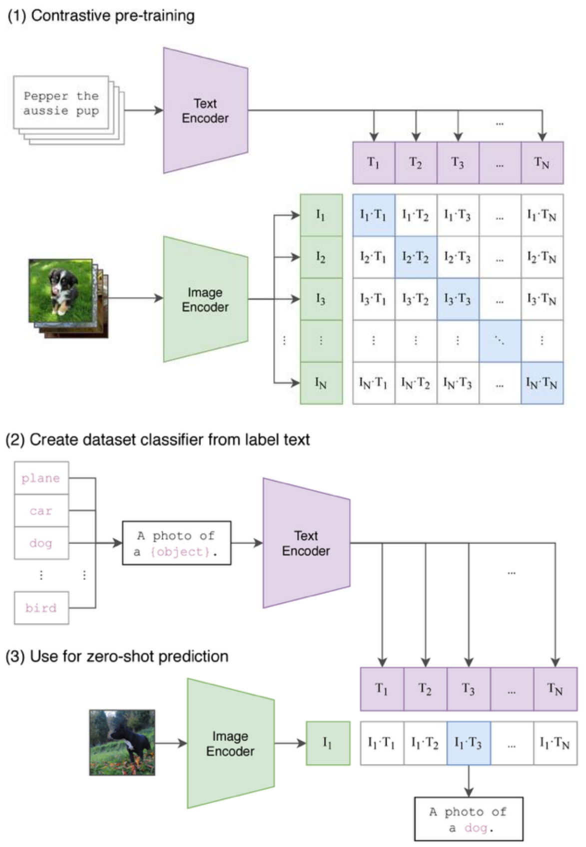


Fig. prompting a pre-trained VLM, credits to [1]

[1] Radford et al. Learning Transferable Visual Models From Natural Language Supervision. In ICML 2021

NNR in textual modality

Revisit Text Prompting under NNR

- Foundation *model* (e.g., VLM) [1] can do zero-shot prediction
- target task: $D^T \neq D^S$ with target data $\{(x^T, y^T)\}$
  - **What?** Text prompting are input manipulations
  - **How?** manipulating *input* by <u>prompting</u> w/ “This is a photo of $[y^T]$”, $\forall\, y^T \in Y^T$
  - **Why?** *task-specific* <u>hint</u> to guide VLM’s behavior; can take even other forms
- Prompting formats do matter for task performance
  - Is “This is a photo of $[y^T]$” always an appropriate prompt? – **No**
  - Yet impractical to manually design for every, unknown task

Caltech101

Prompt	Accuracy
a [CLASS].	82.68
a photo of [CLASS].	80.81
a photo of a [CLASS].	86.29

EuroSAT

Prompt	Accuracy
a photo of a [CLASS].	24.17
a satellite photo of [CLASS].	37.46
a centered satellite photo of [CLASS].	37.56

Fig. different data may need different prompts, credits to [1]

[1] Zhou et al. Learning to Prompt for Vision-Language Models. In IJCV 2022

NNR in textual modality

Revisit Text Prompting under NNR

- Foundation *model* (e.g., VLM) [1] can do zero-shot prediction
- target task: $D^T \neq D^S$ with target data $\{(x^T, y^T)\}$
  - **What?** Text prompting are input manipulations
  - **How?** manipulating *input* by <u>prompting</u> w/ "This is a photo of $[y^T]$", $\forall\, \mathrm{y}^{\mathrm{T}} \in Y^T$
  - **Why?** *task-specific* <u>hint</u> to guide VLM's behavior; can take even other forms
- Prompting formats do matter for task performance
  - Is "This is a photo of $[y^T]$" always an appropriate prompt? – **No**
  - Yet impractical to manually design for every, unknown task
  - Optimize a soft prompt instead, <u>prompting</u> w/ $[V]_1[V]_2\ldots[V]_M\,[\mathrm{CLS}]$.

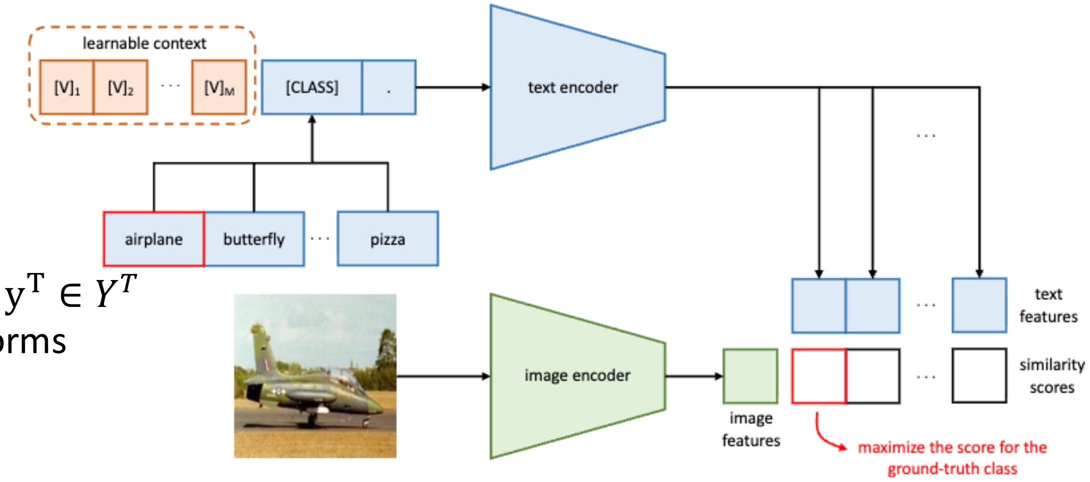


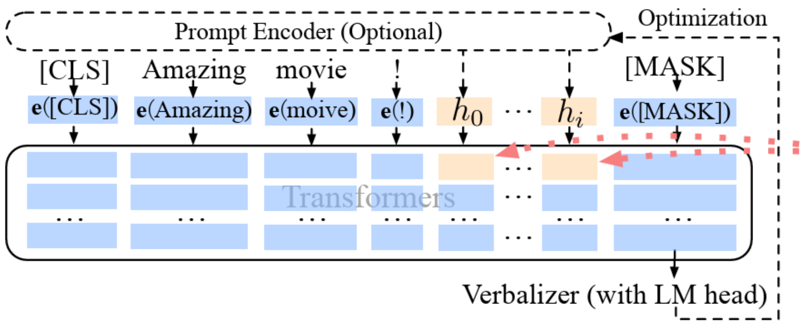
Fig. learning to prompt a pre-trained VLM, credits to [1]

[1] Zhou et al. Learning to Prompt for Vision-Language Models. In IJCV 2022

NNR in textual modality

Revisit Text Prompting under NNR

- Foundation *model* (e.g., VLM) [1] can do zero-shot prediction
- target task: $D^T \neq D^S$ with target data $\{(x^T, y^T)\}$
  - **What?** Text prompting are input manipulations
  - **How?** manipulating *input* by prompting w/ "This is a photo of [y^T]", $\forall\, y^T \in Y^T$
  - **Why?** *task-specific* hint to guide VLM's behavior; can take even other forms
- where to place prompts can be flexibly chosen
  - Early works [1-3] prompt w/ $[V]_1[V]_2 \ldots [V]_M\ [\mathrm{CLS}].$ as token embeddings
  - Often possible to prompt at multiple different locations, e.g., hidden layers [4]



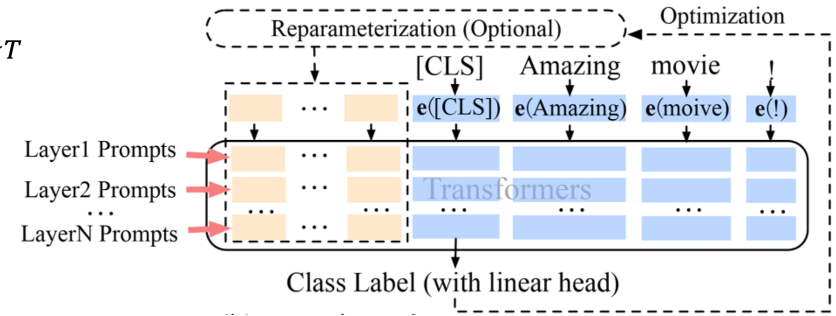


Fig. (u) Placing prompts at embedding layer
(d) Placing prompts at hidden layers [4]

[1] Zhou et al. Learning to Prompt for Vision-Language Models. In IJCV 2022
[2] Lester et al. The Power of Scale for Parameter-Efficient Prompt Tuning. In EMNLP 2021
[3] Liu et al. P-Tuning: Prompt Tuning Can Be Comparablew to Fine-tuning Across Scales and Tasks. In ACL 2022
[4] Liu et al. P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks. In ArXiv 2021

NNR in textual modality

In-Context Learning under NNR

- Emergent capabilities of LLMs [1]
- In-context learning leverages a set of demonstrations of target task
  - **What?** demonstrations $\{(x_i^T, y_i^T)\}_i$ are input manipulations
  - **How?** manipulating *input* x_*^T by concatenating $\{(x_i^T, y_i^T)\}_i$ as new input

pre-trained LLM can then predict y_*^T
does **NOT** need explicit learning procedure

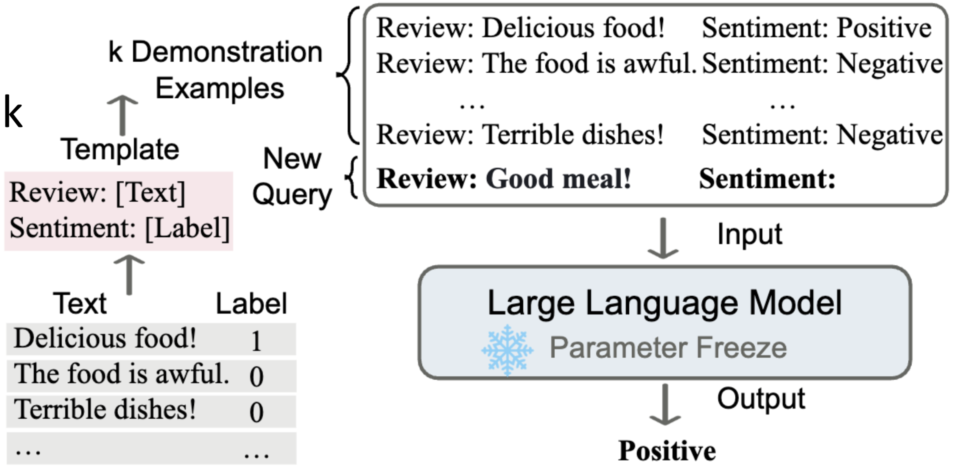


Fig. example of ICL pipeline [1]

[1] Wei et al. Emergent Abilities of Large Language Models. In TMLR 2022

NNR in textual modality

Chain-of-Thought Prompting under NNR

- Emergent capabilities of LLMs [1]
- Chain-of-thought leverages intermediate "thinking" steps
  - **What?** "think step by step" leads to input manipulations
  - **How?** manipulating query input x_*^T by concatenating reasoning st

predict y_*^T with In-Context Reasoning of x_*^T
sample-specific input manipulations

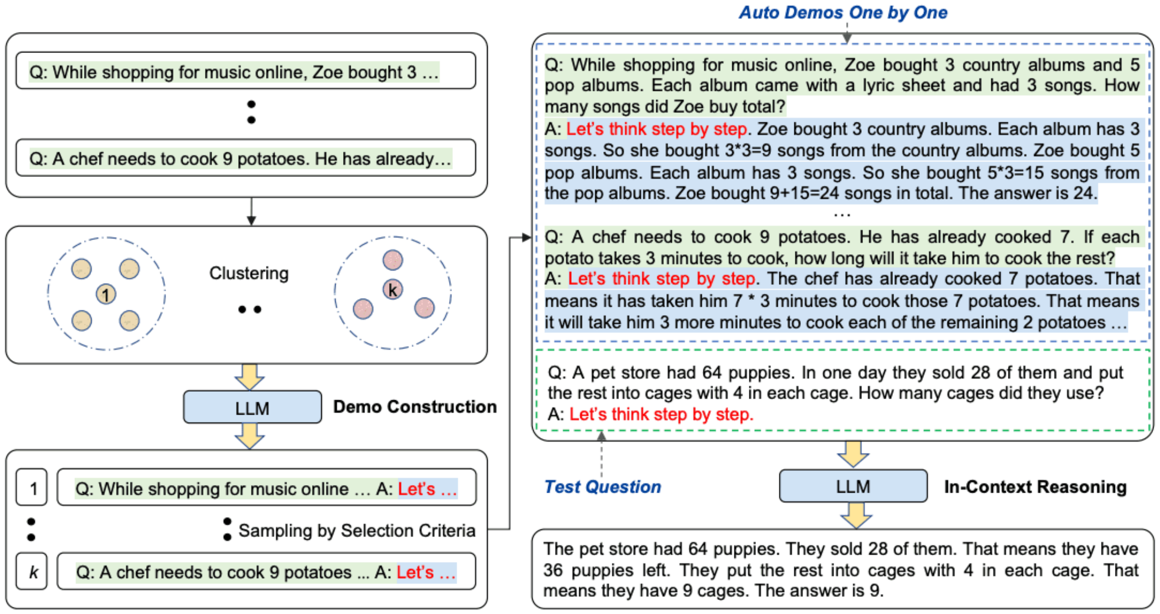


Fig. example of CoT pipeline [2]

[1] Wei et al. Emergent Abilities of Large Language Models. In TMLR 2022
[2] Wei et al. Chain-of-Thoughts Prompting Elicits Reasoning in Large Language Models. In NeurIPS 2022

NNR in textual modality

Broadly, all text promptings are NNR manifestations

- Guide pre-trained *foundation model*
  - **What?** Text prompting are input manipulations
  - **How?** manipulating *input* by prompting w/ *textual* elements
  - **Why?** *task-specific* hint to guide pre-trained model's behavior

- Different *formats* of text prompting as input manipulation
  - Hard (readable) prompting
    - in-context learning [2], chain-of-thought [3], etc.
  - Soft (unreadable) prompting
    - prompt tuning [4], etc.

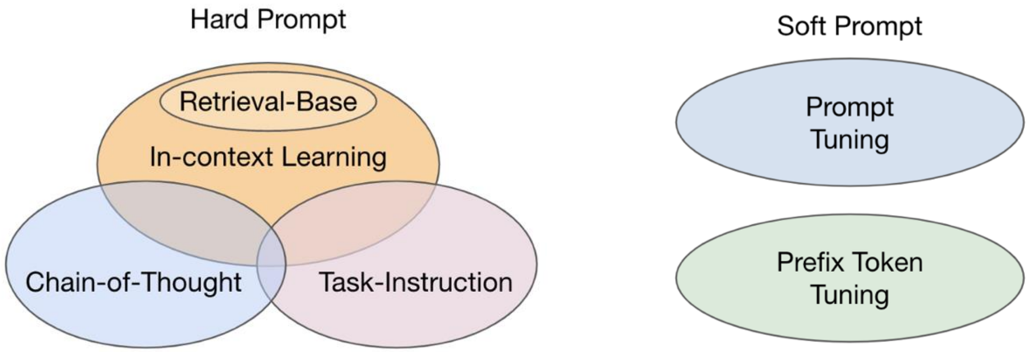


Fig. a categorization of IM for text data, credits to [1]

[1] Wu et al. A Systematic Survey of Prompt Engineering on Vision-Language Foundation Models. In ArXiV 2023
[2] Min et al. What Makes In-Context Learning Work. In ACL 2022
[3] Wei et al. Chain-of-thought prompting elicits reasoning in large language models. In NeurIPS 2022
[4] Zhou et al. Conditional prompt learning for vision-language models. In CVPR 2022
[5] Zhou et al. Learning to Prompt for Vision-Language Models. In IJCV 2022

NNR in visual modality

All the things remain the same in visual modality

- Guide pre-trained vision/vision-language model
  - **What?** visual prompting as input manipulation
  - **How?** manipulating *input* by <u>prompting</u> w/ *visual* elements
  - **Why?** provide task-specific hint about how to handle visual data

Can take various formats as well

- Segment-Anything (SAM) [1] as input manipulation
  - Hard prompting based on visible annotations
    - points, bounding-box, markers, etc.

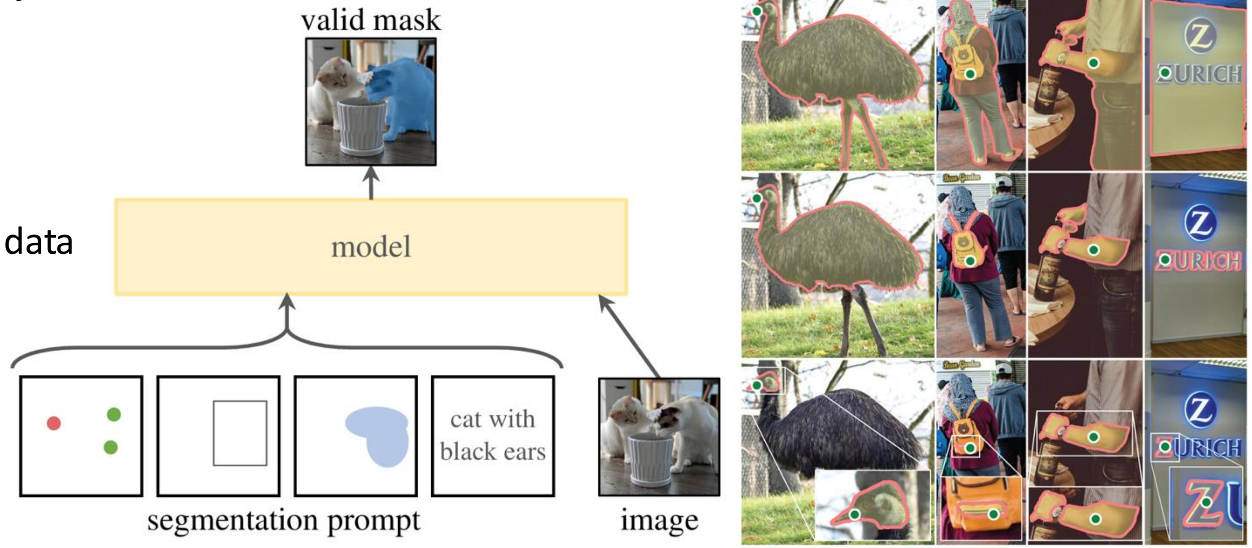


Fig. (l) zero-shot segmentation enabled by different prompts
(r) point prompts leads to generated masks over objects [1]

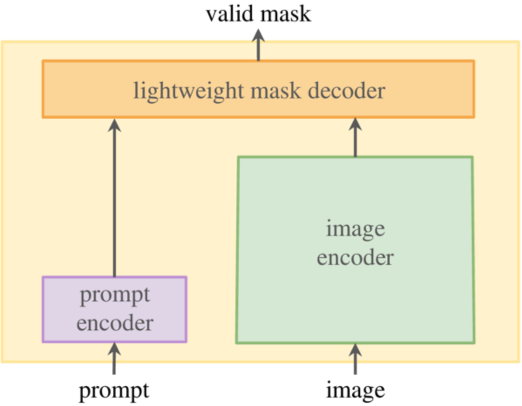
[1] Kirillov et al. Segment Anything. In ICCV 2023.

NNR in visual modality

All the things remain the same in visual modality

- Guide pre-trained vision/vision-language model
  - **What?** visual prompting as input manipulation
  - **How?** manipulating *input* by <u>prompting</u> w/ *visual* elements
  - **Why?** provide task-specific hint about how to handle visual data

- Segment-Anything (SAM) [1] as input manipulation
  - Hard prompting based on visible annotations
    - points, bounding-box, markers, etc.
  - Handled as <u>additional tokens</u> along with image *embeddings*
    - transfer zero-shot to new tasks



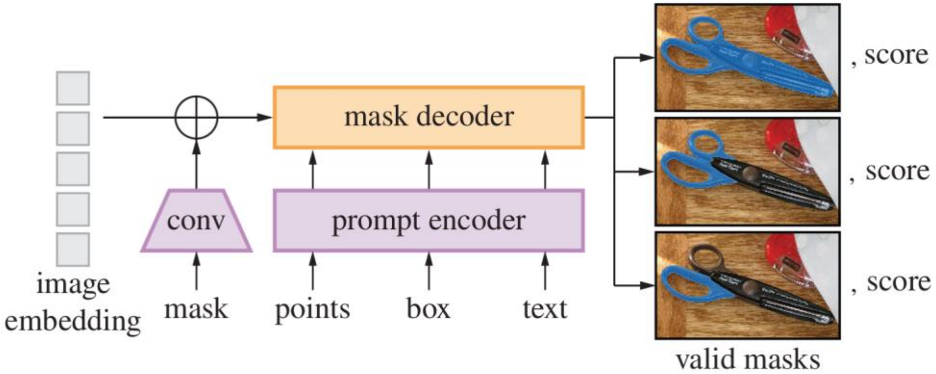


Fig. how visual prompts are handled to predict task-specific output [1]

[1] Kirillov et al. Segment Anything. In ICCV 2023.

NNR in visual modality

All the things remain the same in visual modality

- Guide pre-trained vision/vision-language model
  - **What?** visual prompting as input manipulation
  - **How?** manipulating *input* by prompting w/ *visual* elements
  - **Why?** provide task-specific hint about how to handle visual data

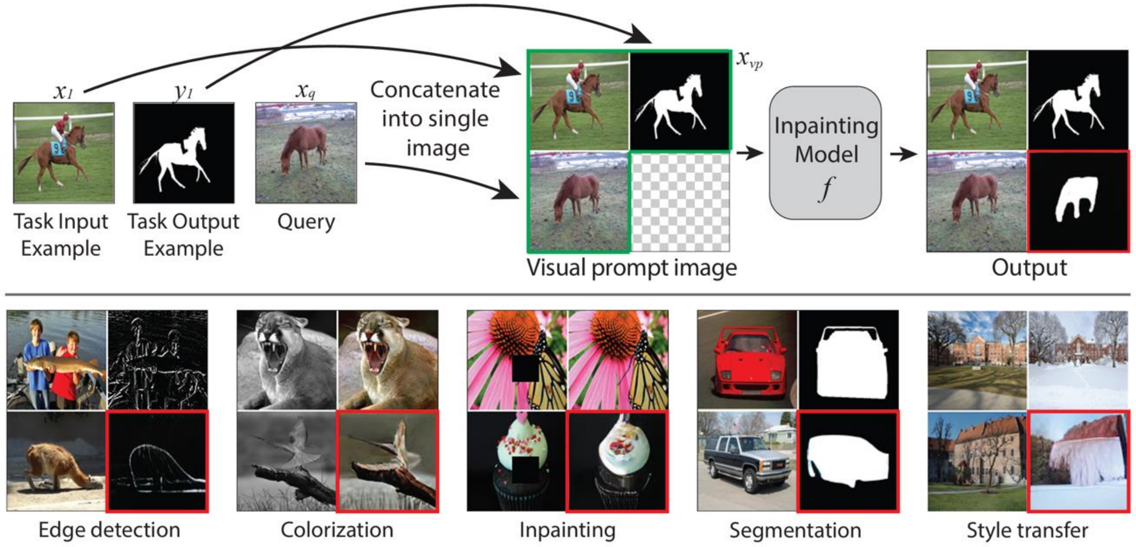


Fig. how visual prompts are handled to predict task-specific output [1]

- Visual Prompting [1] as input manipulation
  - Inpainting model pre-trained on large-scale dataset
    - Random masking, learning to inpaint masked regions
  - Prompting x_q with few-shot task-specific demonstrations
    - Manipulating by building "grid"-like target input and mask
    - Leveraging *frozen* pre-trained model to restore the mask
  - Training-free; can adapt to multiple target tasks (e.g., segmentation, edge detection), provided with proper task output examples

[1] Bar et al. Visual Prompting via Image Inpainting. In NeurIPS 2022.

NNR in visual modality

All the things remain the same in visual modality

- Guide pre-trained vision/vision-language model
  - **What?** visual prompting as input manipulation
  - **How?** manipulating *input* by <u>prompting</u> w/ *visual* elements
  - **Why?** provide task-specific hint about how to handle visual data

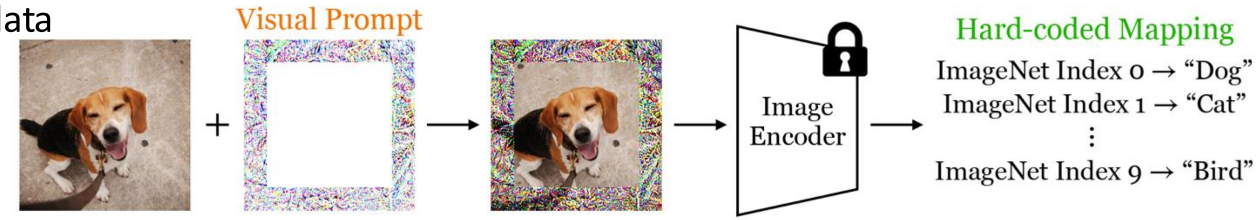


Fig. how visual prompts are handled to predict task-specific output [1]

- Visual Prompting [1] as input manipulation
  - Soft prompting based on unreadable elements
    - noises, etc.
  - Handled as <u>additional pixels</u> along with raw image pixels
    - necessitates optimization with respect to task-specific loss functions

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

Input Manipulation w/ Sample specificity

Sample-specific masks for IM

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

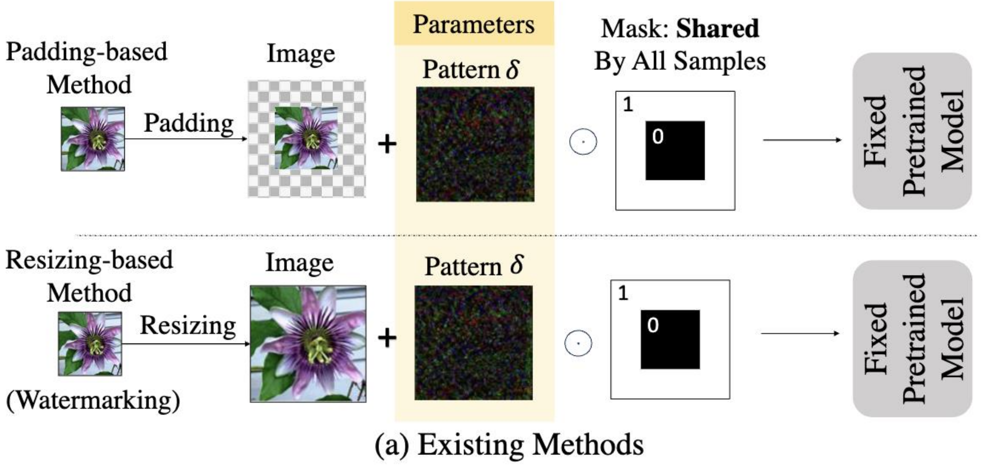


Fig. a shared mask is used across downstream samples [1]

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

Input Manipulation w/ Sample specificity

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

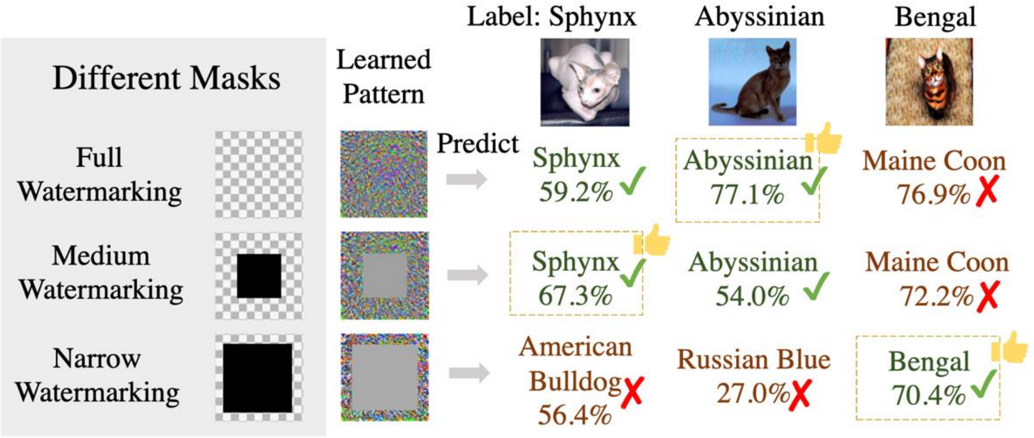


Fig. different samples benefit from different masks [1]

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

Input Manipulation w/ Sample specificity

Sample-specific masks for IM

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

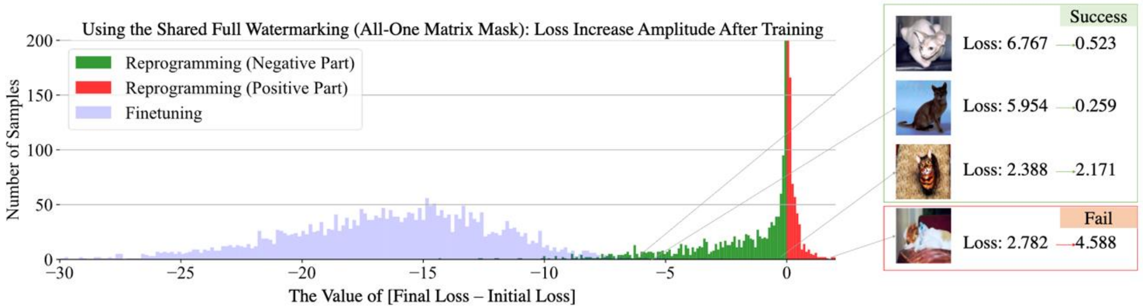


Fig. shared mask leads to loss increase for certain samples [1]

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

Input Manipulation w/ Sample specificity

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

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

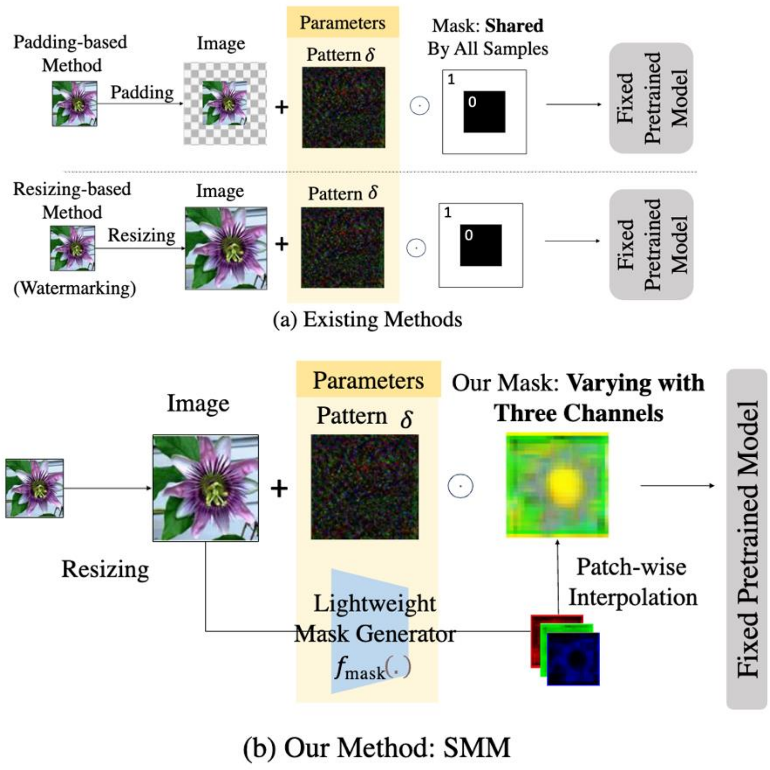


Fig. SMM: generate a mask for each image [1]

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

NNR in visual modality

Broadly, all visual promptings are (also) NNR manifestations

- Guide pre-trained vision/vision-language model
  - **What?** visual prompting as input manipulation
  - **How?** manipulating *input* by <u>prompting</u> w/ *visual* elements
  - **Why?** provide task-specific hint

Can take various formats as well

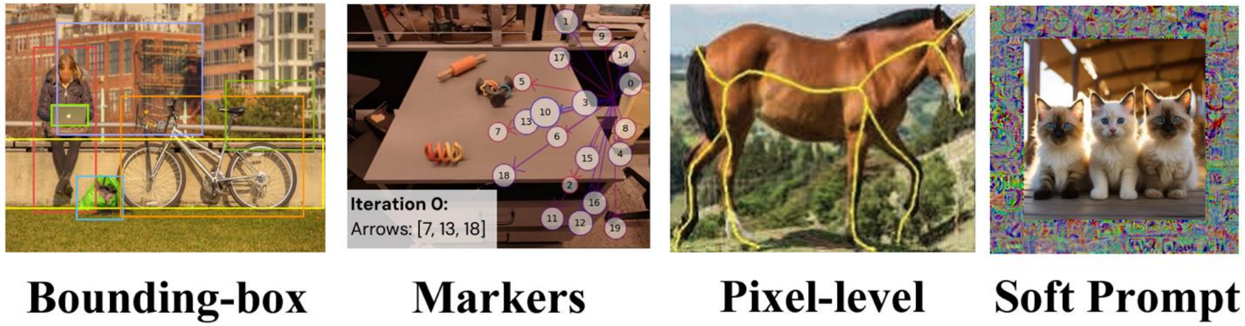


Fig. a categorization of visual prompting, credits to [1]

- Different *formats* of text prompting as input manipulation
  - Hard prompting based on readable annotations
    - bounding-box, markers [2], etc.
  - Soft prompting based on unreadable elements
    - prompt tuning, e.g., trainable noises added to the image [3]

[1] Wu et al. Visual Prompting in Multimodal Large Language Models: A Survey. In ArXiV 2024
[2] Kirillov et al. Segment Anything. In ICCV 2023.
[3] Jia et al. Visual Prompt Tuning. In ECCV 2022

NNR in multi-modality

In multi-modal contexts, textual and visual prompt tuning follow the same idea

- Both “search” for optimal prompts with gradient-descent
  - *learnable* tokens prepended to text, e.g.,

    <Prompt> a puppy

  - *learnable* perturbation added to the image, e.g.,

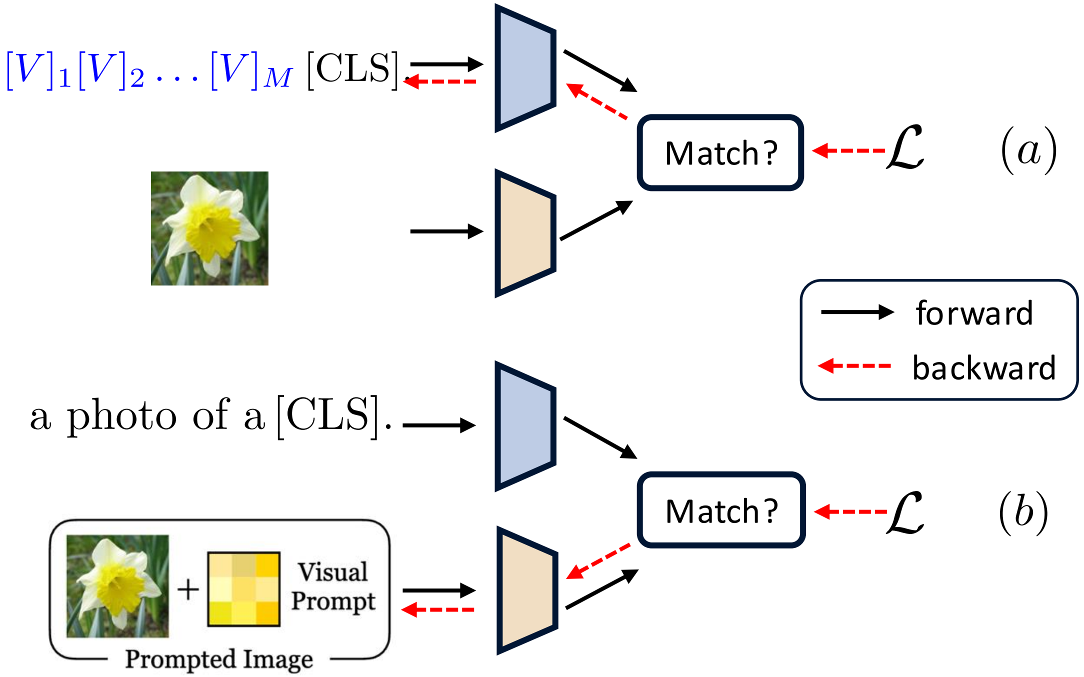


Fig. prompt tuning on (a) text OR (b) image with vision-language models

NNR in multi-modality

In multi-modal contexts, textual and visual prompt tuning follow the same idea

- Both “search” for optimal prompts with gradient-descent
  - *learnable* tokens prepended to text, e.g.,
  - *learnable* perturbation added to the image, e.g.,

- Can be tuning a prompt for a single modality
  - i.e., uni-modal prompting
- Otherwise, tuning prompts for both modalities
  - i.e., multi-modal prompting [1]

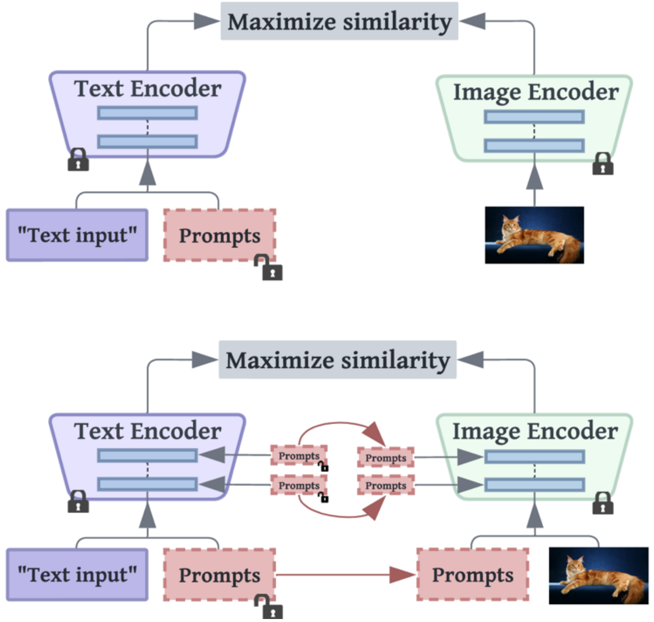


Fig. prompt tuning on (a) text OR/AND (b) image with vision-language models

[1] Khattak et al. MaPLe: Multi-modal Prompt Learning. In CVPR 2023.

NNR in multi-modality

In multi-modal contexts, textual and visual prompt tuning follow the same idea

- Both “search” for optimal prompts with gradient-descent
  - *learnable* tokens prepended to text, e.g.,
  - *learnable* perturbation added to the image, e.g.,
- An example of tuning prompts for both modalities [1]
  - Prompts can be added to multiple places

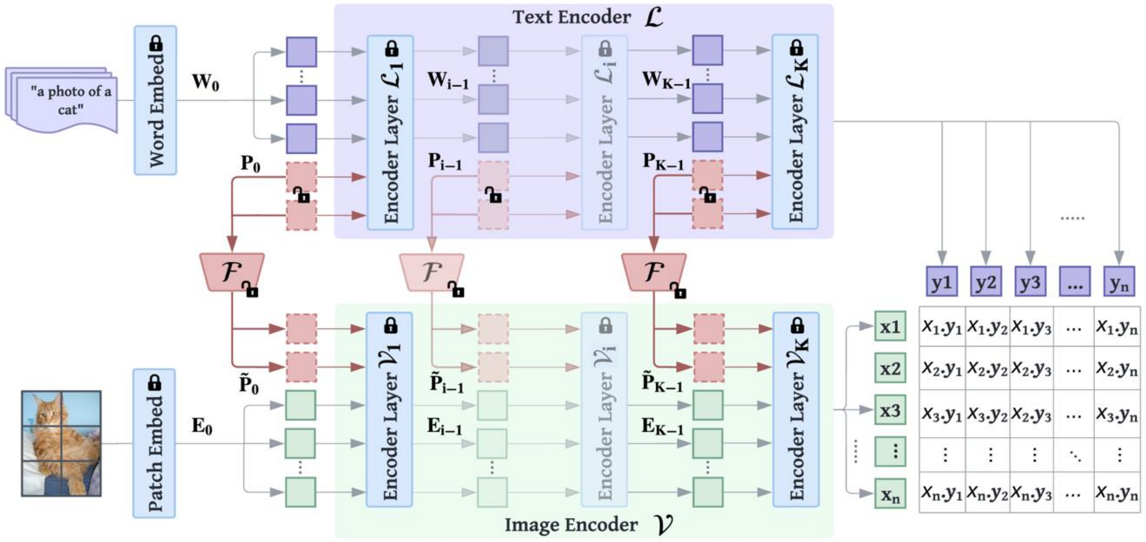


Fig. prompt tuning on (a) text AND (b) image with vision-language models [1]

[1] Khattak et al. MaPLe: Multi-modal Prompt Learning. In CVPR 2023.

NNR in cross-modality

NNR can be manifested even across modalities

- Guide pre-trained acoustic model for (*numeric*) time-series data
  - **What**?
    - Pre-trained AM can be repurposed to handle data from another modality
  - **How?**
    - manipulating *input* by <u>prompting</u> w/ trainable segments
    - aligning *output* by <u>projection</u> w/ hard-coded mappings
  - **Why?**
    - transform time-series data into tokens that frozen AM can handle
    - map from acoustic label space to time-series label space

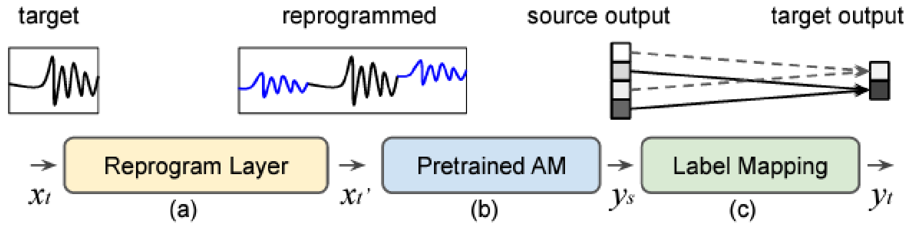


Fig. Framework of Voice2Series [1]

- Risk analysis

$$\underbrace{\mathbb{E}_{\mathcal{D}_\mathcal{T}}[\ell_\mathcal{T}(\widetilde{x}_t(\theta^*), y_t)]}_{\text{target risk}} \leq \underbrace{\epsilon_\mathcal{S}}_{\text{source risk}} + 2\sqrt{K}\cdot \underbrace{\mathcal{W}_1(\mu(z_\mathcal{S}(\widetilde{x}_t(\theta^*))), \mu(z_\mathcal{S}(x_s)))_{x_t\sim\mathcal{D}_\mathcal{T},\, x_s\sim\mathcal{D}_\mathcal{S}}}_{\text{representation alignment loss via reprogramming}},$$

[1] Yang et al. Voice2Series: Reprogramming Acoustic Models for Time Series Classification. In ICML 2021

NNR in cross-modality

NNR can be manifested even across modalities

- Guide pre-trained language model for protein sequences
  - **What**?
    - Pre-trained LM can be repurposed to handle data from irrelevant modality
  - **How?**
    - manipulating *input* by <u>prompting</u> w/ trainable linear projection
    - aligning *output* by <u>projection</u> w/ trainable linear projection
  - **Why?**
    - transform Antibody data into tokens that English-pretrained LM [2] can handle
    - transform English word embeddings back into Antibody

- commendable performance than baselines
  - high diversity, good sequence recovery, and low perplexity

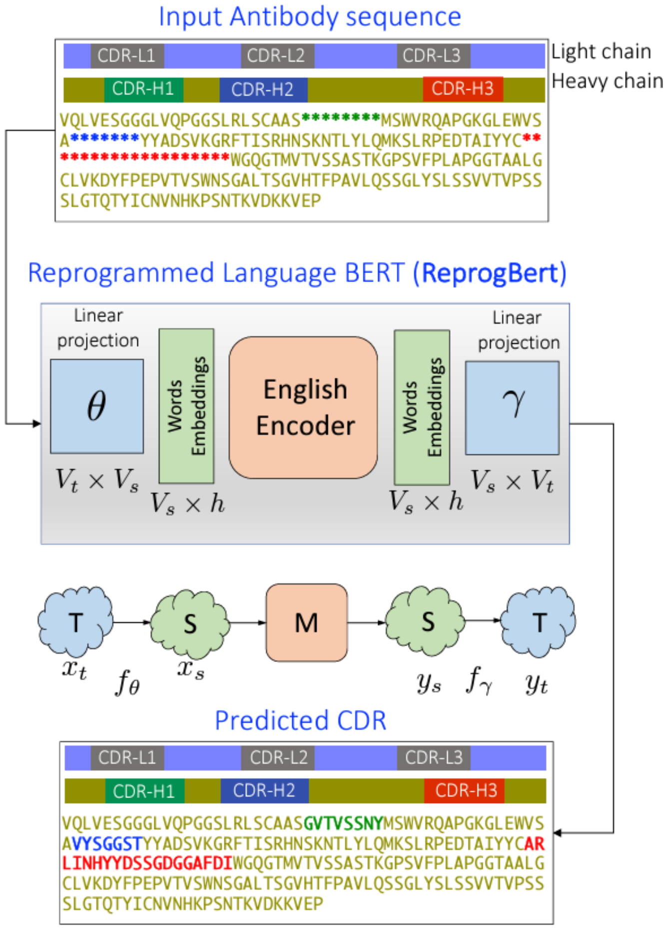


Fig. Framework of protein sequence infilling [1]

[1] Melnyk et al. Reprogramming language model for antibody sequence infilling. In ICML 2023
[2] Devlin et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In NAACL 2019

NNR in cross-modality

NNR can be manifested even across modalities

- Guide pre-trained LLM for (*numeric*) time-series data
  - **What**?
    - LLMs can be repurposed to handle data from another modali
  - **How?**
    - manipulating *input* by <u>prompting</u> w/ trainable tokens
    - aligning *output* by <u>projection</u> w/ trainable layers
  - **Why?**
    - encode numeric data into tokens that LLMs can recognize
    - decode textual output back to numeric values

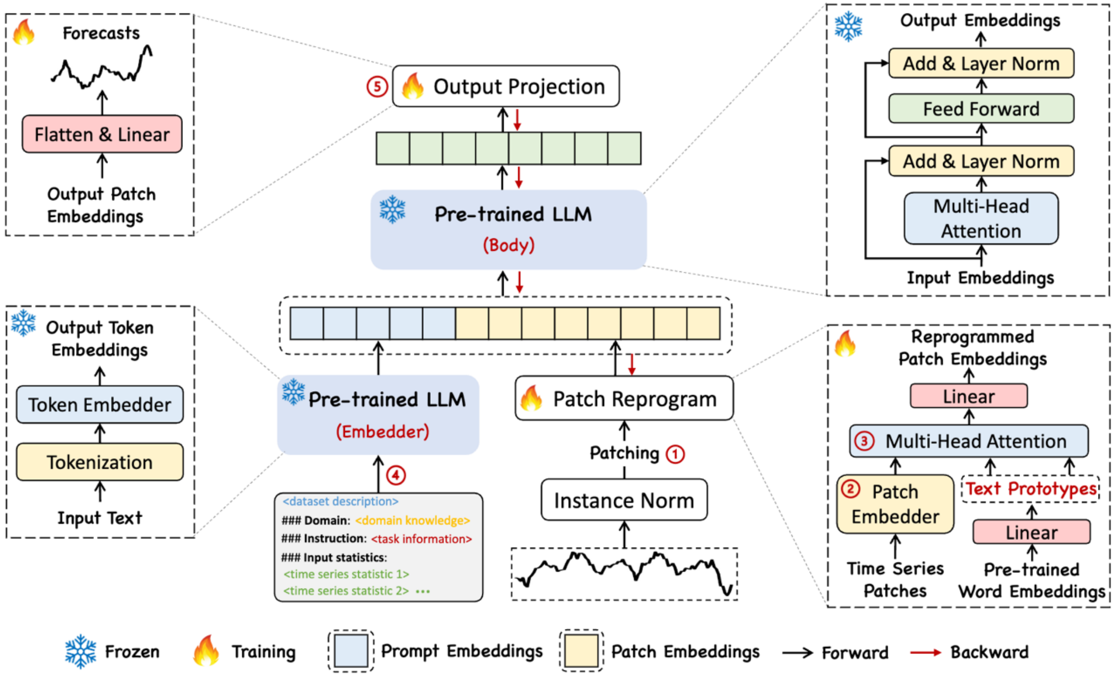


Fig. an example of NNR for time-series, credits to [1]

[1] Jin et al. TIME-LLM: Time Series Forecasting by Reprogramming Large Language Models. In ICLR 2024

NNR in cross-modality

NNR can be manifested regardless of domains / modalities

benefit **low-resource** domains,
where training from scratch is difficult

In-domain adaptation		Cross-domain adaptation	
Source	**Target**	**Source**	**Target**
general image	domain-specific image	image	financial transaction
text	word level task	text	time-series
speech	low-resource speech	text	protein

Fig. diverse application scenarios of NNR [1]

[1] Pin-Yu Chen. Model Reprogramming: Resource-Efficient Cross-Domain Machine Learning. In AAAI 2024

NNR as a unifying umbrella

NNR is a *general* idea, not limited by modality, as well as:

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

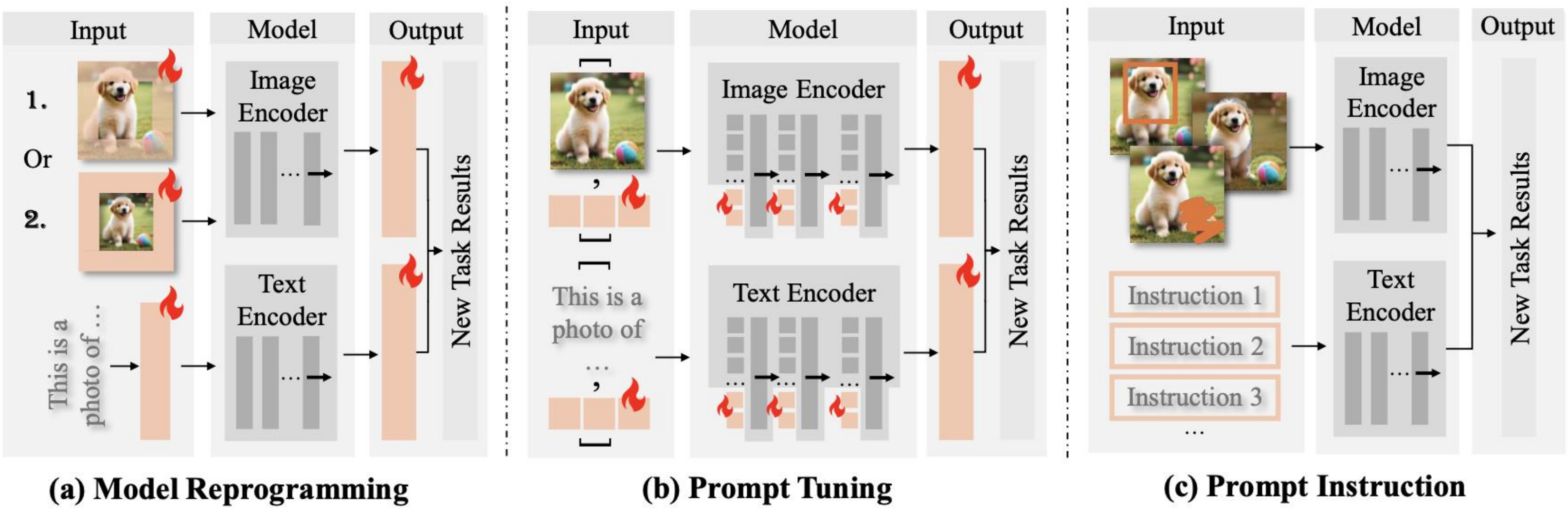


Fig. NNR manifests in different ways across different PEFT methodologies, credits to [1]

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

PART II: ICL and CoT under NNR

In-Context Learning under NNR

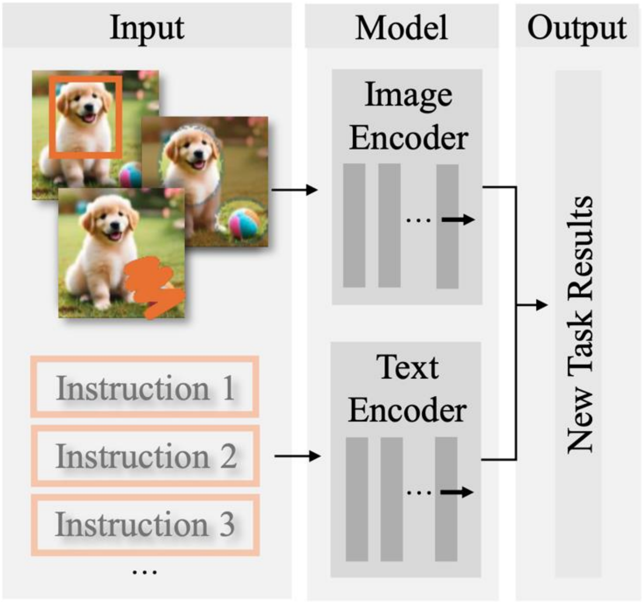


Fig. concept of ICL pipeline [1]

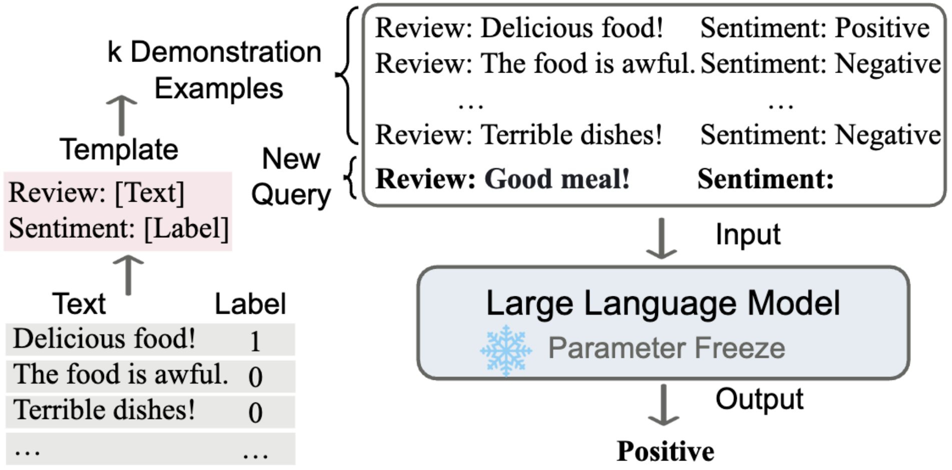


Fig. example of ICL pipeline [2]

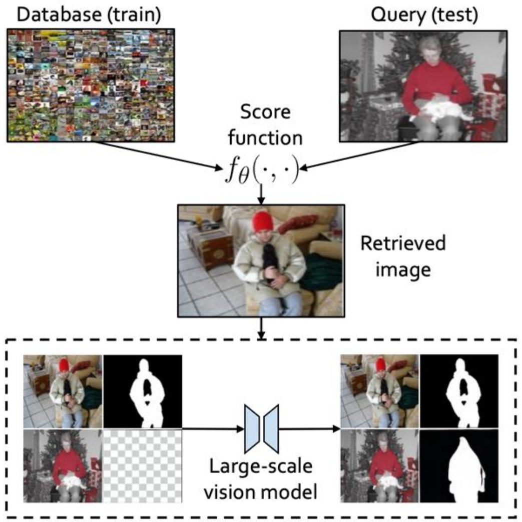


Fig. example of ICL pipeline [3]

- Format: Fixed, non-learnable manipulation
- Location: exclusively at model input space
- Operator: concatenative (text) / additive (visual)
- Output: minimal, mostly implicit identity/rule-based OA

[1] Ye et al. Neural Network Reprogrammability: A Unified Theme on Model Reprogramming, Prompt Tuning, and Prompt Instruction. To appear.
[2] Dong et al. A survey of In-Context Learning. In ArXiV 2022
[3] Zhang et al. What makes good examples for visual in-context learning? In NeurIPS 2023

Chain-of-Thought under NNR

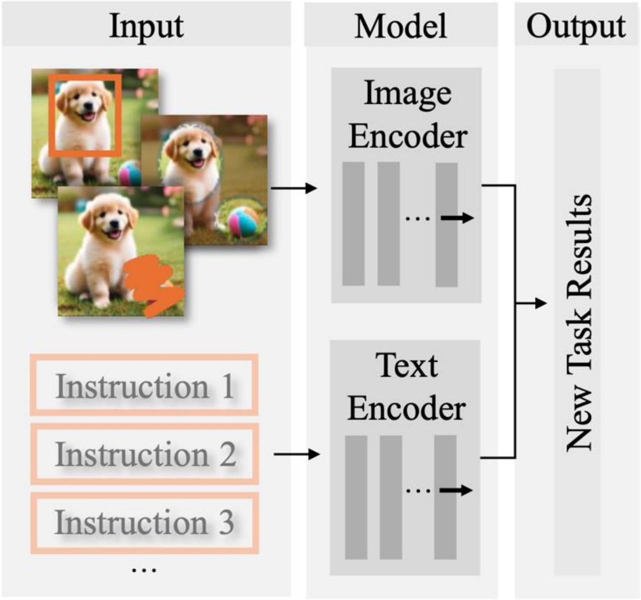


Fig. concept of ICL pipeline [1]

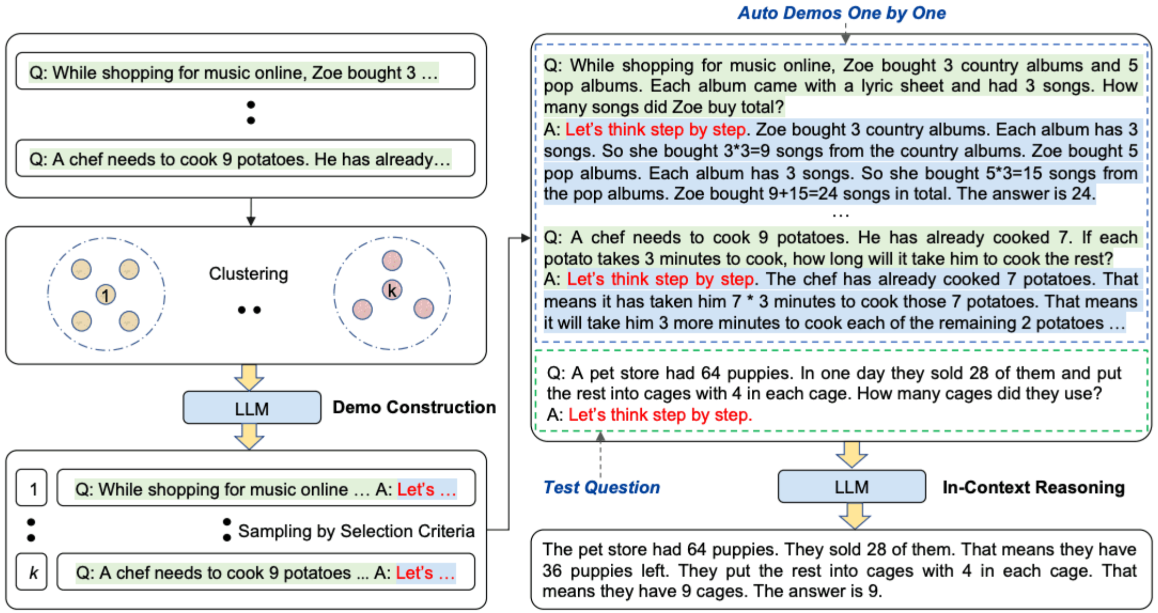


Fig. example of CoT pipeline [2]

- Format: Fixed, non-learnable manipulation
- Location: exclusively at model input space
- Operator: concatenative (text)
- Constraints: append (intermediate, sample-specific) reasoning results to manipulations

[1] Ye et al. Neural Network Reprogrammability: A Unified Theme on Model Reprogramming, Prompt Tuning, and Prompt Instruction. To appear.
[2] Wei et al. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In TMLR 2022

PART III: Useful Resources

Useful Resources

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 <u>four-dimensional</u> 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

Types of assets (will be available soon!)

- 📚 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.

Useful Resources

TMLR
TRUSTWORTHY MACHINE LEARNING AND REASONING

THE UNIVERSITY OF MELBOURNE

Reference Implementations and Summaries

- **Awesome-MR**
  - one-top information about existing model reprogramming research
- **Awesome-Reprogrammability**
  - comprehensive survey and materials about NNR
- **SMM (ICML 2024 spotlight)**
  - comprehensive (trainable) IM (with a focus on visual recognition)
- **BayesianLM (NeurIPS 2024 oral)**
  - comprehensive learning/statistics-based explicit OA
- **AttrVR (ICLR 2025)**
  - can LLMs improve (visual) input manipulation for VLM?
- **Decoupled VP (ICML 2025)**
  - can LLMs disentangle input manipulation optimization?

Check them out!

MR survey | MR-repository

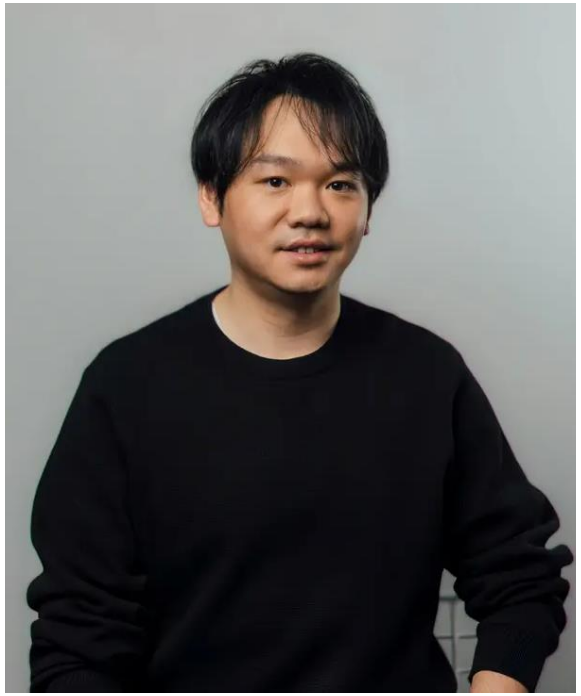
SMM | BayesianLM | AttrVR | DVP

Amazing Co-authors

Pin-Yu Chen
IBM Research AI
MIT-IBM Watson AI Lab

Jianzhong Qi
University of Melbourne

Amazing Team



Zesheng Ye,
Postdoc
AI Safety, Adaptation

Chengyi Cai,
PhD Student
Adaptation

THE UNIVERSITY OF MELBOURNE

Thank you

fengliu.ml@gmail.com

https://fengliu90.github.io/

Special thanks to Dr Zesheng Ye for preparing the most of materials.