The 40th Annual AAAI Conference on Artificial Intelligence
JANUARY 20 – JANUARY 27, 2026 | SINGAPORE

AAAI 2026 Tutorial TH20

Model Reuse in the LLM Era

Leveraging Pre-Trained Resources with Classical and Modern Approaches

Han-Jia Ye
Nanjing University

Da-Wei Zhou
Nanjing University

Zhenguo Li
HKUST & Autel

Outline

- **Introduction, Steps of Model Reuse**
- **Challenges in Reusing PTMs**
- **Selecting PTMs**
  - PTM selection and transferability measure
  - Semantic-based, Metric-based, and Learning-based approaches.
  - Model Routing
- **Related Topics in Selecting PTMs and Discussions**

Han-Jia Ye

- **Reuse Pre-Trained Models**
  - Reuse PTMs from data-level, model-level, and algorithm level.
  - Reuse heterogeneous models.
  - Reuse multiple PTMs.
  - Other types of model reuse.
- **Applications**

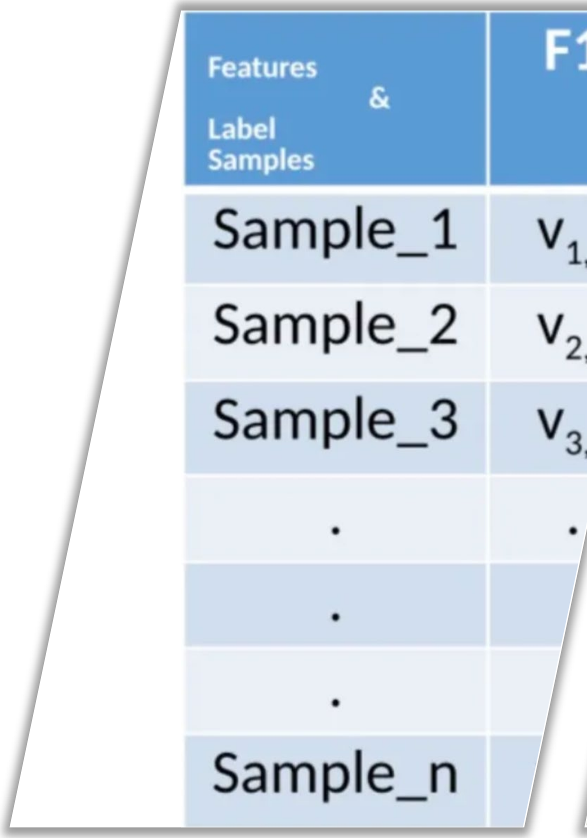
Da-Wei Zhou

Slides of the Tutorial

Introduction

Machine Learning Applications

Machine learning has been applied in various fields successfully.



Features & Label Samples	F1
Sample_1	$v_{1,1}$
Sample_2	$v_{2,1}$
Sample_3	$v_{3,1}$
.	.
.	
.	
Sample_n	

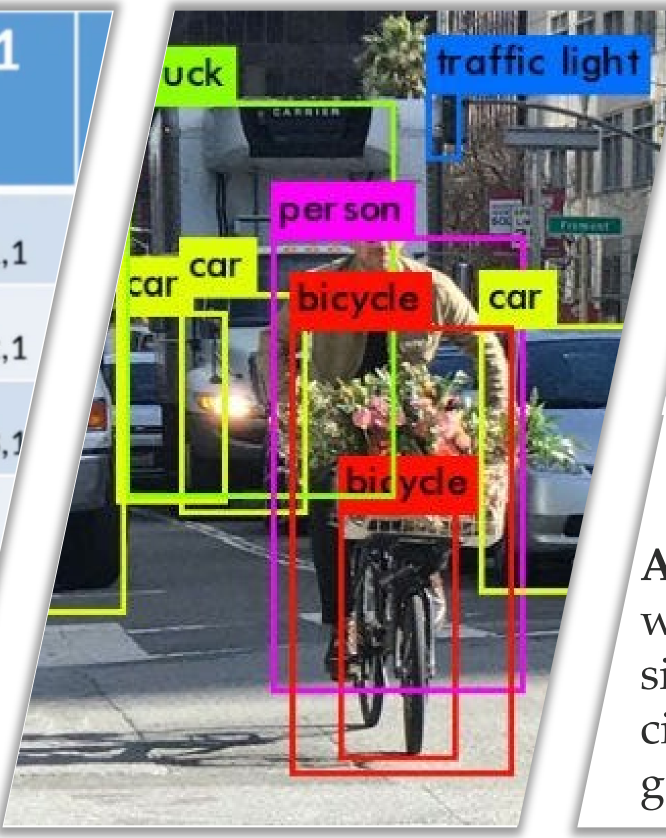


Image recognition and detection

Q: Could you depict the picture?

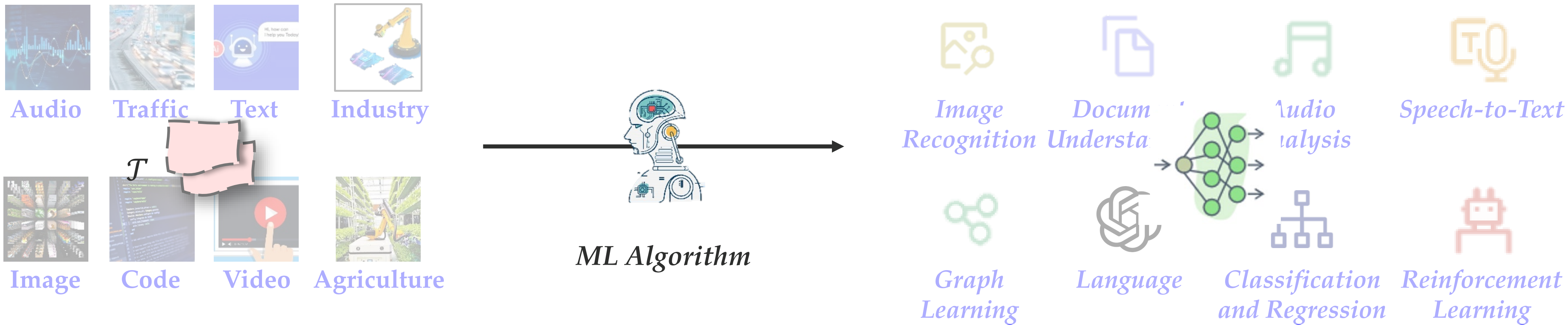
A: The image shows a white motor yacht cruising on the water with a city skyline in the back ground.

Vision-language understanding

Tiny potato kings: wearing intricate crowns, seated on grand thrones, overlooking a vast kingdom made of potato farms and villages.

Image/text generation

Machine Learning Paradigm



Classical learning paradigm

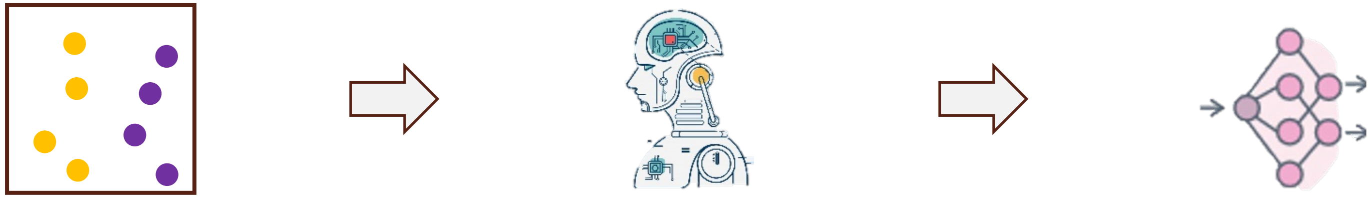
- Given a task $\mathcal{T} = \{(\boldsymbol{x}_i, y_i)_{i=1}^N\}$
- The model $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$, contains the architecture f and weights θ is learned via $f_\theta = \mathcal{A}(\theta_0, f \mid \mathcal{D})$, e.g.,

$$\min_{f_\theta} \sum_{(\boldsymbol{x}_i, y_i) \sim \mathcal{T}} \ell(f_\theta(\boldsymbol{x}_i), y_i)$$

Learn the model ***from scratch***. For example, with gradient-based optimizer, we update θ from initialization θ_0 with gradients $\nabla_\theta \ell$.

We need to carefully determine *which data to use, the model architecture, training objective*, and *inference strategies*, etc.

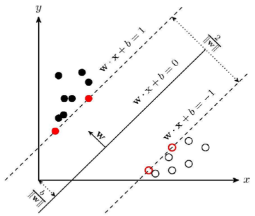
Machine Learning Pre-trained Models



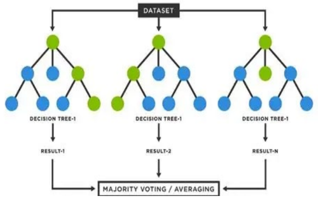
Input Data *ML Algorithm* *Output Model*

A Pre-Trained Model (PTM) is a machine learning model (usually well-trained) based on a certain dataset.

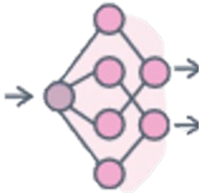
These PTMs encapsulate *valuable inductive biases* beneficial for downstream tasks.


y
$\mathbf{w}\cdot\mathbf{x}+b=1$
$\mathbf{w}\cdot\mathbf{x}+b=0$
$\mathbf{w}\cdot\mathbf{x}+b=-1$
$\frac{2}{\|\mathbf{w}\|}$
$\mathbf{w}$
$\frac{b}{\|\mathbf{w}\|}$
x

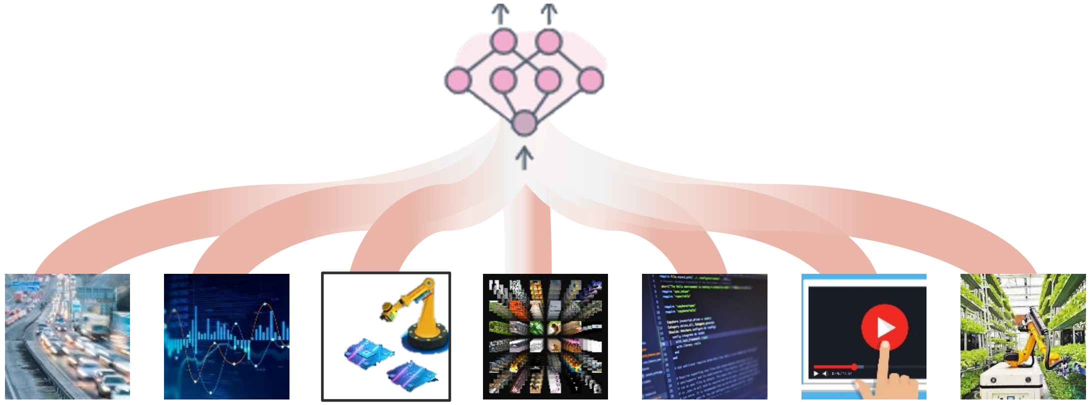
Linear model



Decision Tree



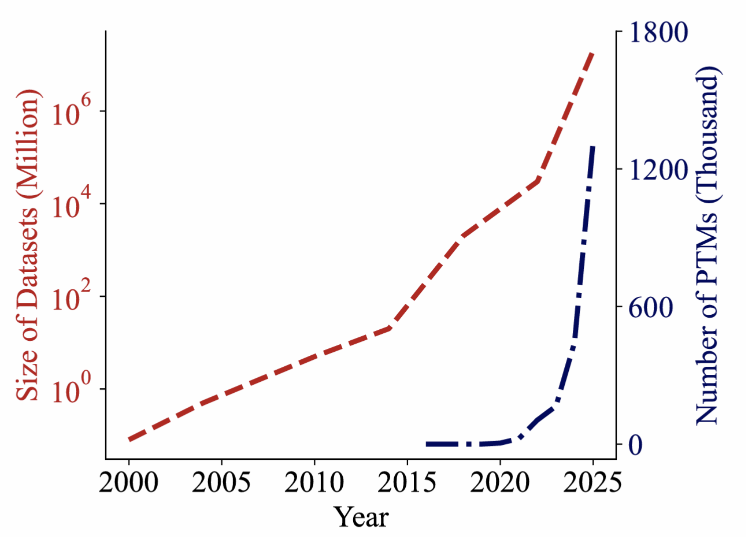
Neural Network



Transportation *Audio* *Industry* *Image* *Code* *Video* *Agriculture*

Machine Learning Pre-trained Models

Hugging Face



1.3M+ PTMs

Hugging Face https://huggingface.co/

ModelZoo https://modelzoo.co/

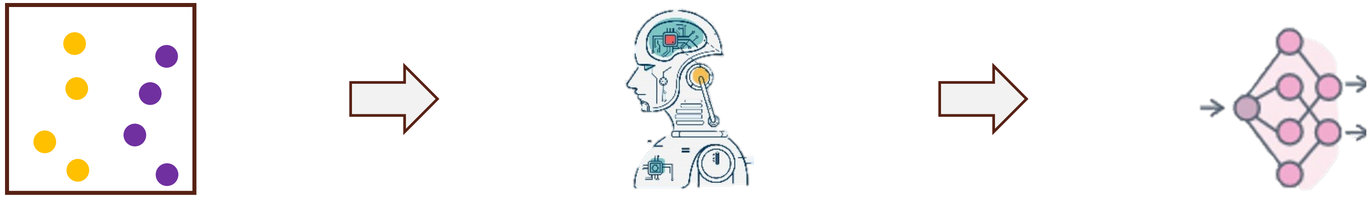
kaggle https://www.kaggle.com/models

…….

Types of PTMs

- Multi-modal: image2text, text2image, …
- Vision: image classification, detection, …
- Language: text generation, Translation, …
- Tabular: classification, regression, …
- Audio: speech recognition, text2speech, …
- RL: robotics, …

Learning Paradigm

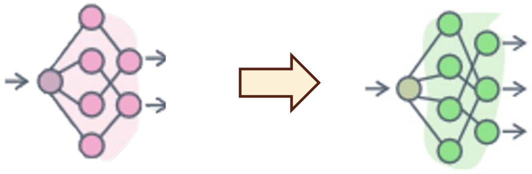


Input Data

ML Algorithm

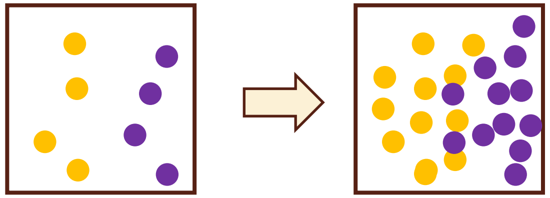
Output Model

Model-Centric



improve performance by
changing the model

Data-Centric



improve performance by
changing the data

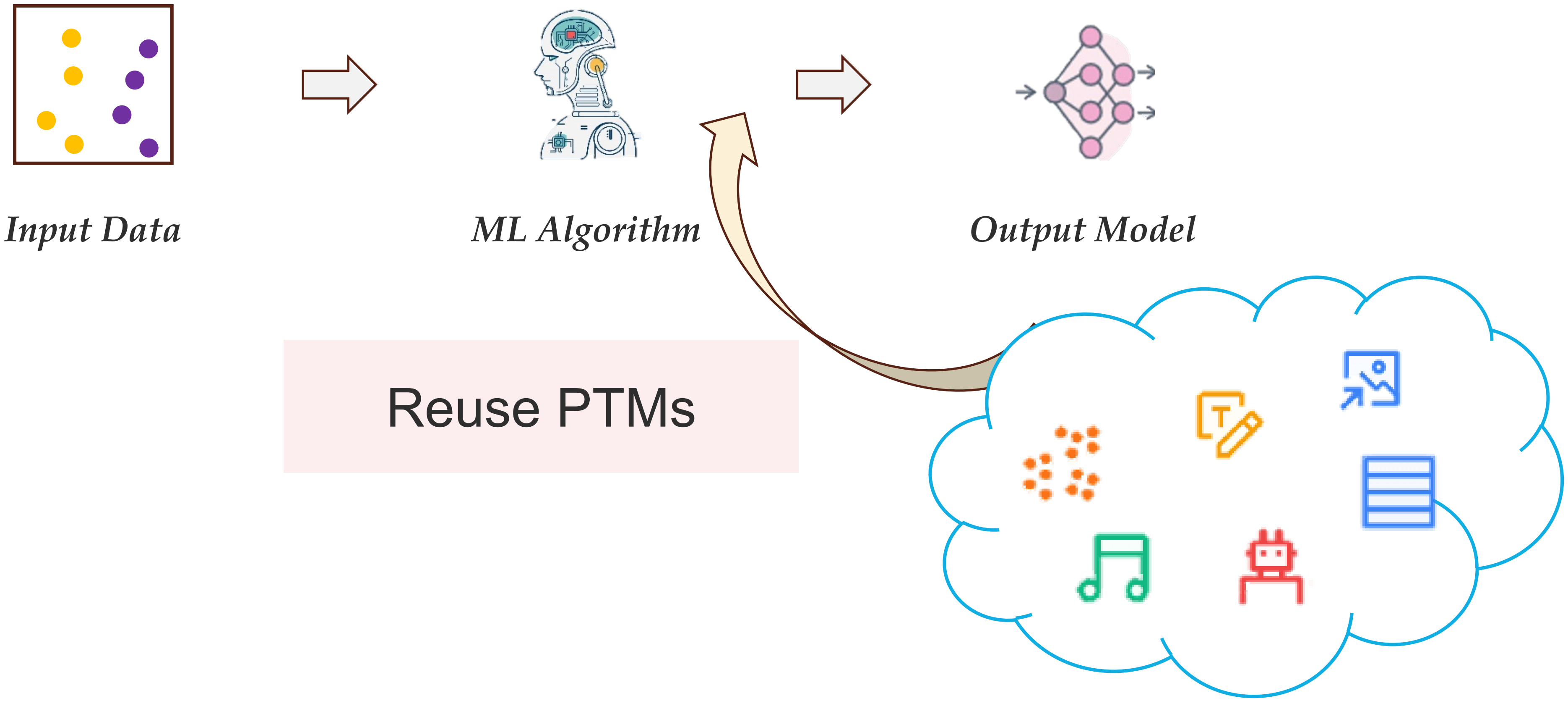
Next step

https://www.datacentricai.org/neurips21/; https://dmlr.ai/

data-centric

Learning from Data → Learning from Data and *Pre-trained Models*

pre-trained-model-centric



Input Data

ML Algorithm

Output Model

Reuse PTMs

Possible Usage

Incorporate a set of PTMs into the machine learning process.

```
from model_reuse_lib import Client
client = Client()
target_model = client.fit(training_data=(train_X, train_Y), model_zoo_path=zip_path)
pred_y = target_model.predict(test_x)
```

Expert-level Ability

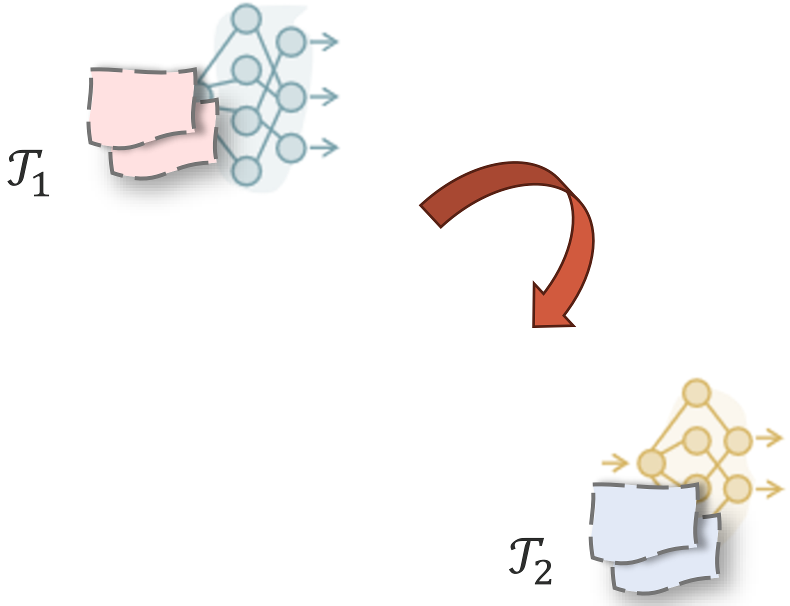
Few-Shot Training

Avoid Forgetting

…

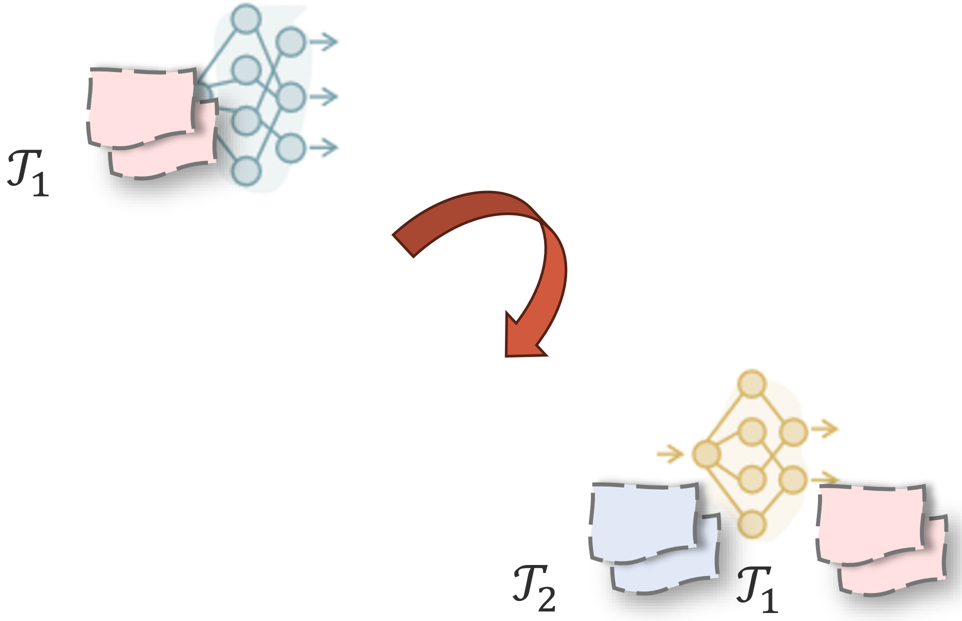
Advantages and Goals of Model Reuse

Knowledge Transfer, $A \rightarrow B$



- Make the target model f more discriminative
- The target task could be unlabeled/few-shot

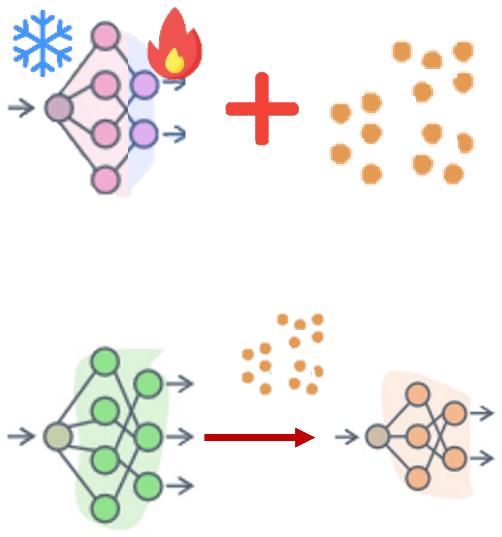
Knowledge Aggregation, $A + B$



- To improve the target model
- Make a model capable of various tasks

Possible Reuse Paradigms

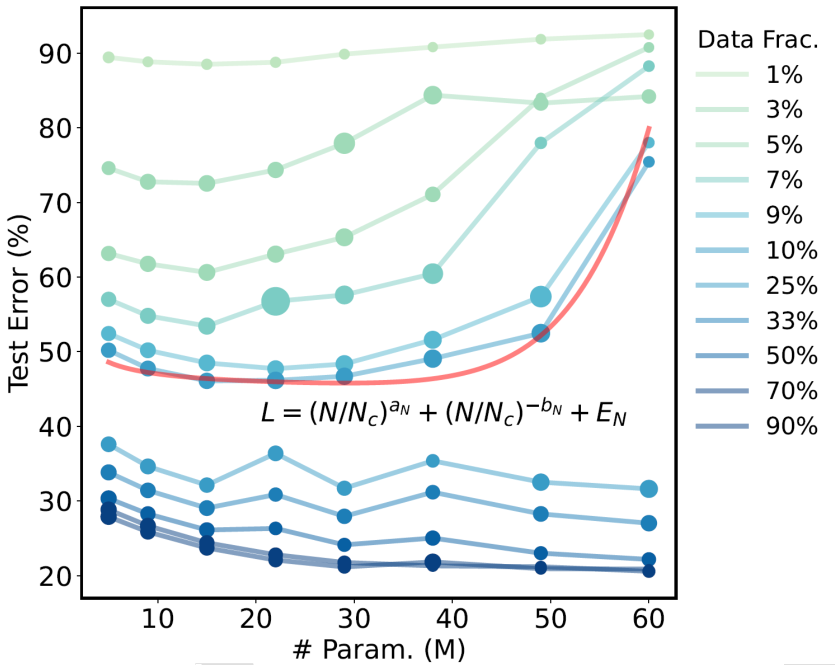
Reuse a single (given) PTM



- By building on existing models, reuse *accelerates training, lowers computational and data costs, and mitigates catastrophic forgetting* in dynamic or continually evolving environments.
- With well-designed reuse strategies, knowledge from these PTMs can be efficiently extracted and *extended beyond their original scope*, facilitating numerous applications.

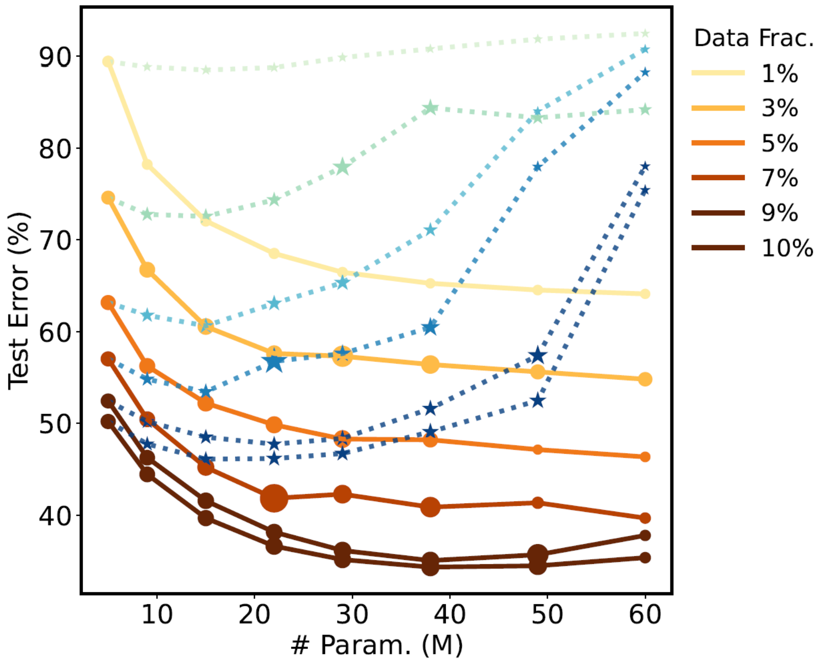
Reuse PTMs Instead of Learning from Scratch

Different from the common neural scaling law (the performance, training data, and model size generally should follow a *power law*), the performance of a model drops significantly in the data scarcity case [Wang, Panda, Wang, ICML'23].



Scaling model size for ViT

The common power law is broken once entering the data scarcity regime.

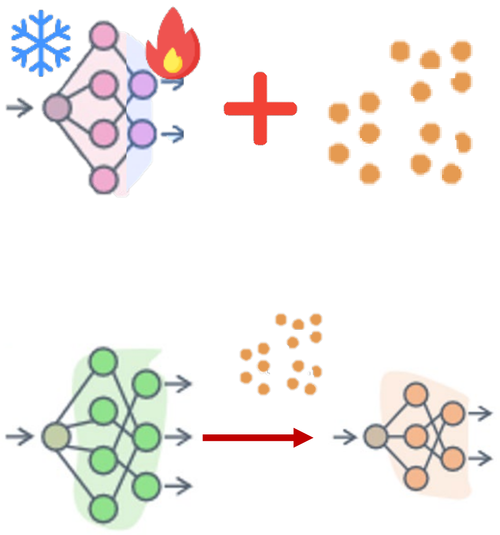


Scaling model size for ViT

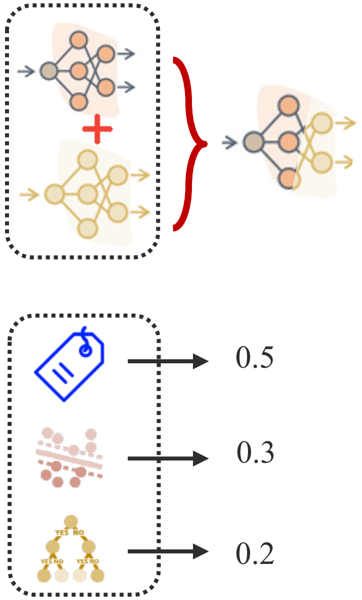
*By **reusing** PTM, the power law is reproduced under the data scarcity regime.*

Possible Reuse Paradigms

Reuse a single (given) PTM



Reuse PTMs from a model zoo



Given a fixed set of PTMs $\mathcal{G} = \{g^1_{\boldsymbol{\Theta}_1}, g^2_{\boldsymbol{\Theta}_2}, \dots, g^M_{\boldsymbol{\Theta}_M}\}$, design a ***model reuse*** algorithm $\mathcal{A}'$ such that

$$f_\theta = \mathcal{A}'(\theta_0, f \mid \mathcal{T}; \mathcal{G})$$

For example,

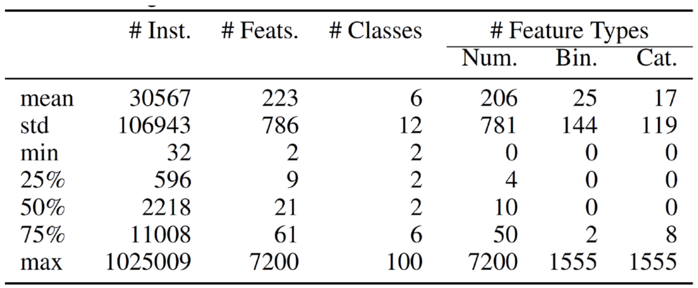
$$\min_{f_\theta} \sum_{(\boldsymbol{x}_i, y_i) \sim \mathcal{T}} \ell\left(f_\theta\left(\boldsymbol{x}_i \middle| \left\{g^1_{\boldsymbol{\Theta}_1}, g^2_{\boldsymbol{\Theta}_2}, \dots, g^M_{\boldsymbol{\Theta}_M}\right\}\right), y_i\right)$$

- By building on existing models, reuse *accelerates training, lowers computational and data costs, and mitigates catastrophic forgetting* in dynamic or continually evolving environments.
- With well-designed reuse strategies, knowledge from these PTMs can be efficiently extracted and *extended beyond their original scope*, facilitating numerous applications.

Use Foundation Models?

Why not utilize one foundation model for different tasks directly?

One foundation model cannot deal with different tasks.

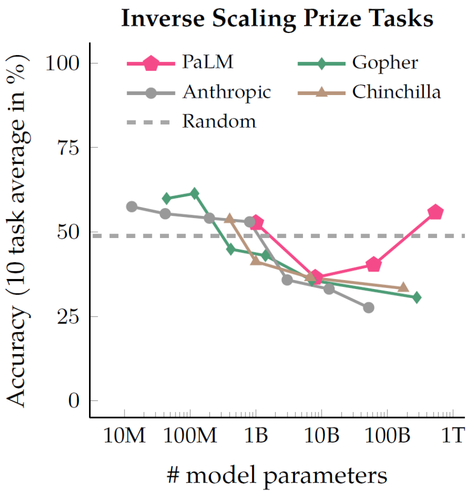


	# Inst.	# Feats.	# Classes	# Feature Types		
				Num.	Bin.	Cat.
mean	30567	223	6	206	25	17
std	106943	786	12	781	144	119
min	32	2	2	0	0	0
25%	596	9	2	4	0	0
50%	2218	21	2	10	0	0
75%	11008	61	6	50	2	8
max	1025009	7200	100	7200	1555	1555

Summary statistics for all 176 tabular datasets.

[Grinsztajn, Oyallon, Varoquaux. NeurIPS’22]

Inverse scaling law indicates larger model may not help in some cases.



[Wei et al., EMNLP’23]

There exist ability divergence among foundation models.

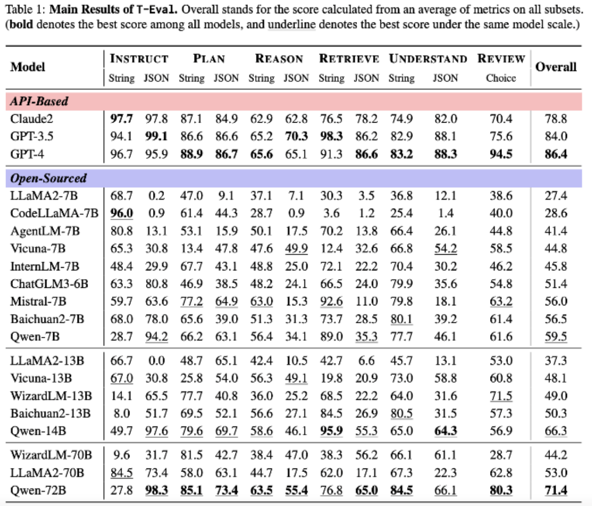


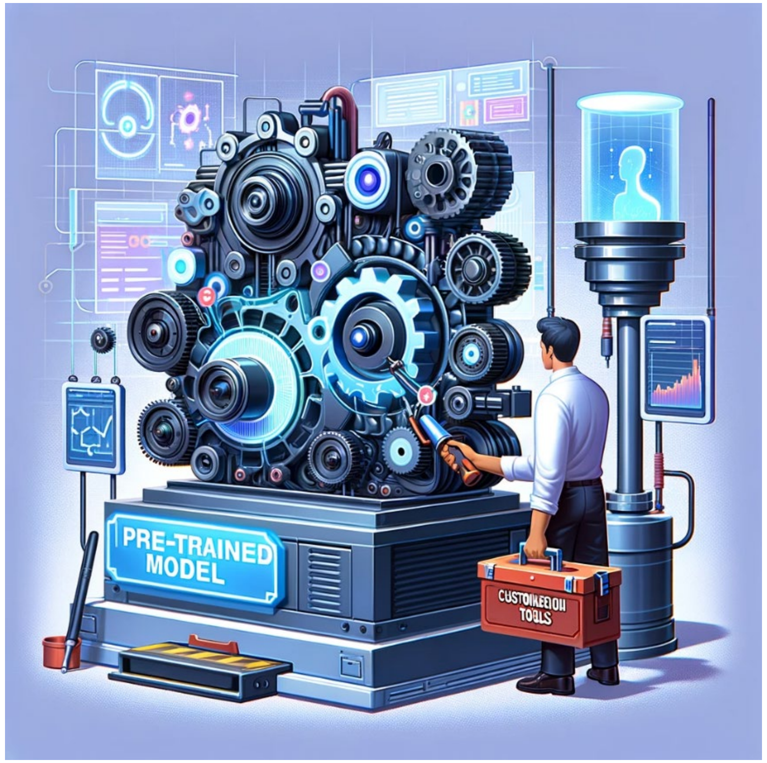
Table 1: **Main Results of T-Eval.** Overall stands for the score calculated from an average of metrics on all subsets. (**bold** denotes the best score among all models, and underline denotes the best score under the same model scale.)

Model	INSTRUCT		PLAN		REASON		RETRIEVE		UNDERSTAND		REVIEW	Overall
	String	JSON	String	JSON	String	JSON	String	JSON	String	JSON	Choice	
API-Based												
Claude2	**97.7**	97.8	87.1	84.9	62.9	62.8	76.5	78.2	74.9	82.0	70.4	78.8
GPT-3.5	94.1	**99.1**	86.6	86.6	65.2	**70.3**	**98.3**	86.2	82.9	88.1	75.6	84.0
GPT-4	96.7	95.9	**88.9**	**86.7**	**65.6**	65.1	91.3	**86.6**	**83.2**	**88.3**	**94.5**	**86.4**
Open-Sourced												
LLaMA2-7B	68.7	0.2	47.0	9.1	37.1	7.1	30.3	3.5	36.8	12.1	38.6	27.4
CodeLLaMA-7B	**96.0**	0.9	61.4	44.3	28.7	0.9	3.6	1.2	25.4	1.4	40.0	28.6
AgentLM-7B	80.8	13.1	53.1	15.9	50.1	17.5	70.2	13.8	66.4	26.1	44.8	41.4
Vicuna-7B	65.3	30.8	13.4	47.8	47.6	49.9	12.4	32.6	66.8	54.2	58.5	44.8
InternLM-7B	48.4	29.9	67.7	43.1	48.8	25.0	72.1	22.2	70.4	30.2	46.2	45.8
ChatGLM3-6B	63.3	80.8	46.9	38.5	48.2	24.1	66.5	24.0	79.9	35.6	54.8	51.4
Mistral-7B	59.7	63.6	77.2	64.9	63.0	15.3	92.6	11.0	79.8	18.1	63.2	56.0
Baichuan2-7B	68.0	78.0	65.6	39.0	51.3	31.3	73.7	28.5	80.1	39.2	61.4	56.5
Qwen-7B	28.7	94.2	66.2	63.1	56.4	34.1	89.0	35.3	77.7	46.1	61.6	59.5
LLaMA2-13B	66.7	0.0	48.7	65.1	42.4	10.5	42.7	6.6	45.7	13.1	53.0	37.3
Vicuna-13B	67.0	30.8	25.8	54.0	56.3	49.1	19.8	20.9	73.0	58.8	60.8	48.1
WizardLM-13B	14.1	65.5	77.7	40.8	36.0	25.2	68.5	22.2	64.0	31.6	71.5	49.0
Baichuan2-13B	8.0	51.7	69.5	52.1	56.6	27.1	84.5	26.9	80.5	31.5	57.3	50.3
Qwen-14B	49.7	97.6	79.6	69.7	58.6	46.1	**95.9**	55.3	65.0	**64.3**	56.9	66.3
WizardLM-70B	9.6	31.7	81.5	42.7	38.4	47.0	38.3	56.2	66.1	61.1	28.7	44.2
LLaMA2-70B	84.5	73.4	58.0	63.1	44.7	17.5	62.0	17.1	67.3	22.3	62.8	53.0
Qwen-72B	27.8	**98.3**	**85.1**	**73.4**	**63.5**	**55.4**	76.8	**65.0**	**84.5**	66.1	**80.3**	**71.4**

[Chen et al., CoRR’23]

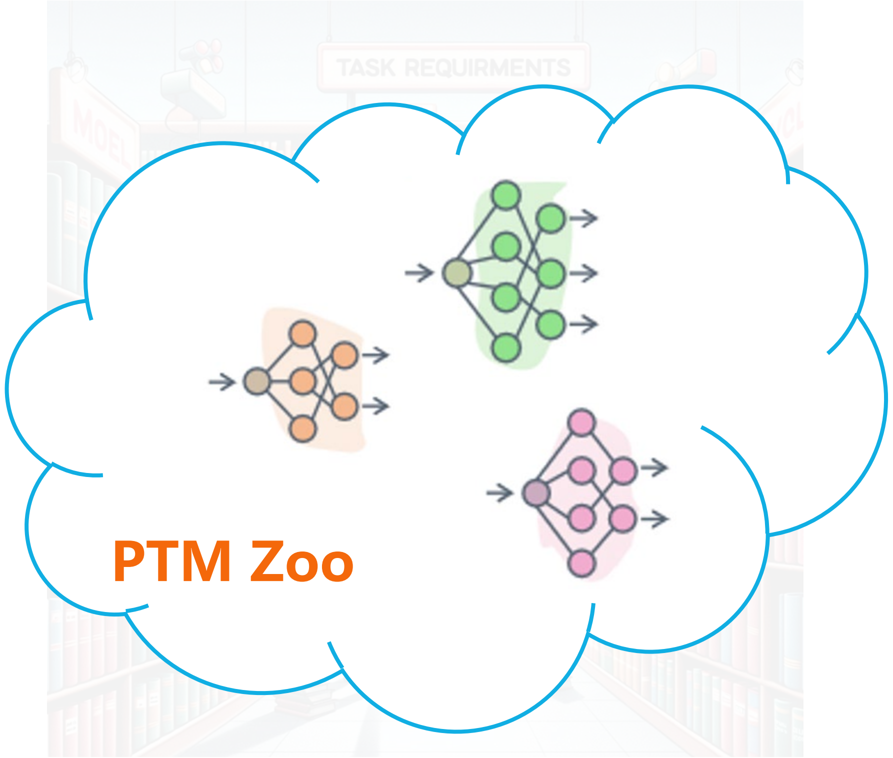
How to Reuse PTMs Effectively?

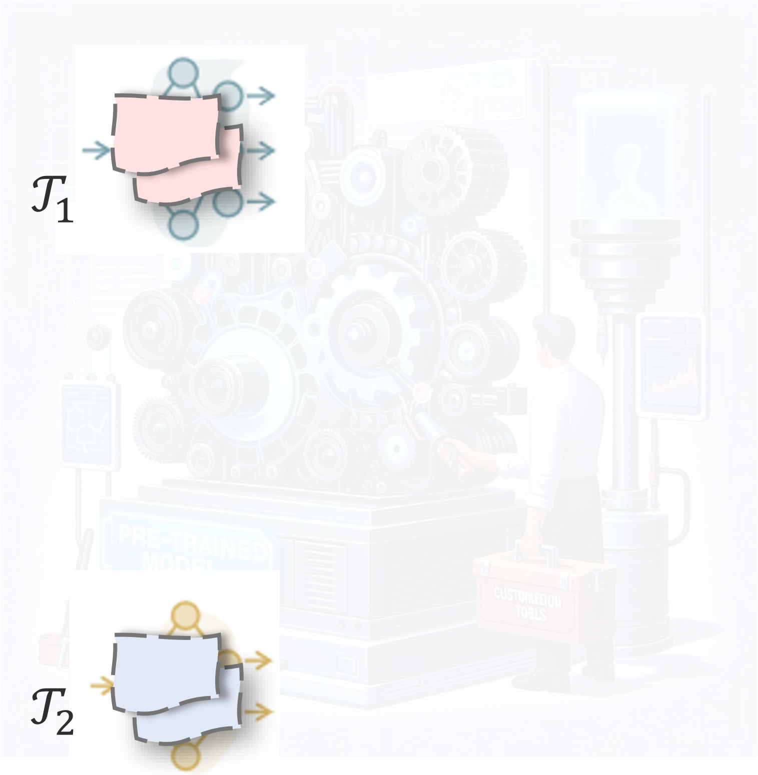
PTM Selection



PTM Adaptation

PTM Selection and Adaptation





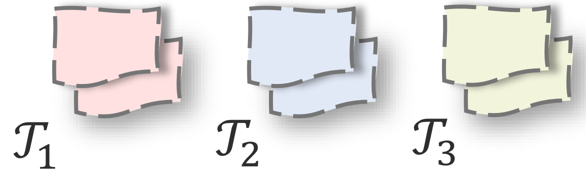
PTM Selection

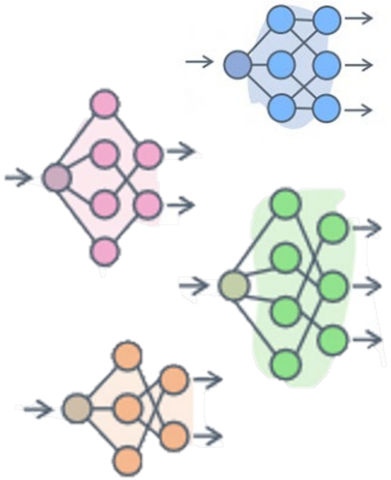
PTM Adaptation

[Zhou, FCS'16]

PTM Selection

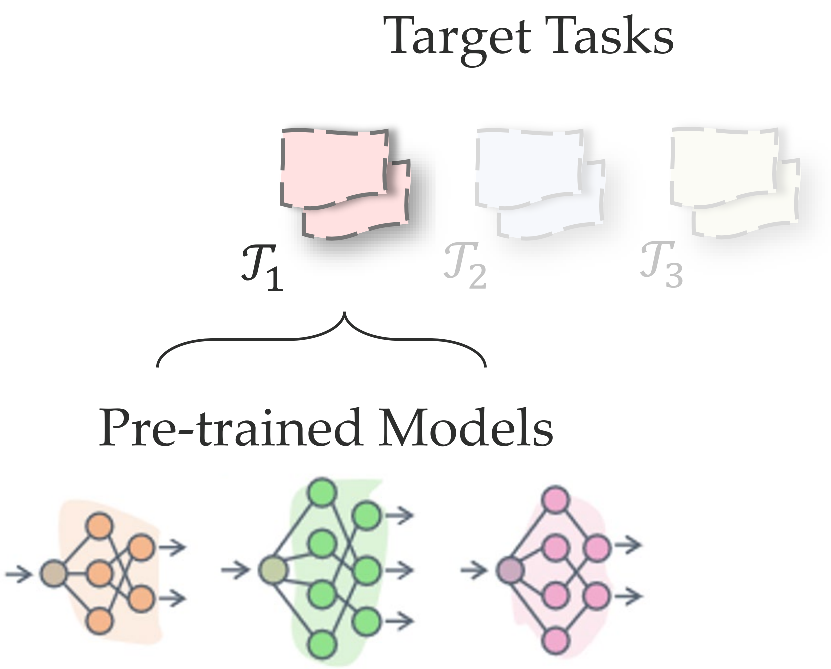
Target Tasks





Pre-trained Models

PTM Selection



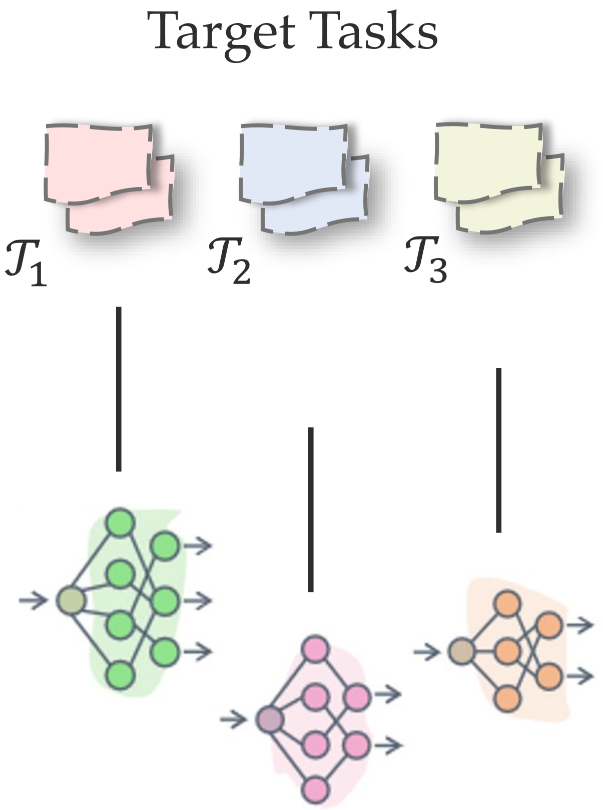
Target Tasks

$\mathcal{T}_1$ $\mathcal{T}_2$ $\mathcal{T}_3$

Pre-trained Models

Fine-tuned performance: **85%** **93%** **73%**

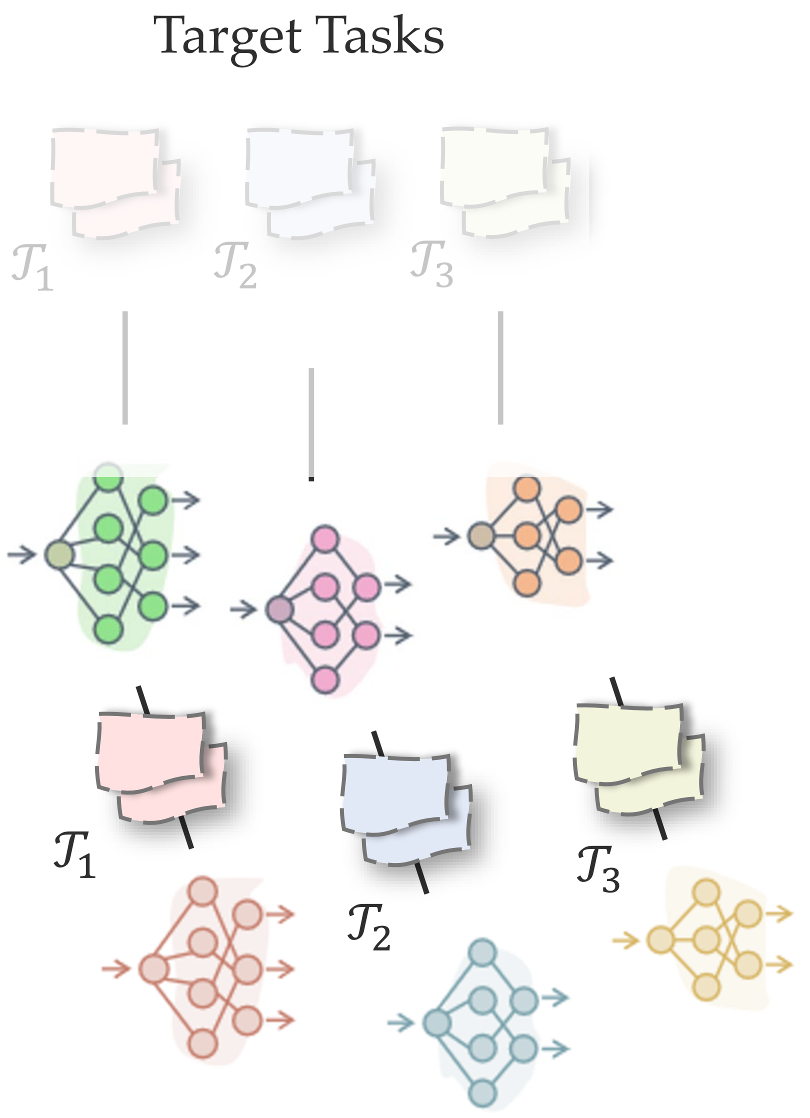
PTM Selection



Pre-trained Model **Selection**

selecting **suitable** pre-trained models for target tasks

PTM Adaptation



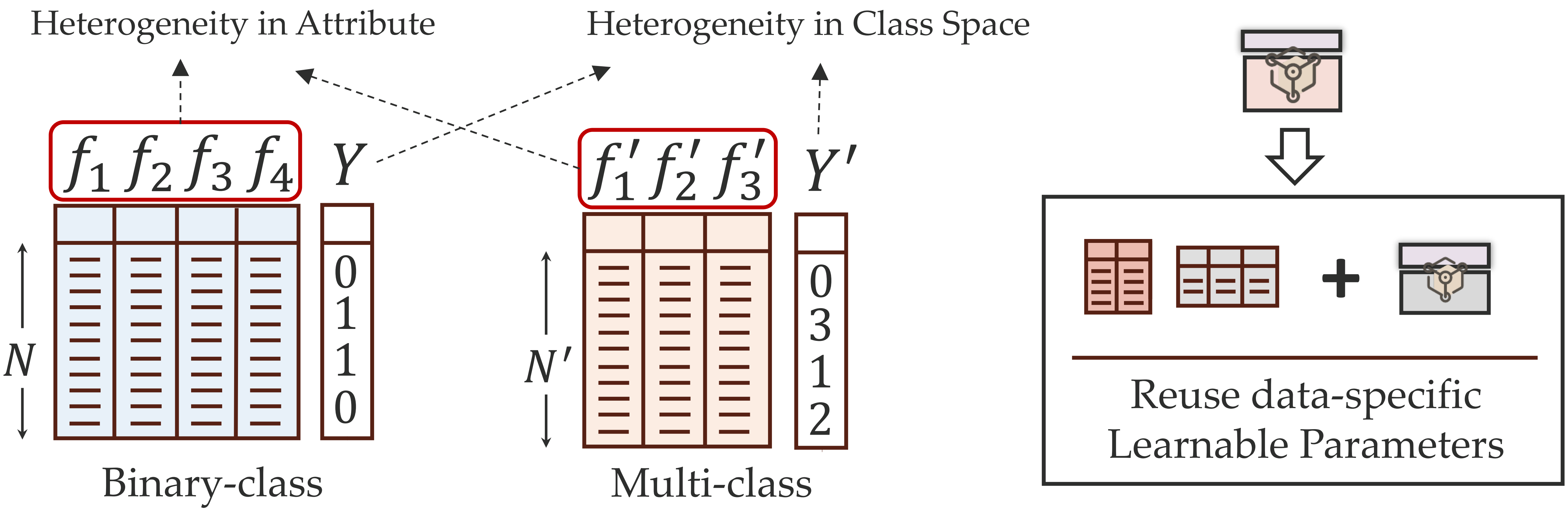
Adapt selected PTMs towards the target task

Across various domains, e.g., tabular data [Kuzborskij and Orabona, ICML’13], image [Zhou et al., IJCV’22], texts [He et al., ICLR’22], …

Across various tasks, e.g., zero/few-shot learning [Radford et al., ICML’21], continual learning [Zhou et al., IJCAI’24], …

Select “**most suitable**” PTMs first, and then apply the adaptation scheme to improve the ***joint*** performance

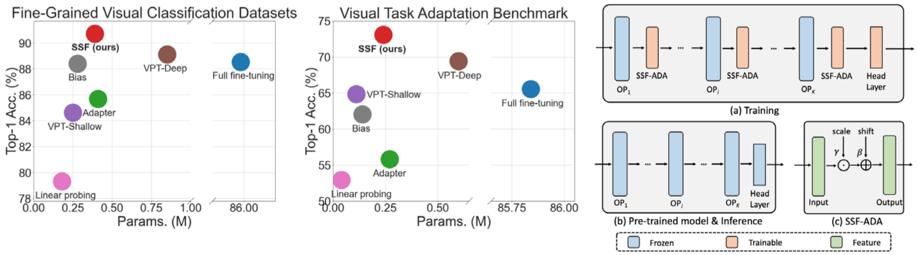
PTM Adaptation for Tabular Data


Heterogeneity in Attribute
Heterogeneity in Class Space
f_1 f_2 f_3 f_4
Y
N
0
1
1
0
Binary-class
f'_1 f'_2 f'_3
Y'
N'
0
3
1
2
Multi-class
+
Reuse data-specific Learnable Parameters

Reuse the pre-trained classification/regression model on one or more tabular datasets to a target tabular dataset [Kuzborskij and Orabona, ICML'13][Ye et al., ICML'18][Zhu et al., ICML'23][Zhou et al., CoRR'23].

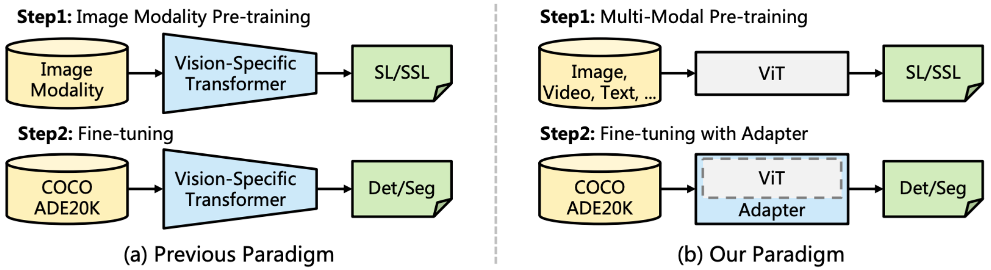
PTM Adaptation for Vision and NLP Tasks

Image Classification



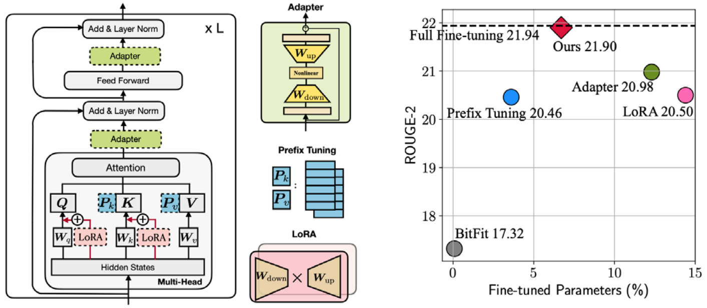
[Lian et al., NeurIPS’22]

Object Detection


Step1: Image Modality Pre-training
Image Modality
Vision-Specific Transformer
SL/SSL
Step2: Fine-tuning
COCO ADE20K
Vision-Specific Transformer
Det/Seg
(a) Previous Paradigm
Step1: Multi-Modal Pre-training
Image, Video, Text, ...
ViT
SL/SSL
Step2: Fine-tuning with Adapter
COCO ADE20K
ViT
Adapter
Det/Seg
(b) Our Paradigm

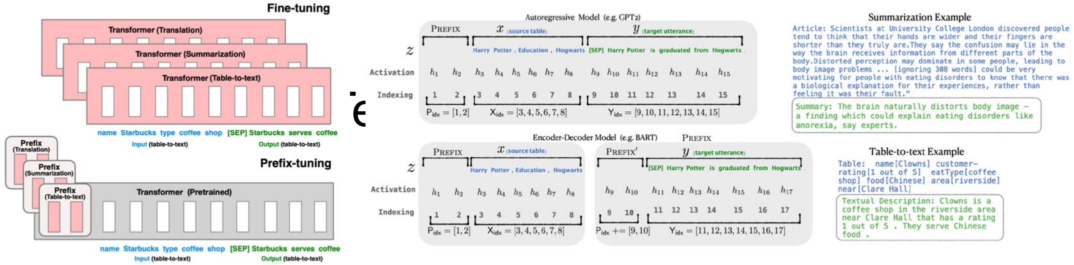
[Chen et al., ICLR’23]

Machine Translation



[He et al., ICLR’22]

Text Summarization


Fine-tuning
Transformer (Translation)
Transformer (Summarization)
Transformer (Table-to-text)
name Starbucks type coffee shop [SEP] Starbucks serves coffee
Input (table-to-text)
Output (table-to-text)
Prefix (Translation)
Prefix (Summarization)
Prefix (Table-to-text)
Prefix-tuning
Transformer (Pretrained)
name Starbucks type coffee shop [SEP] Starbucks serves coffee
Input (table-to-text)
Output (table-to-text)
Autoregressive Model (e.g. GPT2)
PREFIX
x (source table)
y (target utterance)
z
Harry Potter , Education , Hogwarts [SEP] Harry Potter is graduated from Hogwarts .
Activation
h_1 h_2 h_3 h_4 h_5 h_6 h_7 h_8 h_9 h_{10} h_{11} h_{12} h_{13} h_{14} h_{15}
Indexing
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
$P_{idx} = [1, 2]$
$X_{idx} = [3, 4, 5, 6, 7, 8]$
$Y_{idx} = [9, 10, 11, 12, 13, 14, 15]$
Encoder-Decoder Model (e.g. BART)
PREFIX
x (source table)
PREFIX′
y (target utterance)
z
Harry Potter , Education , Hogwarts
[SEP] Harry Potter is graduated from Hogwarts .
Activation
h_1 h_2 h_3 h_4 h_5 h_6 h_7 h_8 h_9 h_{10} h_{11} h_{12} h_{13} h_{14} h_{15} h_{16} h_{17}
Indexing
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
$P_{idx} = [1, 2]$
$X_{idx} = [3, 4, 5, 6, 7, 8]$
$P_{idx} += [9, 10]$
$Y_{idx} = [11, 12, 13, 14, 15, 16, 17]$
Summarization Example
Table-to-text Example

Article: Scientists at University College London discovered people tend to think that their hands are wider and their fingers are shorter than they truly are.They say the confusion may lie in the way the brain receives information from different parts of the body.Distorted perception may dominate in some people, leading to body image problems ... [ignoring 308 words] could be very motivating for people with eating disorders to know that there was a biological explanation for their experiences, rather than feeling it was their fault."

Summary: The brain naturally distorts body image – a finding which could explain eating disorders like anorexia, say experts.

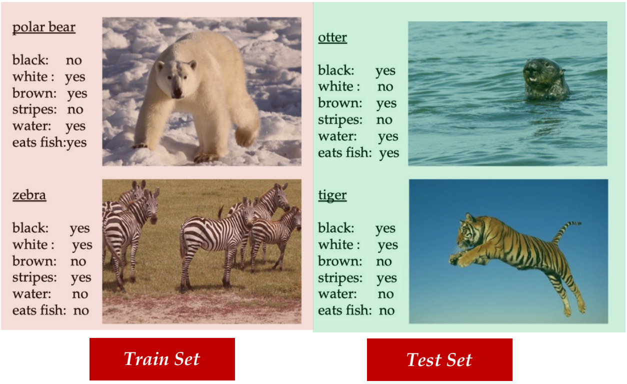
Table: name[Clowns] customer-rating[1 out of 5] eatType[coffee shop] food[Chinese] area[riverside] near[Clare Hall]

Textual Description: Clowns is a coffee shop in the riverside area near Clare Hall that has a rating 1 out of 5 . They serve Chinese food .

[Li et al., ACL’21]

PTM Adaptation for Zero/Few-Shot Learning

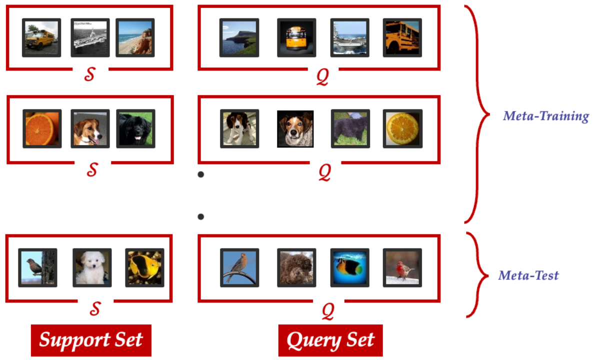
Zero-Shot Learning



The classes in train set and test sets are ***nonoverlapping***.

The task needs ***side information*** (attributes).

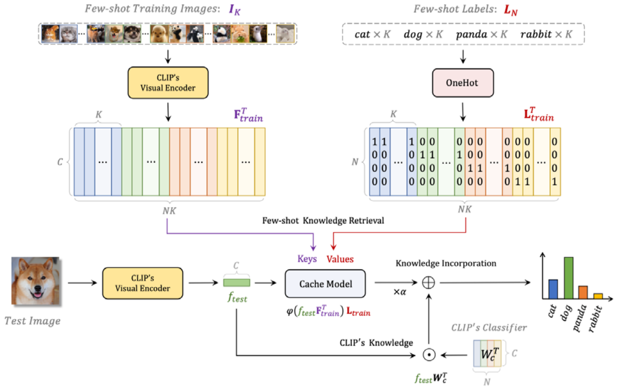
Few-Shot Learning



The classes in meta-training and meta-test stage are ***nonoverlapping***.

Learning new classes with few-shot data in meta-test stage.

PTM for Zero-Shot/Few-Shot Learning



Constructs the adapter via a key-value cache model from the few-shot data [Zhang et al., ECCV'22].

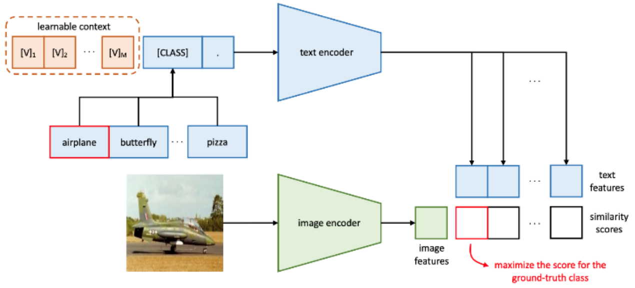


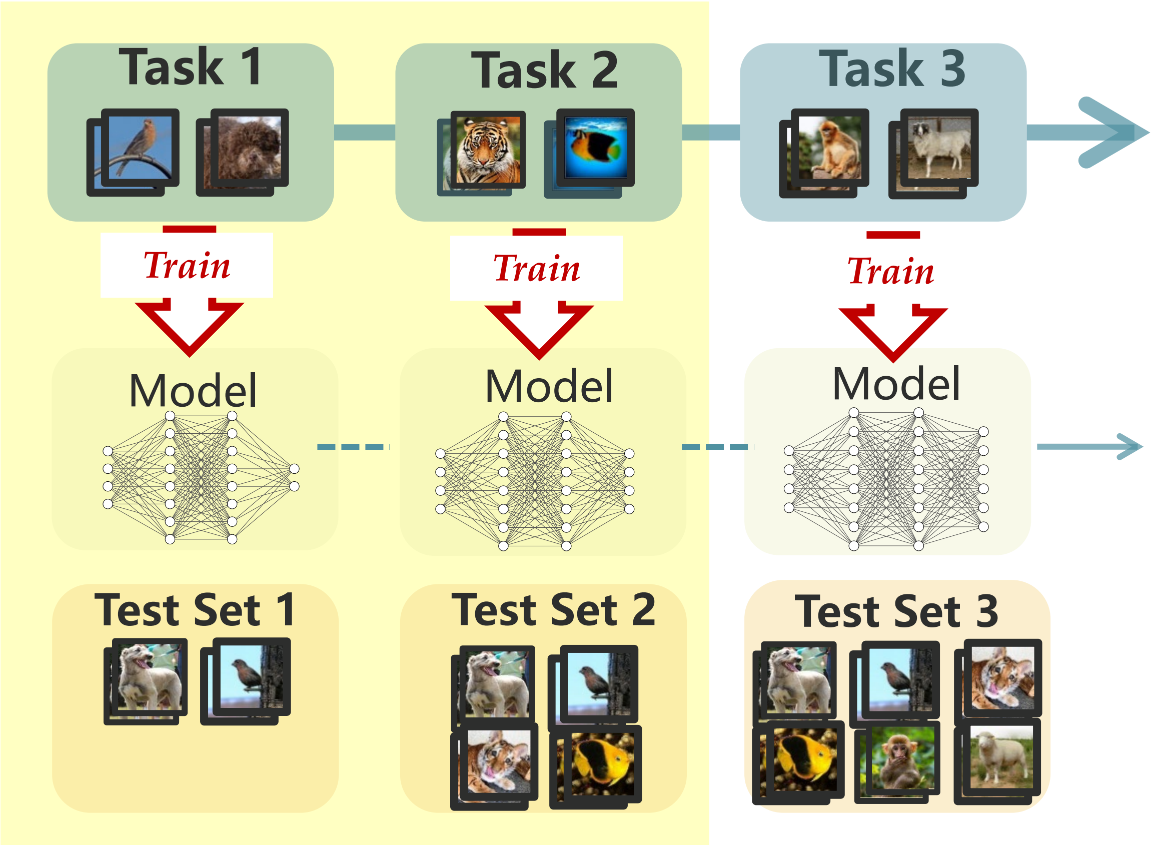
Figure 2: Overview of context optimization (CoOp).

Adapt the text encoder and keep the zero-shot ability of a VLM [Zhou et al., IJCV'22].

Reuse a PTM, and construct a classifier on the target task even with zero/few training data.

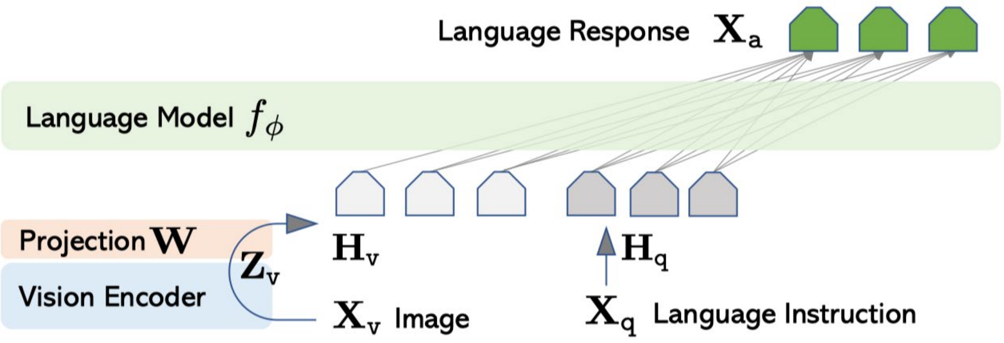
PTM Adaptation for Continual Learning

- ***Continual Learning (a.k.a. incremental learning/lifelong learning)*** refers to the scenario where the model needs to be *continually updated with new data*.
- Data comes in the *stream* format, which cannot be held for storage or privacy issues [Chen, Liu, Lifelong Machine Learning, 2018] [Ven, Tuytelaars, Tolias, NMI'22].

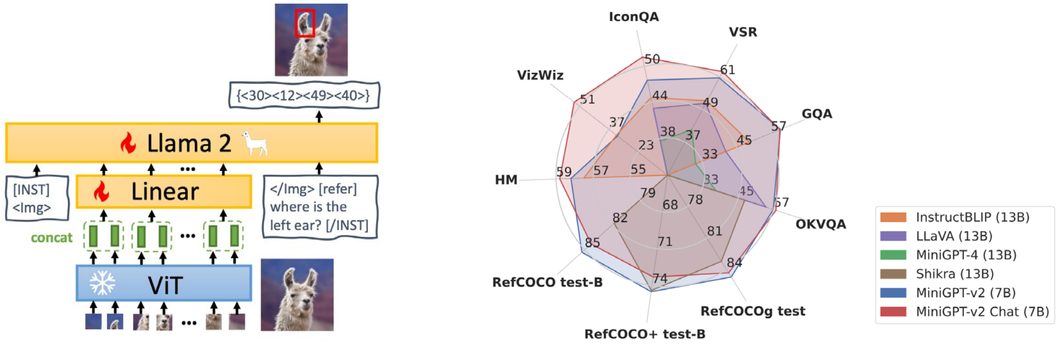


Reuse the old model given the data from the new task, without fully access to the data from old classes [Zhou et al. IJCAI'24][Zhou et al., TPAMI'25].

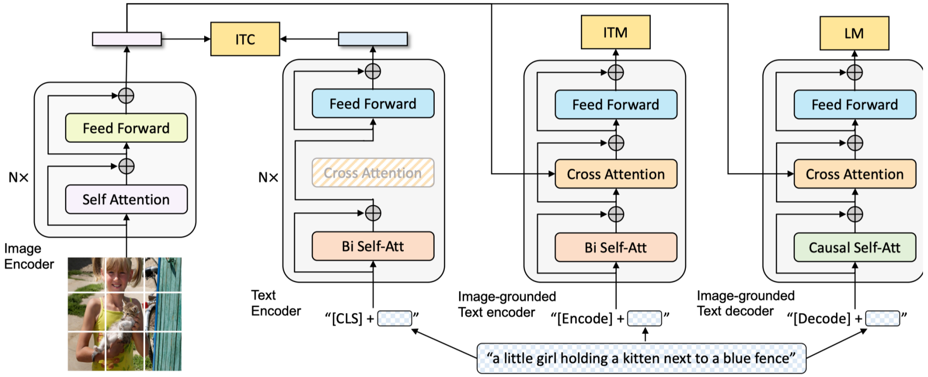
PTM Adaptation for Large Language Models



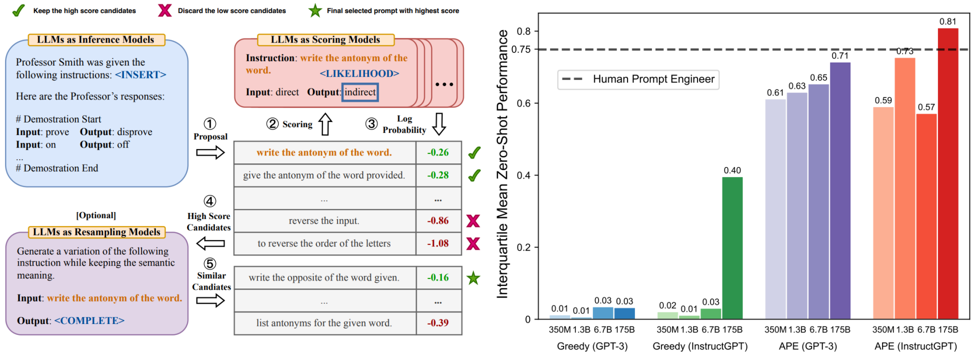
[Liu et al., NeurIPS’23]



[Chen et al., NeurIPS’23]



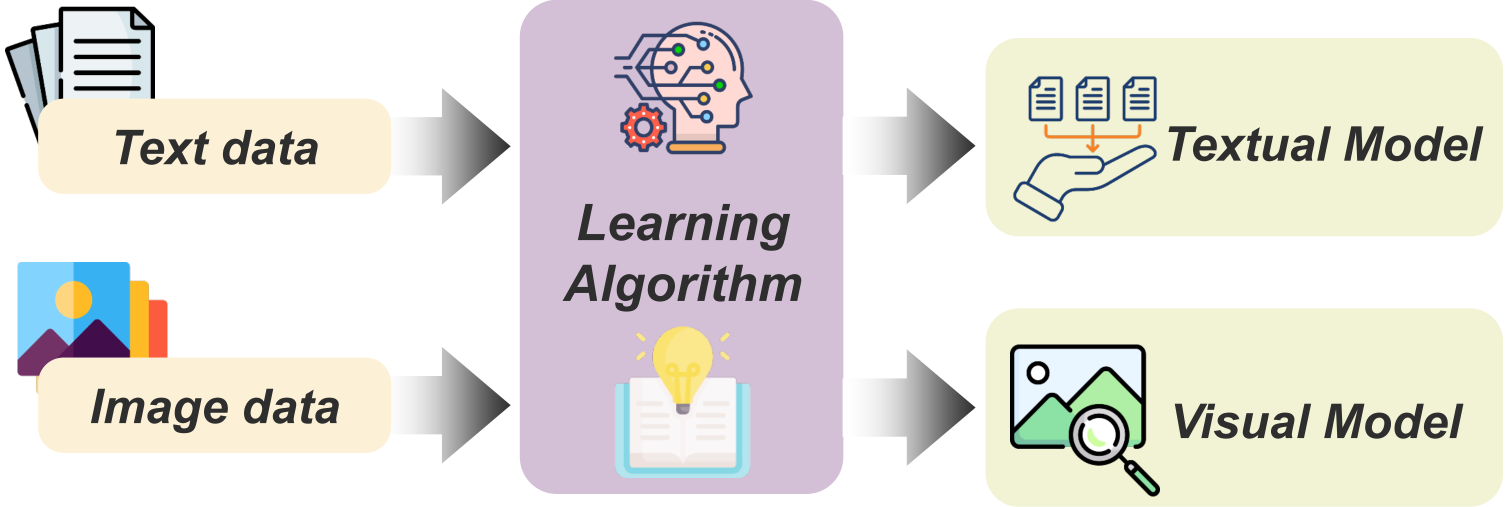
[Li et al., ICML’22]



[Zhou et al., ICLR’23]

A Short Summary of Model Reuse

Typical Machine Learning Process



Text data

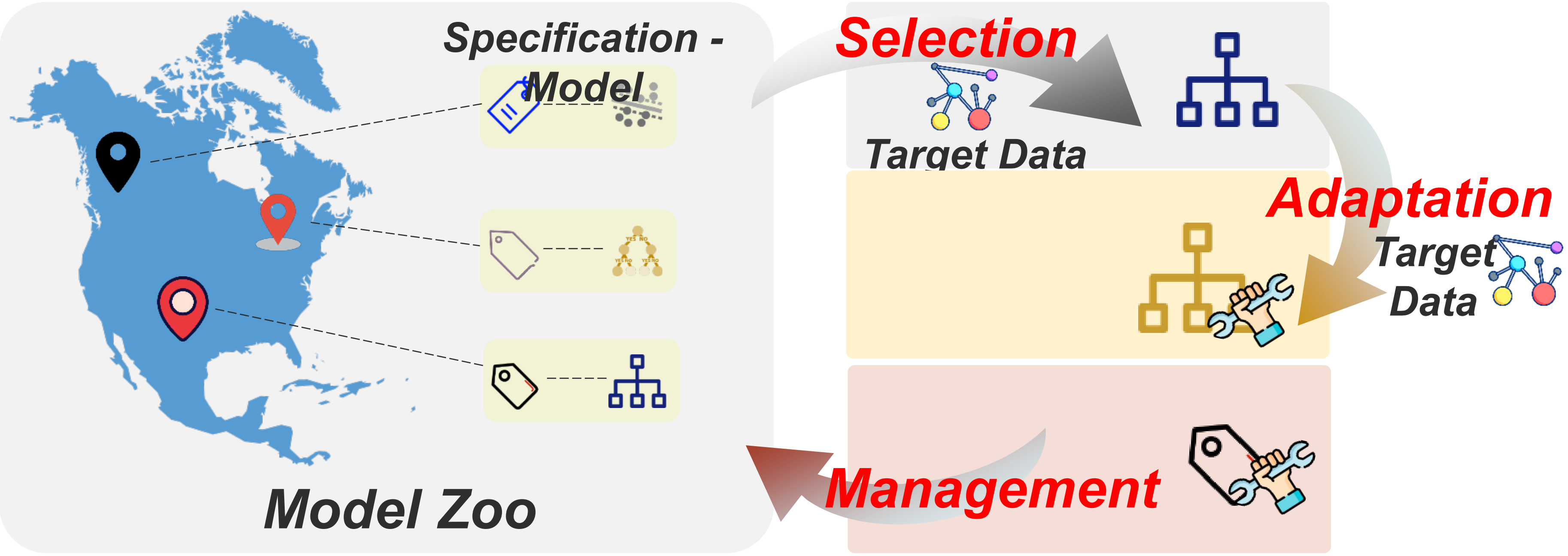
Image data

Learning Algorithm

Textual Model

Visual Model

Learning via Model Reuse



Specification - Model

Model Zoo

Selection

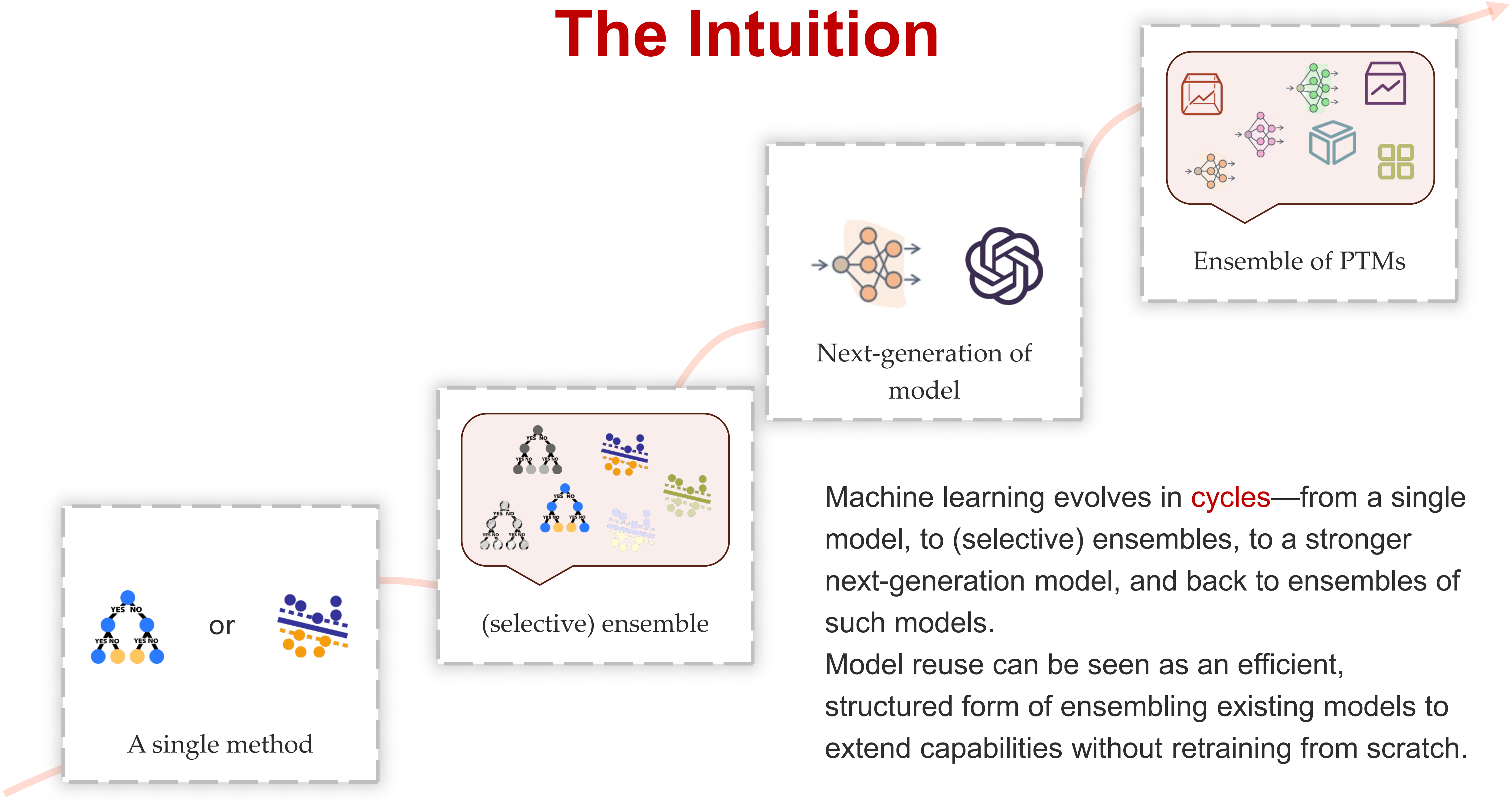
Target Data

Adaptation

Target Data

Management

The Intuition



A single method

or

(selective) ensemble

Next-generation of model

Ensemble of PTMs

Machine learning evolves in cycles—from a single model, to (selective) ensembles, to a stronger next-generation model, and back to ensembles of such models.

Model reuse can be seen as an efficient, structured form of ensembling existing models to extend capabilities without retraining from scratch.

Challenges

Challenges

- Unify fragmented research efforts across different fields

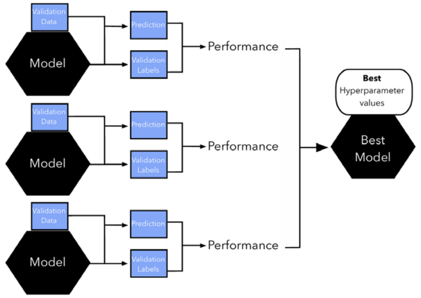
  As model reuse becomes a natural choice in various applications, a holistic and task-agnostic perspective is needed. Establishing a big-picture view of model reuse can bridge the gaps between specialized subfields and promote broader applicability.

- Integrating model reuse across both small and large PTMs

  With the emergence of large PTMs such as foundation models, reusing such models involves both shared principles and diverse implementations. Integrating model reuse strategies can help connect diverse methodologies and inspire the design of novel model reuse strategies.

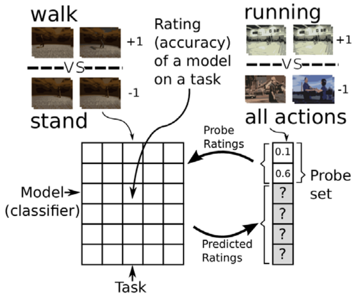
“Synonyms” in PTM Selection

Model Selection



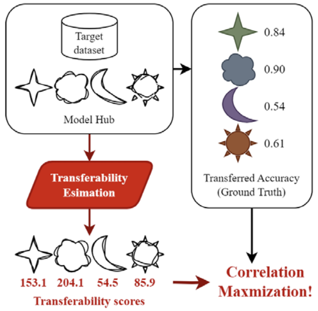
[Raschka et al., CoRR’18]

Model Recommendation



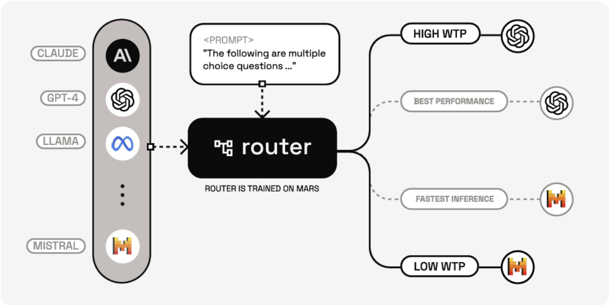
[Matikainen et al., CVPR’12]

Transferability Estimation



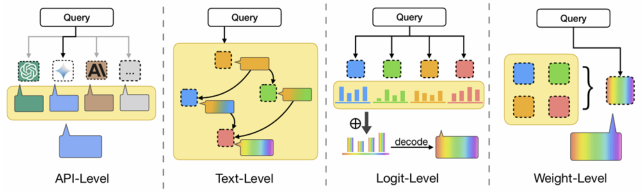
[Ding et al., CoRR’24]

Model Routing



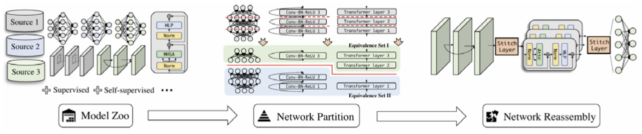
[Hu et al., CORR’24]

Model Collaboration



[Feng et al., CoRR’25]

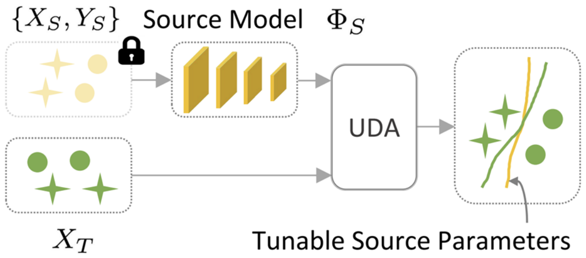
Model Reassembly



[Yang et al., NeurIPS’22]

“Synonyms” in PTM Adaptation

Source-free DA



White-box SFUDA

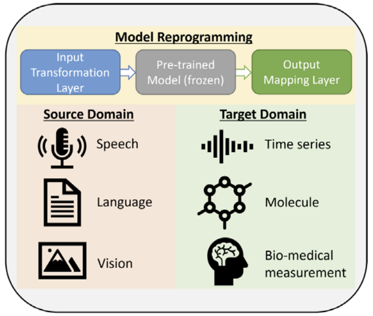
[Liang, Hu, Feng. ICML’20]

Hypothesis Transfer

$$\min_{f_\theta} \sum_{(\boldsymbol{x}_i, y_i)\sim\mathcal{T}} \ell(f_\theta(\boldsymbol{x}_i|g_\Theta), y_i)$$
$$+\lambda\|\theta - \Theta\|_2^2$$

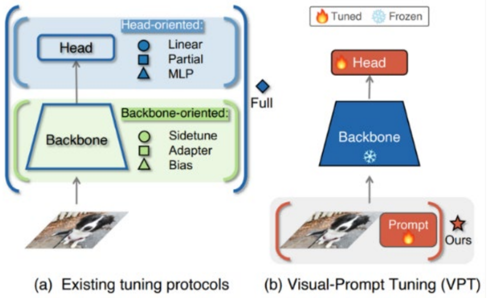
[Kuzborskij & Orabona, ICML’13]

Model Reprogramming



[Chen, AAAI’22]

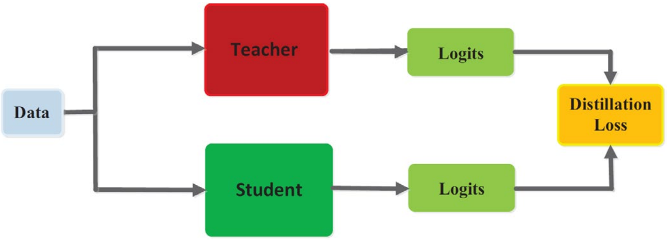
HEFT



(a) Existing tuning protocols (b) Visual-Prompt Tuning (VPT)

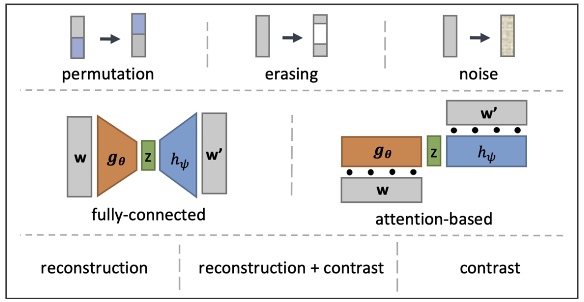
[Jia et al., ECCV’22]

Distillation



[Hinton et al., CoRR’15]

Model Weights as Modalities



[Schürholt et al., NeurIPS’21]

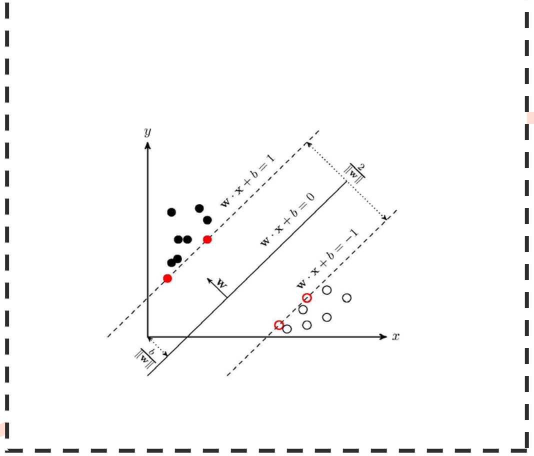
Challenges

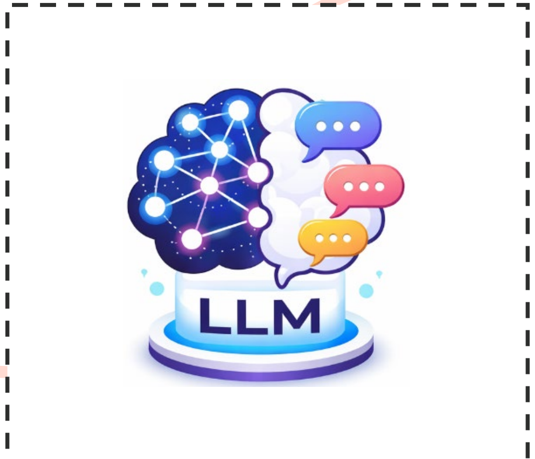
- Unify fragmented research efforts across different fields
  As model reuse becomes a natural choice in various applications, a holistic and task-agnostic perspective is needed. Establishing a big-picture view of model reuse can bridge the gaps between specialized subfields and promote broader applicability.

- Integrating model reuse across both small and large PTMs
  With the emergence of large PTMs such as foundation models, reusing such models involves both shared principles and diverse implementations. Integrating model reuse strategies can help connect diverse methodologies and inspire the design of novel model reuse strategies.

Small and Large PTMs

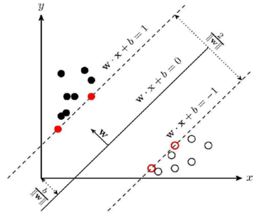
There is no strict boundary between small and large PTMs.



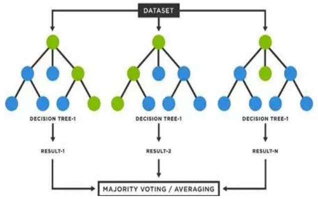


The emergence of large PTMs has significantly impacted the model reuse workflow, as *computational costs* have become a crucial factor, and there are diverse strategies for extracting expert knowledge from these models.

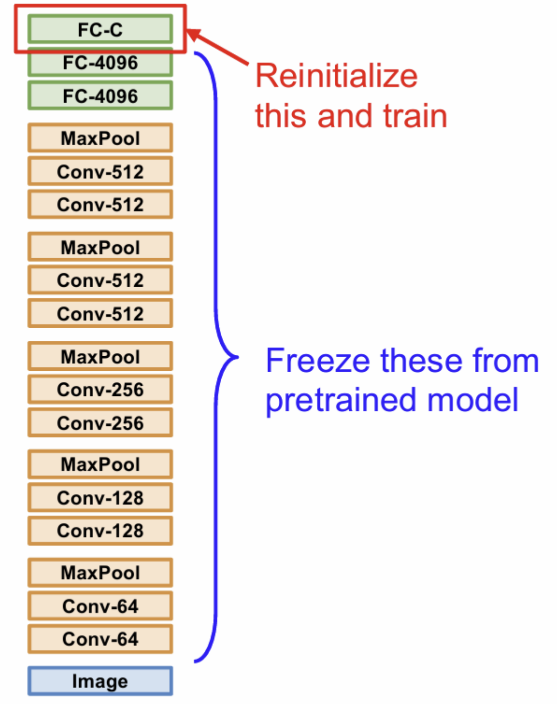
Evolving of the PTM’s Ability



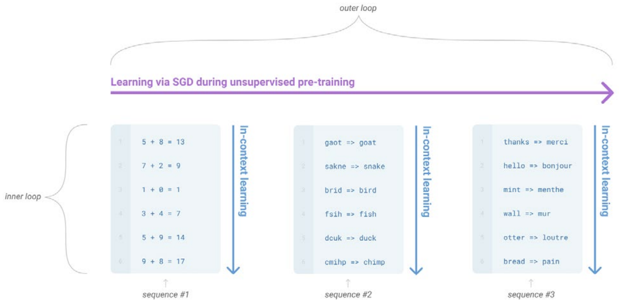
Linear model

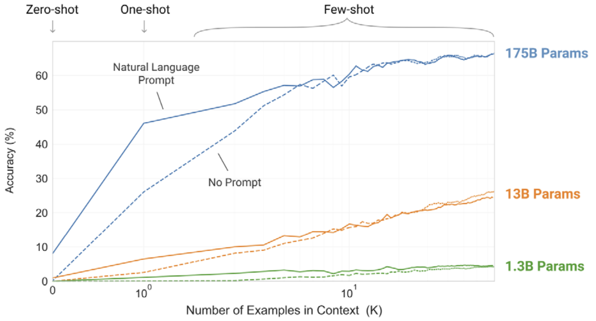


Decision Tree



[Donahue et al., ICML’14]





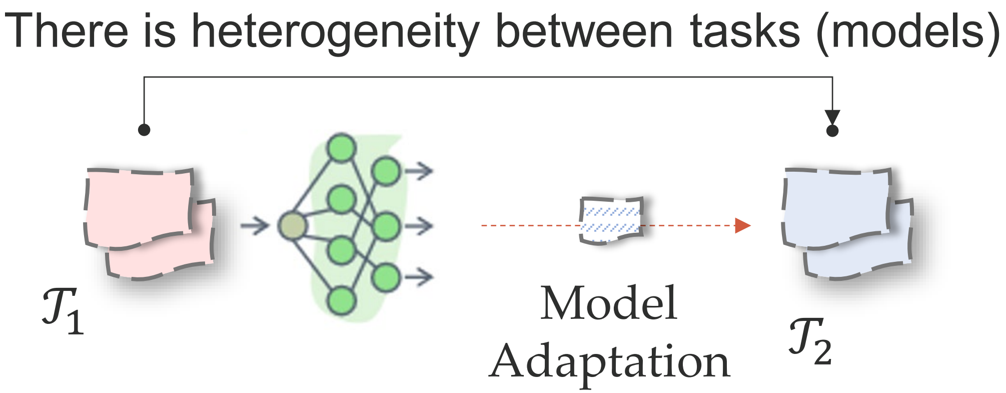
[Brown et al., NeurIPS’20]

Could be applied to new instances from a task

Could be applied to new tasks with slight adaptation

Could be applied to new tasks directly

From Classical PTMs to Foundation PTMs



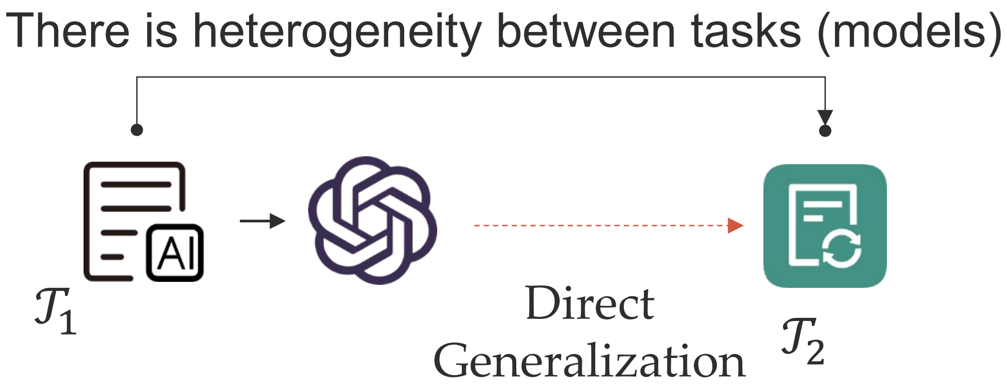
There is heterogeneity between tasks (models)

$\mathcal{T}_1$

Model Adaptation

$\mathcal{T}_2$

w/ an explicit *adaptation* phase



There is heterogeneity between tasks (models)

$\mathcal{T}_1$

Direct Generalization

$\mathcal{T}_2$

w/ *zero-shot generalization* ability

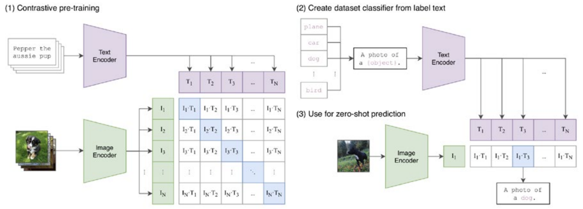
directly applied to downstream tasks

Different (foundation) PTMs with enhanced capabilities.
Decouple the "selection-adaptation" phases of PTM reuse.

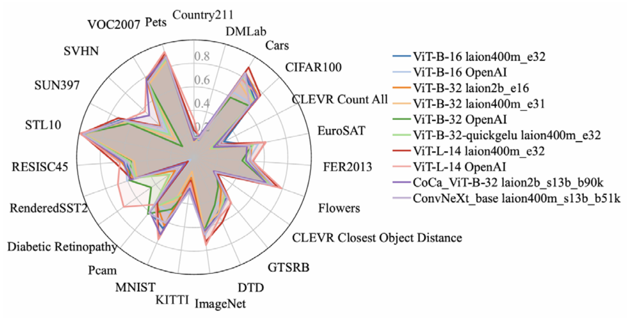
Models with Strong Generalization Ability

Zero-Shot PTMs also demonstrate diverse ability across domains.

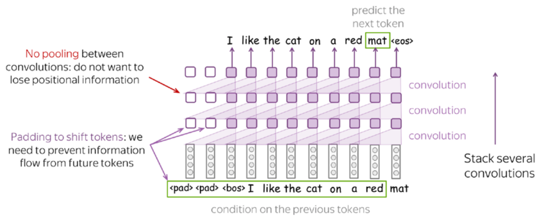
Image-Text Multimodal PTM



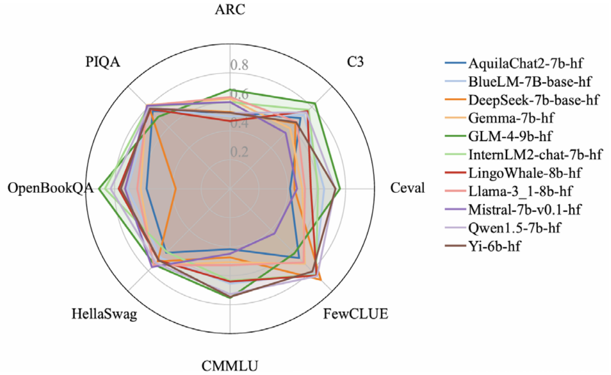
[Radford et al., ICML'21]



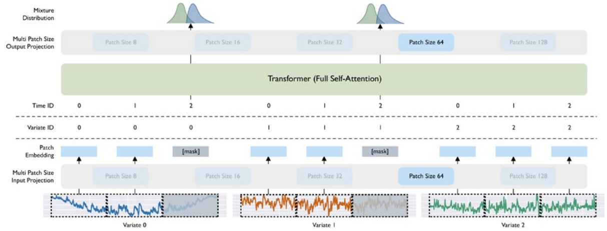
Language Generation PTM


predict the next token
I like the cat on a red mat <eos>
No pooling between convolutions: do not want to lose positional information
convolution
convolution
convolution
Padding to shift tokens: we need to prevent information flow from future tokens
Stack several convolutions
<pad> <pad> <bos> I like the cat on a red mat
condition on the previous tokens

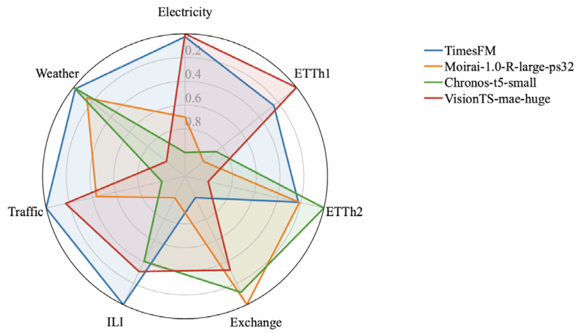
[Brown et al., NeurIPS'20]



Time Series Forecasting PTM



[Ansari et al., TMLR'24]



This tutorial connects the reuse of both types of PTMs, highlighting commonalities and differences to provide deeper insights into model reuse methodologies.

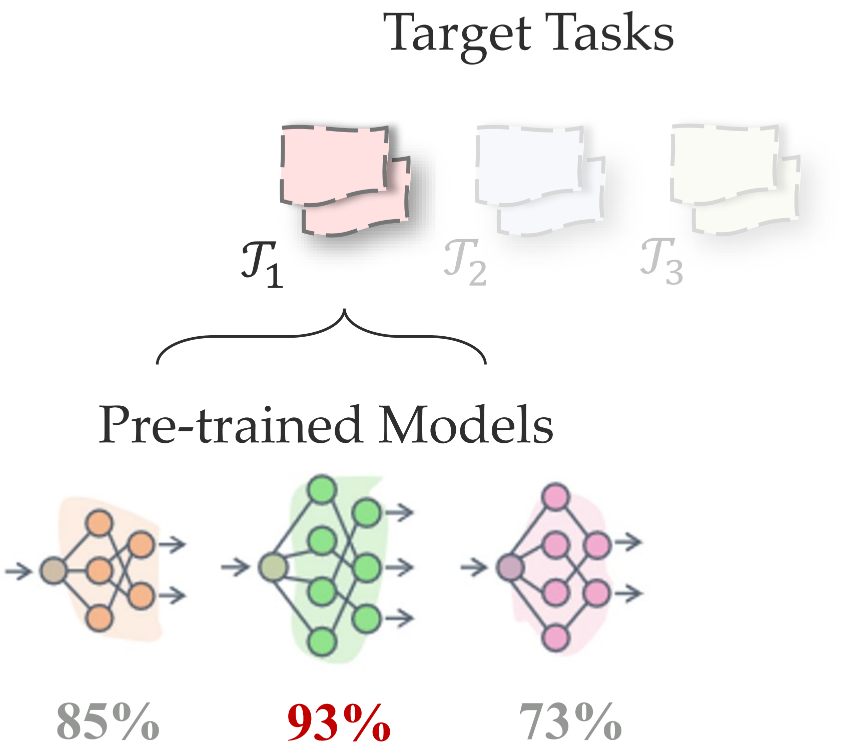
Survey on Model Reuse

- Model Reuse
  - Model Selection
    - Semantic/rule-based methods
      - For small models: Taskonomy [Zamir *et al.*, 2018], Semantic Specification [Zhou and Tan, 2024]
      - For large models: HuggingGPT [Shen *et al.*, 2024], VisProg [Gupta and Kembhavi, 2023]
    - Metric-based methods
      - For small models: GBC [Pándy *et al.*, 2022], NCE [Tran *et al.*, 2019], LEEP [Nguyen *et al.*, 2020], Deep Attribution [Song *et al.*, 2019]
      - For large models: LOVM [Zohar *et al.*, 2023], SWAB [Yi *et al.*, 2024]
    - Learning-based methods
      - For small models: Task2Vec [Achille *et al.*, 2019], MRE [Ding *et al.*, 2022], Model Spider [Zhang *et al.*, 2023]
      - For large models: Mixture of Experts [Jacobs *et al.*, 1991], Model-SAT [Zhang *et al.*, 2025], Zooter [Lu *et al.*, 2024], Routoo [Mohammadshahi *et al.*, 2024],
  - Model Adaptation
    - Adapt PTMs for target data preparation
      - For small models: HMR [Wu *et al.*, 2019], C-LoRA [Smith *et al.*, 2023]
      - For large models: VCD [Menon and Vondrick, 2023], CuPL [Pratt *et al.*, 2023]
    - Adapt PTMs for target model training
      - For small models: ℓ_2 [Li *et al.*, 2018], SN-GANs [Miyato *et al.*, 2018], REFORM [Ye *et al.*, 2021], Knowledge Transfer [Hinton *et al.*, 2015], Twice Learning [Zhou and Jiang, 2004], FitNets [Romero *et al.*, 2015], RKD [Park *et al.*, 2019]
      - For large models: MiniLLM [Gu *et al.*, 2024], LoRA [Hu *et al.*, 2022], Model Reprogramming [Chen, 2024], LLaVA [Liu *et al.*, 2023], BLIP-2 [Li *et al.*, 2023]
    - Adapt PTMs for target model inference
      - For small models: NCM [Mensink *et al.*, 2013], Head2Toe [Evci *et al.*, 2022],VQT [Tu *et al.*, 2023], FisherMerging [Matena and Raffel, 2022], WiSE-FT [Wortsman *et al.*, 2022], OTFusion [Singh and Jaggi, 2020]
      - For large models: RAG [Lewis *et al.*, 2020], CoT [Brown *et al.*, 2020]
  - Other Topics
    - Model Assembly [Yang *et al.*, 2022], Model Representation Learning [Schürholt *et al.*, 2022], Model Editing [Mitchell *et al.*, 2022], Model Repair [Luo *et al.*, 2023], Model Relationship Discovery [Yu and Wang, 2024], Managing LLMs [Shen *et al.*, 2024]

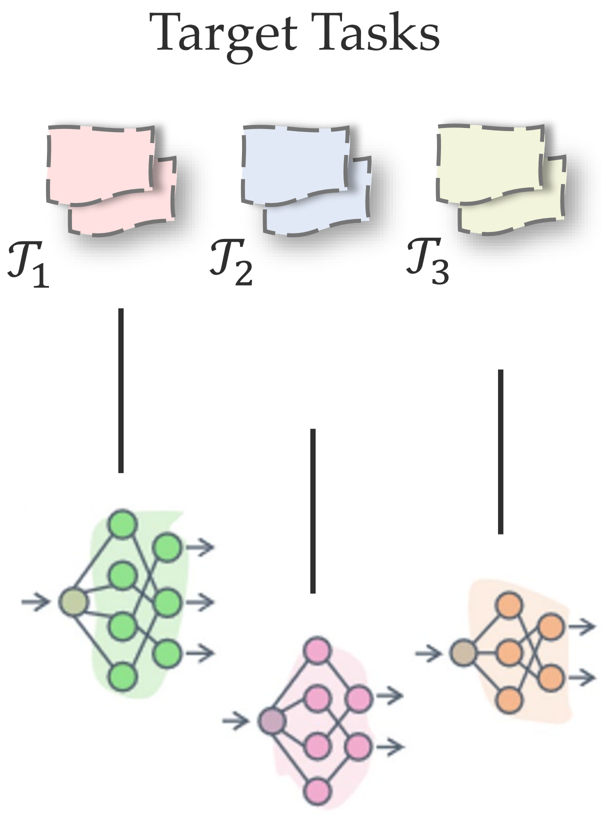
Da-Wei Zhou, Han-Jia Ye. *A Unifying Perspective on Model Reuse: From Small to Large Pre-Trained Models*. IJCAI 2025.

Part 1: PTM Selection

PTM Selection



Fine-tuned performance

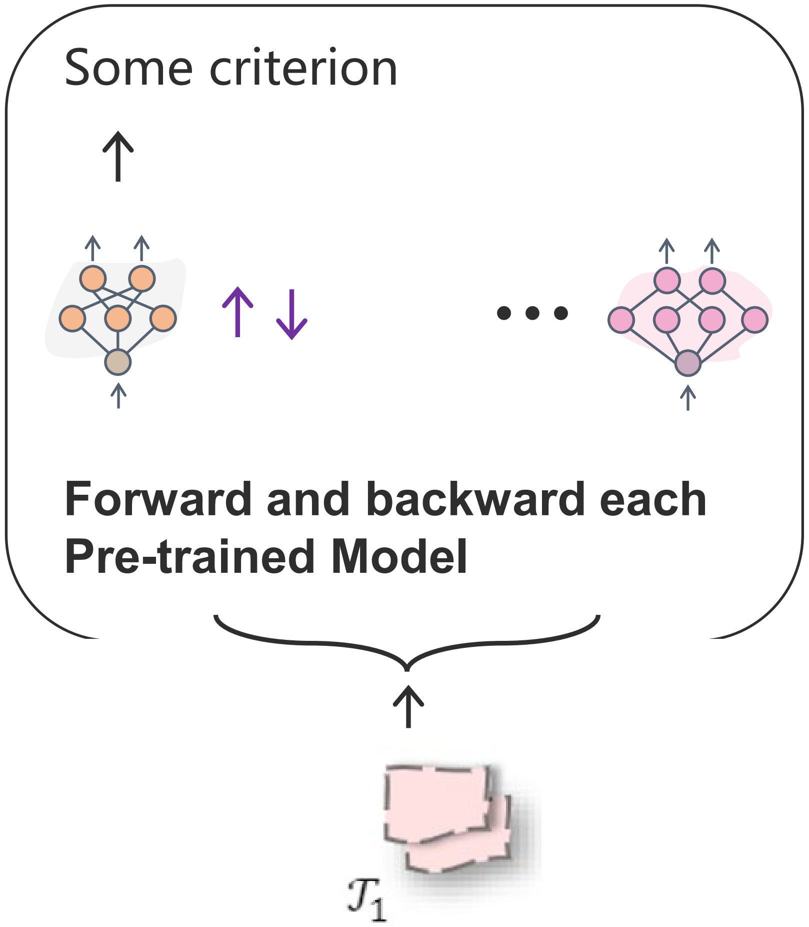


selecting **suitable** pre-trained models for target tasks

Previous Methods

PTM Selection

The vanilla strategy



Forward and backward each Pre-trained Model

To select on M pre-trained models

1 task, but $\boldsymbol{M}$ **forwards** and **backwards**

Select PTMs Efficiently

Taxonomy of PTM Selection

The target of model selection is to choose one or multiple related PTMs from the model zoo that are suitable for the related dataset. This is a typical *retrieval* process, where the core problem is to *rank* all models in the model zoo correctly.

Based on how to calculate the matching degree, there are three main types of methods

Semantic/rule-based methods

*Semantic/rule-based methods are designed to utilize the **semantic information** (e.g., natural language descriptions) or **manually designed rules** to find related models.*

Metric-based methods

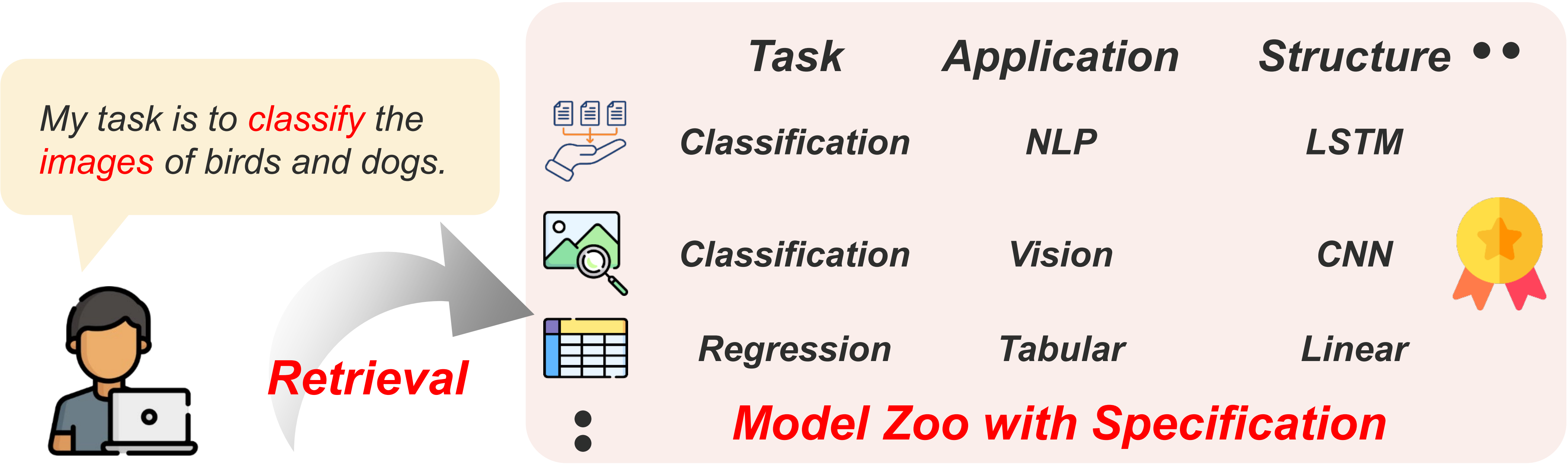
*Metric-based methods rely on defining the **matching degree** between the model's output and the corresponding dataset.*

Learning-based methods

*Learning-based methods directly **learn** the similarity between data and models.*

Semantic/Rule-Based Methods

Main idea: directly describe the task or define the rule of model selection



My task is to classify the images of birds and dogs.

Retrieval

	Task	*Application*	*Structure*
	Classification	***NLP***	***LSTM***
	Classification	***Vision***	***CNN***
	Regression	***Tabular***	***Linear***

Model Zoo with Specification

This semantic information is commonly represented by natural languages, expressing the capability of models and the upstream training data.

Semantic/Rule-Based Methods

PTMs in the model zoo are accompanied by descriptions (known as “semantic specification” or “statistical specification”). When facing a new task, the users are only required to describe their requirements as a new query and search within the model zoo.

The semantic specification contains descriptive information related to the model, including the *target task* (*e.g.*, classification or regression), *the kind of machine learning applications* (*e.g.*, natural language processing or image recognition), the *model structure* (*e.g.*, SVM, CNN, or Decision Tree), etc.

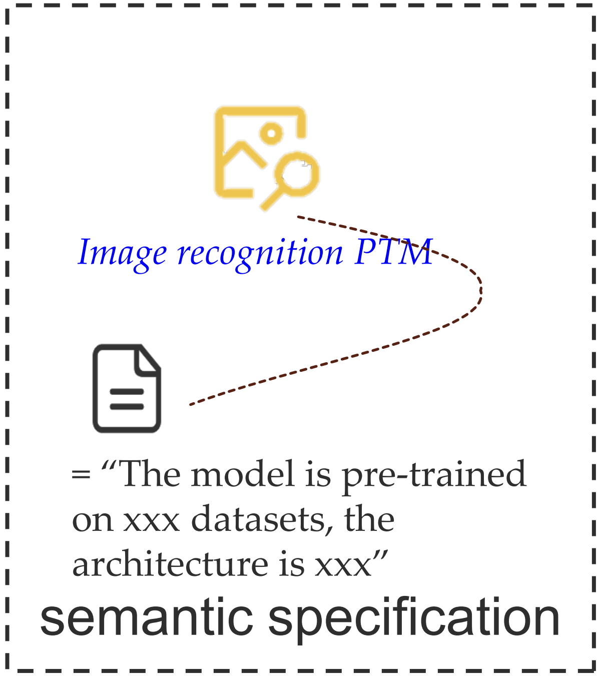


Image recognition PTM

= “The model is pre-trained on xxx datasets, the architecture is xxx”

semantic specification

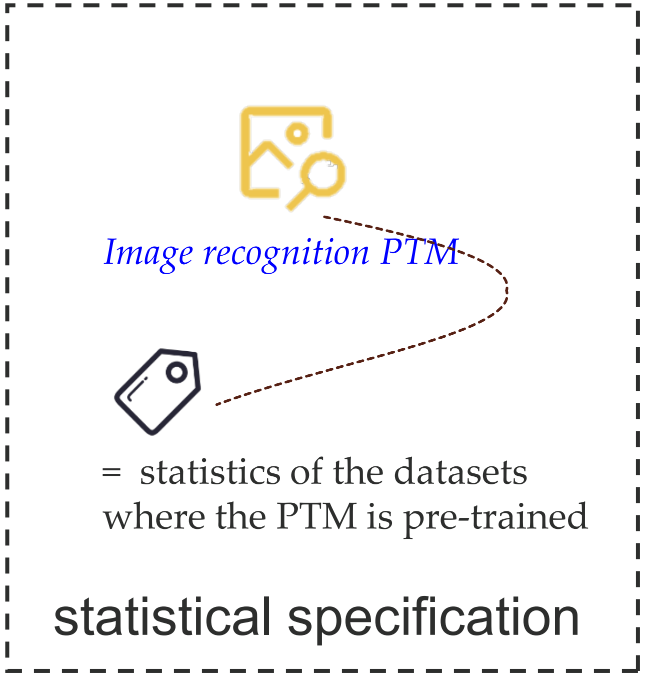


Image recognition PTM

= statistics of the datasets where the PTM is pre-trained

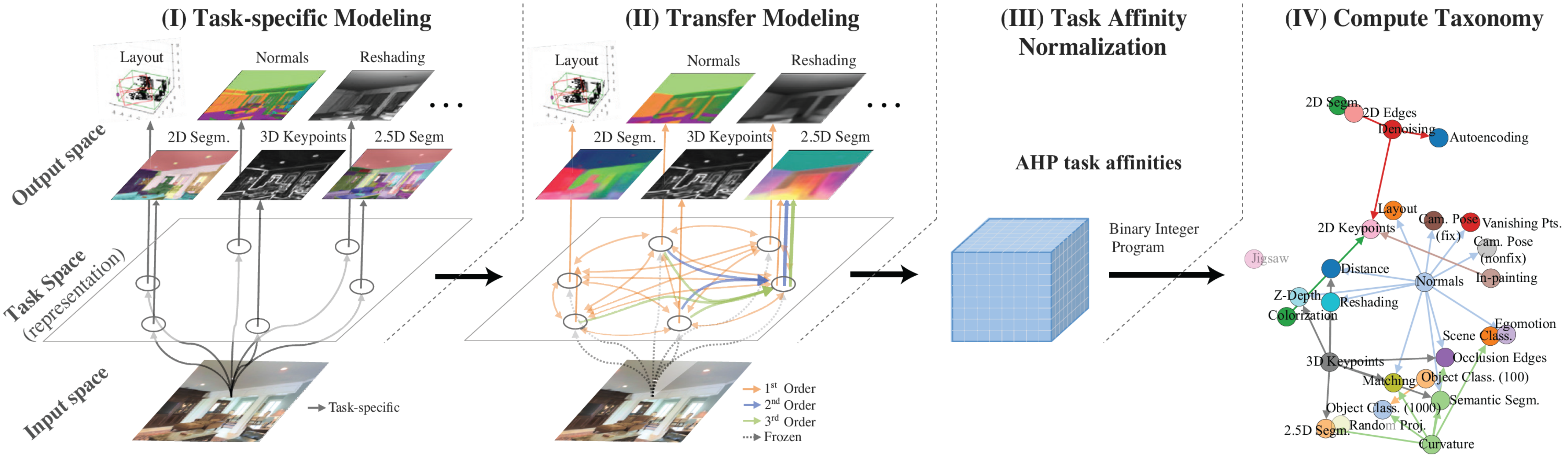
statistical specification

Zhi-Hua Zhou, Zhi-Hao Tan. *Learnware: small models do big*. Science China Information Sciences. 2024.

Semantic/Rule-Based Methods

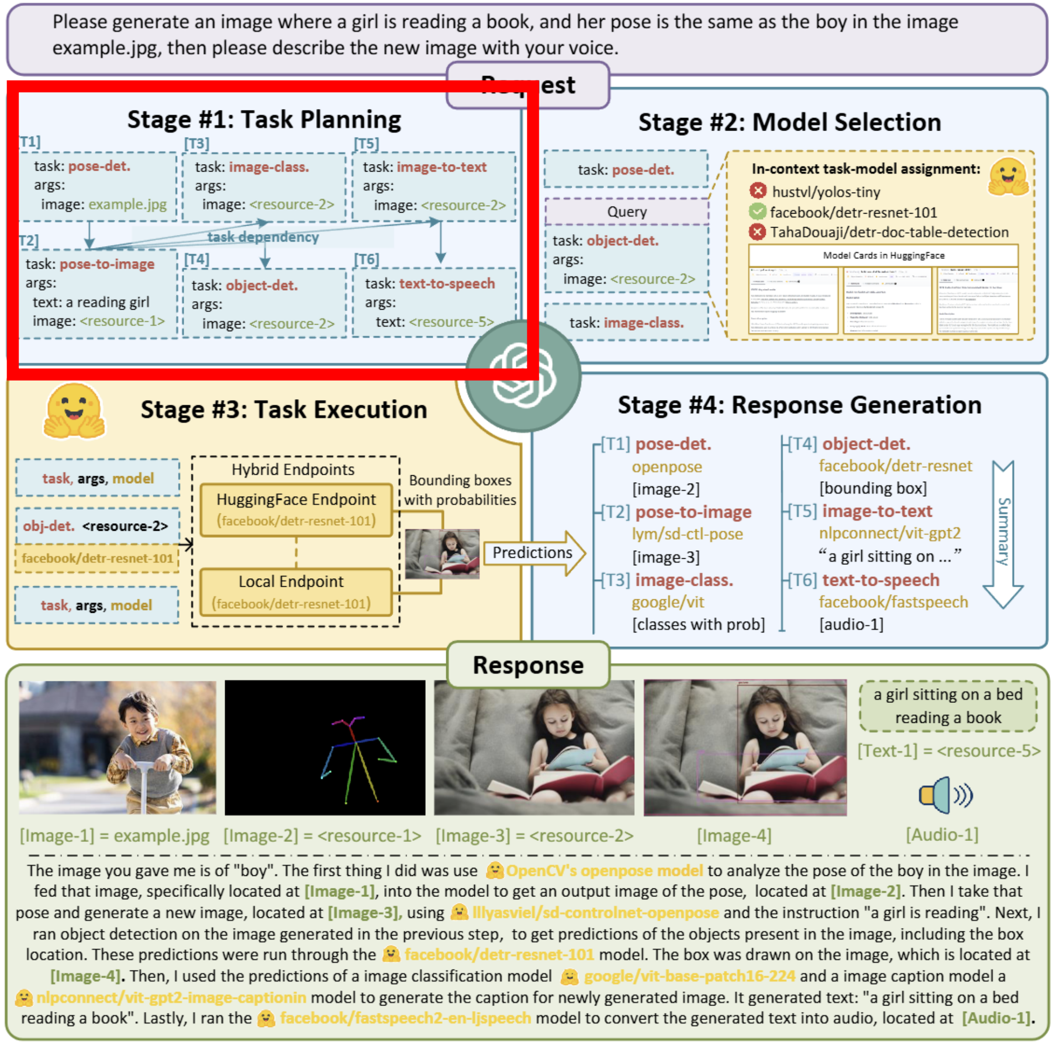
Taskonomy proposes a fully computational approach to discover and model the underlying structure of the space of visual tasks.

Reuse the related source encoder based on the constructed Taskonomy. Attach a small, shallow transfer network on top of the frozen source encoder.



Amir Zamir et al., *Taskonomy: Disentangling Task Transfer Learning.* CVPR 2018.

Semantic/Rule-Based Methods



Please generate an image where a girl is reading a book, and her pose is the same as the boy in the image example.jpg, then please describe the new image with your voice.

Request
Stage #1: Task Planning
[T1] task: pose-det. args: image: example.jpg
[T3] task: image-class. args: image: <resource-2>
[T5] task: image-to-text args: image: <resource-2>
task dependency
[T2] task: pose-to-image args: text: a reading girl image: <resource-1>
[T4] task: object-det. args: image: <resource-2>
[T6] task: text-to-speech args: text: <resource-5>
Stage #2: Model Selection
task: pose-det.
Query
task: object-det. args: image: <resource-2>
task: image-class.
In-context task-model assignment:
hustvl/yolos-tiny
facebook/detr-resnet-101
TahaDouaji/detr-doc-table-detection
Model Cards in HuggingFace
Stage #3: Task Execution
task, args, model
obj-det. <resource-2>
facebook/detr-resnet-101
task, args, model
Hybrid Endpoints
HuggingFace Endpoint (facebook/detr-resnet-101)
Local Endpoint (facebook/detr-resnet-101)
Bounding boxes with probabilities
Predictions
Stage #4: Response Generation
[T1] pose-det. openpose [image-2]
[T2] pose-to-image lym/sd-ctl-pose [image-3]
[T3] image-class. google/vit [classes with prob]
[T4] object-det. facebook/detr-resnet [bounding box]
[T5] image-to-text nlpconnect/vit-gpt2 “a girl sitting on ...”
[T6] text-to-speech facebook/fastspeech [audio-1]
Summary
Response
a girl sitting on a bed reading a book
[Text-1] = <resource-5>
[Image-1] = example.jpg
[Image-2] = <resource-1>
[Image-3] = <resource-2>
[Image-4]
[Audio-1]

The image you gave me is of "boy". The first thing I did was use OpenCV's openpose model to analyze the pose of the boy in the image. I fed that image, specifically located at [Image-1], into the model to get an output image of the pose, located at [Image-2]. Then I take that pose and generate a new image, located at [Image-3], using lllyasviel/sd-controlnet-openpose and the instruction "a girl is reading". Next, I ran object detection on the image generated in the previous step, to get predictions of the objects present in the image, including the box location. These predictions were run through the facebook/detr-resnet-101 model. The box was drawn on the image, which is located at [Image-4]. Then, I used the predictions of a image classification model google/vit-base-patch16-224 and a image caption model a nlpconnect/vit-gpt2-image-captionin model to generate the caption for newly generated image. It generated text: "a girl sitting on a bed reading a book". Lastly, I ran the facebook/fastspeech2-en-ljspeech model to convert the generated text into audio, located at [Audio-1].

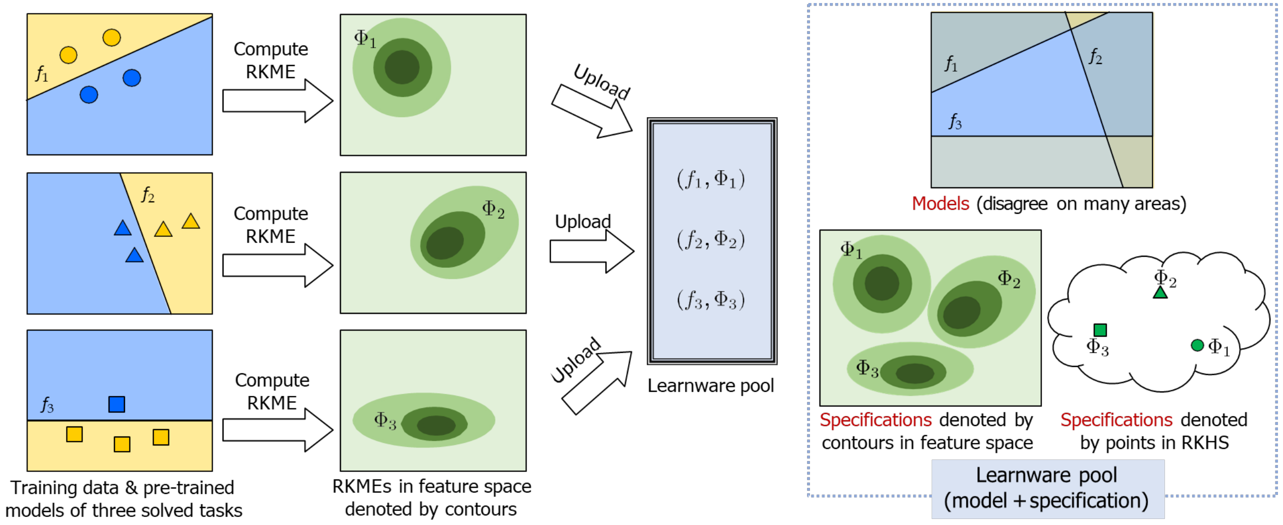
HuggingGPT utilizes GPT to manage millions of models on Hugging-Face. Natural language is utilized as the interface to align all models. HuggingGPT involves four core steps:

- Task Planning: Using GPT to analyze the intention
- Model Selection: Ask GPT to select the most suitable model on HuggingFace
- Task Execution: Invoke and execute selected models
- Response Generation: Using GPT to integrate predictions of all models

Yongliang Shen, et al. *HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face*. NeurIPS 2024.

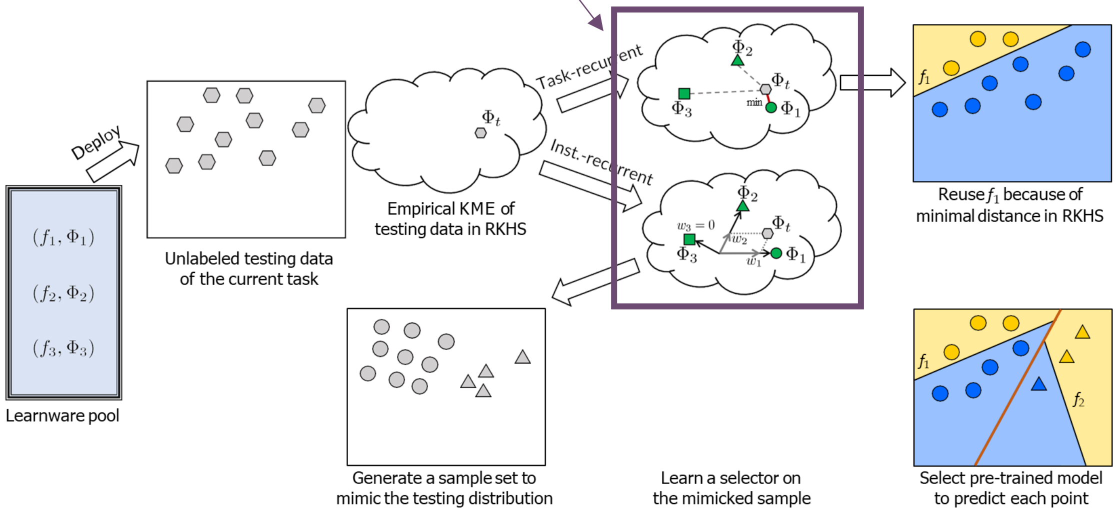
Statistical Specification

Use RKME to generate statistical specification on the datasets where the PTM is pre-trained. RKME guarantees the privacy issue and helps the selection of the PTM in the deployment stage.


f_1
Compute RKME
Φ_1
Upload
f_2
Compute RKME
Φ_2
Upload
f_3
Compute RKME
Φ_3
Upload
Training data & pre-trained models of three solved tasks
RKMEs in feature space denoted by contours
(f_1, Φ_1)
(f_2, Φ_2)
(f_3, Φ_3)
Learnware pool
f_1
f_2
f_3
Models (disagree on many areas)
Φ_1
Φ_2
Φ_3
Specifications denoted by contours in feature space
Φ_2
Φ_3
Φ_1
Specifications denoted by points in RKHS
Learnware pool (model + specification)

upload phase: model specification construction

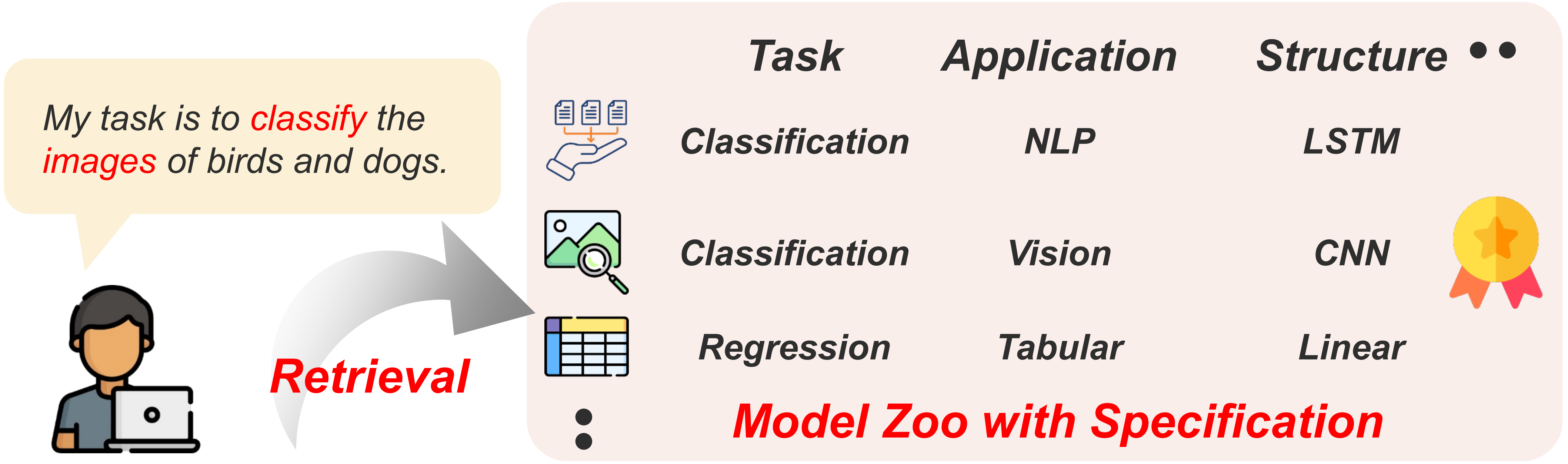
choose the model with minimum kernel-space distance


(f_1, Φ_1)
(f_2, Φ_2)
(f_3, Φ_3)
Learnware pool
Deploy
Unlabeled testing data of the current task
Φ_t
Empirical KME of testing data in RKHS
Task-recurrent
Φ_2
Φ_t
Φ_3
min
Φ_1
Inst.-recurrent
Φ_2
$w_3 = 0$
w_2
Φ_t
Φ_3
w_1
Φ_1
f_1
Reuse f_1 because of minimal distance in RKHS
Generate a sample set to mimic the testing distribution
Learn a selector on the mimicked sample
f_1
f_2
Select pre-trained model to predict each point

deploy phase: model selection based on specification

Xi-Zhu Wu, Wenkai Xu, Song Liu, Zhi-Hua Zhou. *Model Reuse with Reduced Kernel Mean Embedding Specification*. TKDE. 2023.
Peng Tan, Zhi-Hao Tan, Yuan Jiang, Zhi-Hua Zhou. Handling Learnwares Developed from Heterogeneous Feature Spaces without Auxiliary Data. IJCAI 2023.
Jian-Dong Liu, Zhi-Hao Tan, Zhi-Hua Zhou. Towards Making Learnware Specification and Market Evolvable. AAAI 2024.

Discussion: Semantic/Rule-Based Methods



My task is to classify the images of birds and dogs.

Retrieval

	Task	***Application***	***Structure***
	Classification	***NLP***	***LSTM***
	Classification	***Vision***	***CNN***
	Regression	***Tabular***	***Linear***

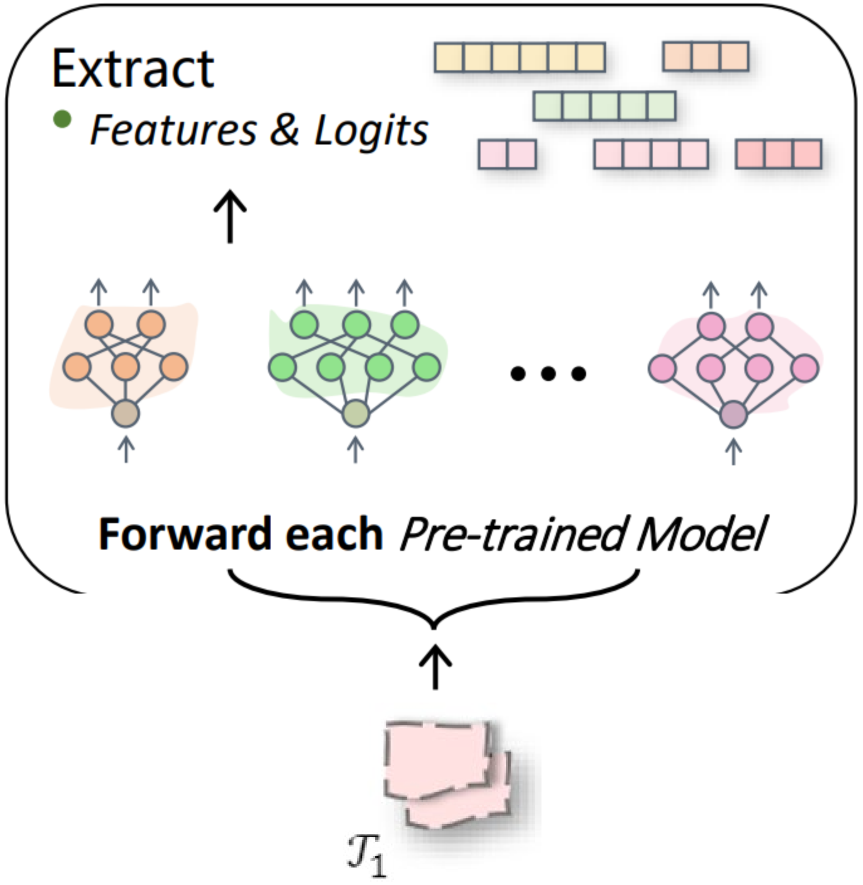
Model Zoo with Specification

- Can be easily adapted to various domains and tasks by simply tagging models with appropriate semantic labels.
- This flexibility makes them suitable for dynamic environments where new types of tasks frequently emerge.

- The effectiveness heavily relies on the accuracy and comprehensiveness of the semantic specifications provided. Inaccurate or vague descriptions can lead to poor performance.
- Manual semantic tagging becomes impractical as the number of models grows.

Metric-Based Methods

Main idea: define the metric to represent the *fitness between models and datasets*. Also known as "transferability estimation".



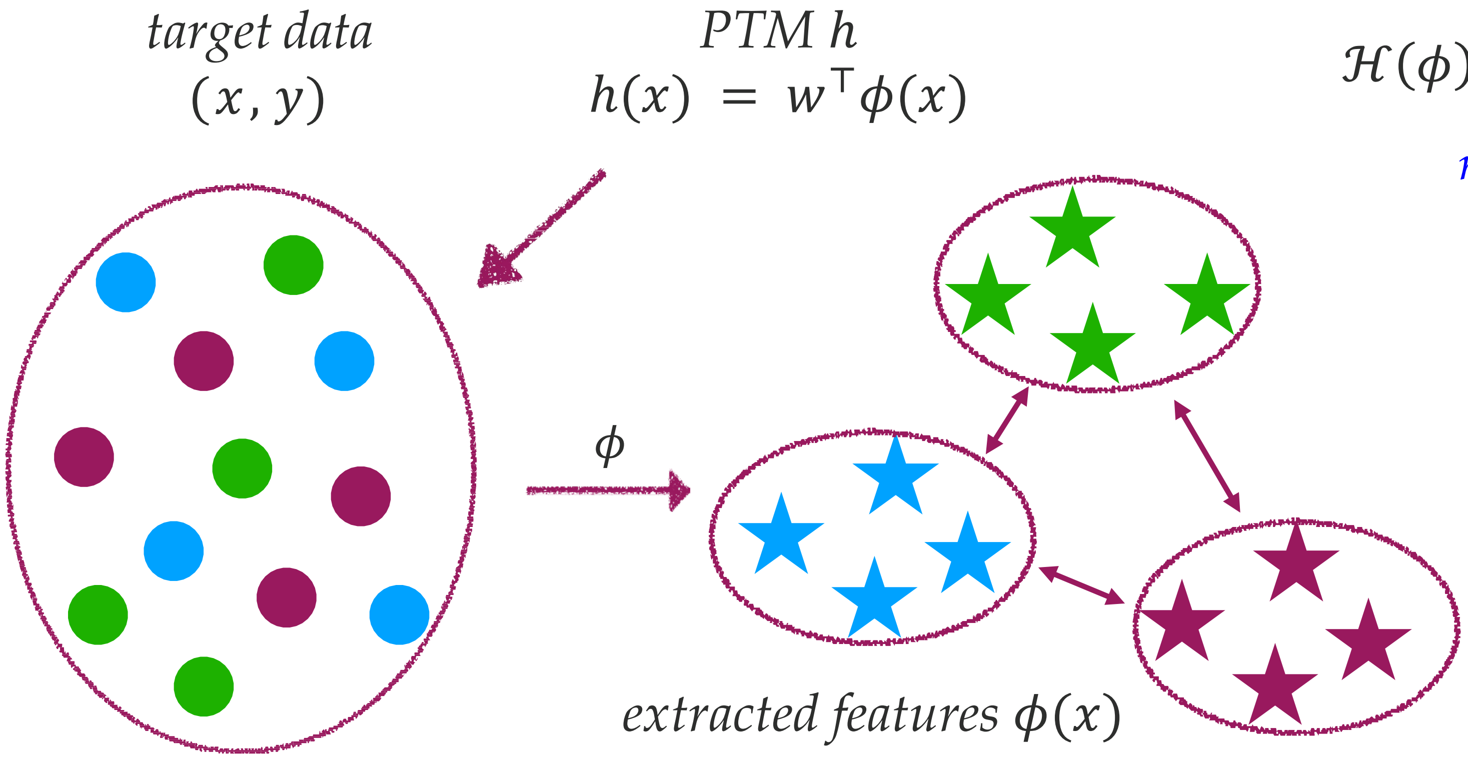
Reduce the burden of multiple backwards to one or more forwards.

LEEP [Nguyen et al., ICML'20], LogME [You et al. ICML'21], GBC [Pándy et al. CVPR'22], etc.

Yuhe Ding et al., *Which Model to Transfer? A Survey on Transferability Estimation*. CoRR 2024.
Yihao Xue et al., *A Review on Transferability Estimation in Deep Transfer Learning*. TAI 2024.

H-Score

H-score estimates the transferability of learned representations from a source task to a target task. The score characterizes the *discriminative power of the features for the target labels*.



$$\mathcal{H}(\phi) = \operatorname{tr}\left(\operatorname{cov}(\phi(x))^{-1} \operatorname{cov}\left(\mathrm{E}_{p(x|y)}[\phi(x) \mid y]\right)\right)$$

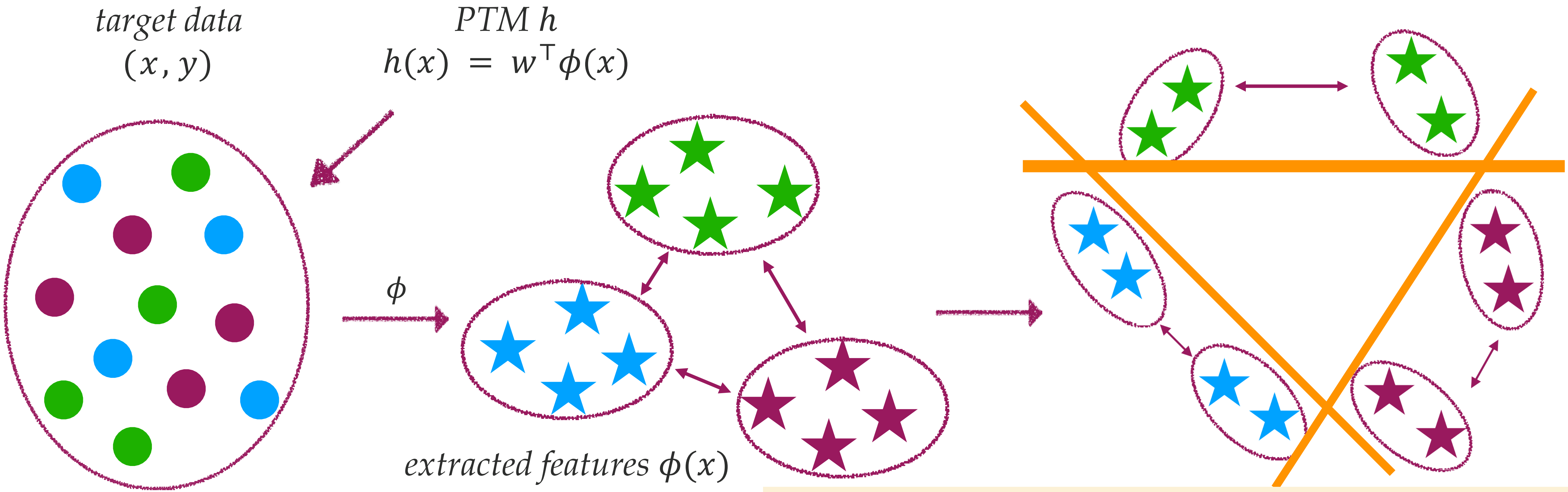
regularization of global feature variance   *inter-class feature variance*

- If the features have large distances among classes, meanwhile be similar within the same class, then ϕ preserves classification information.
- Directly utilize ϕ in calculation, thus H-score also works for unsupervised PTMs.

Yajie Bao et al., *An Information-Theoretic Approach to Transferability in Task Transfer Learning*. ICIP 2019.

H-Score

H-score estimates the transferability of learned representations from a source task to a target task. The score characterizes the *discriminative power of the features for the target labels*.



The intra-class feature variance needs not to be small.

The re-trained prediction head can handle this.

NCE

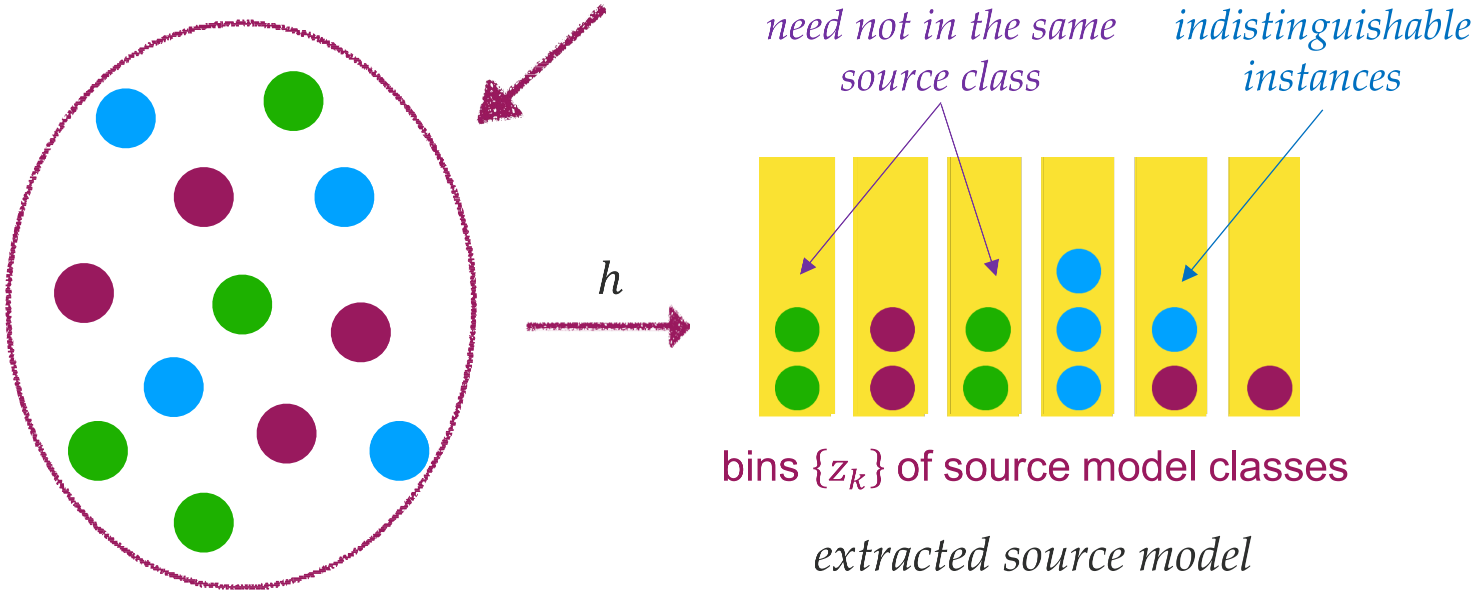
Negative conditional entropy (NCE) between label sequences of source and target tasks reflects the information required to predict labels of one task based on the knowledge of another, thereby providing an estimate of task transferability and hardness.

target data
(x, y)

PTM h
$h(x) = w^\top \phi(x)$

$$NCE(h) = -H(y \mid z) = -\sum_{y\in\mathcal{Y}, z\in\mathcal{Z}} \hat{p}(y,z)\log[\hat{p}(y \mid z)]$$

the purity of source class bin z



need not in the same source class

indistinguishable instances

bins $\{z_k\}$ of source model classes

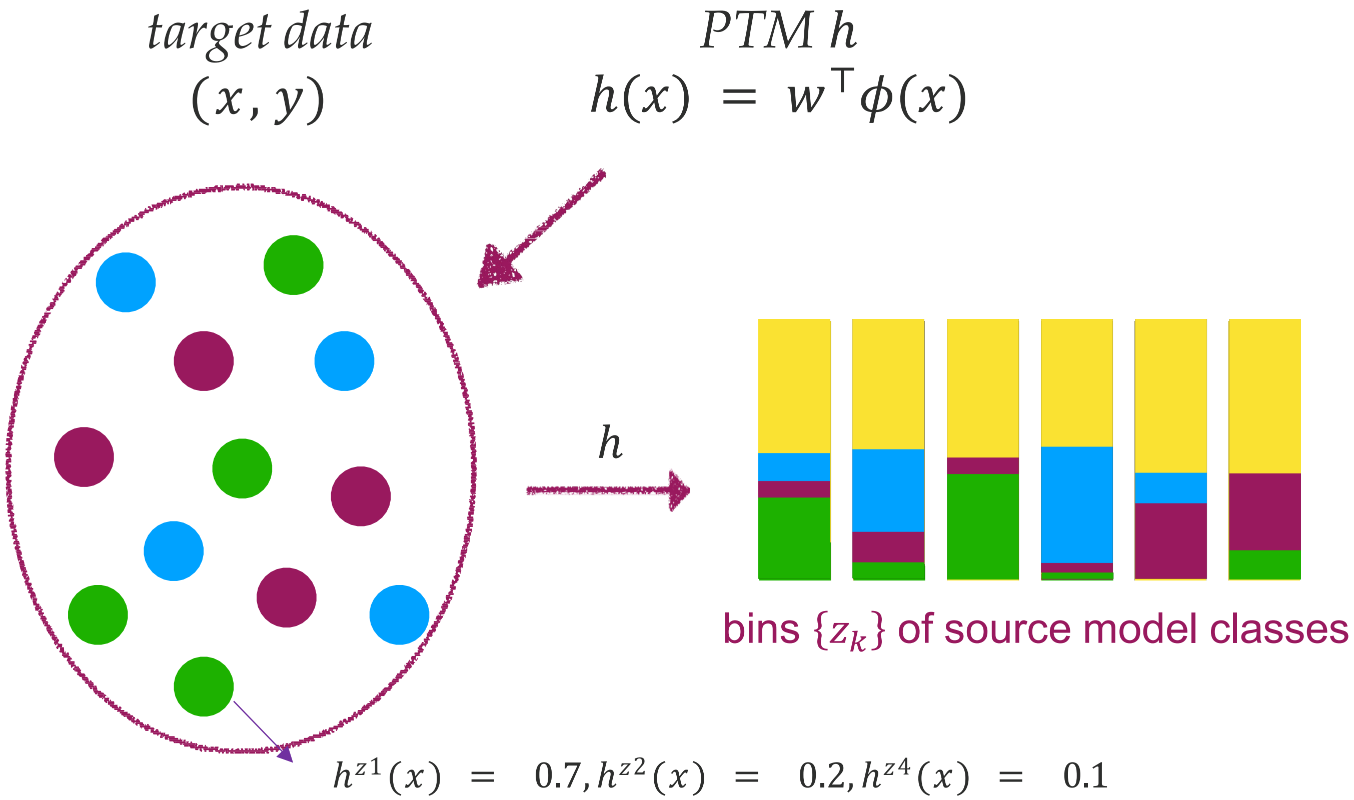
extracted source model prediction $h(x)$

- If the purity of the source model classes is high, which indicates that the features are sufficiently distinguishable, then ϕ preserves classify information.
- Also measures the mutual information between y and z since $I(y; z) = H(y) - H(y \mid z)$ and $H(y)$ isn't affected by the source model.

Anh T. Tran, Cuong V. Nguyen, Tal Hassner. *Transferability and Hardness of Supervised Classification Tasks*. ICCV 2019.

LEEP

Log Expected Empirical Prediction (LEEP) can be treated as a soft version of NCE, which utilize source prediction probabilities instead of hard labels.



target data
(x, y)

PTM h
$h(x) = w^\top \phi(x)$

h

bins $\{z_k\}$ of source model classes

$h^{z1}(x) = 0.7, h^{z2}(x) = 0.2, h^{z4}(x) = 0.1$

soft source label prediction

$$LEEP(h) = \frac{1}{N}\sum_{i=1}^{N} \log\left(\sum_{z \in \mathcal{Z}} \hat{p}(y_i \mid z) p(z \mid x_i)\right)$$

the purity of
source class bin z

$$NCE(h) = -H(y \mid z) = -\sum_{y \in \mathcal{Y}, z \in \mathcal{Z}} \hat{p}(y, z) \log[\hat{p}(y \mid z)]$$

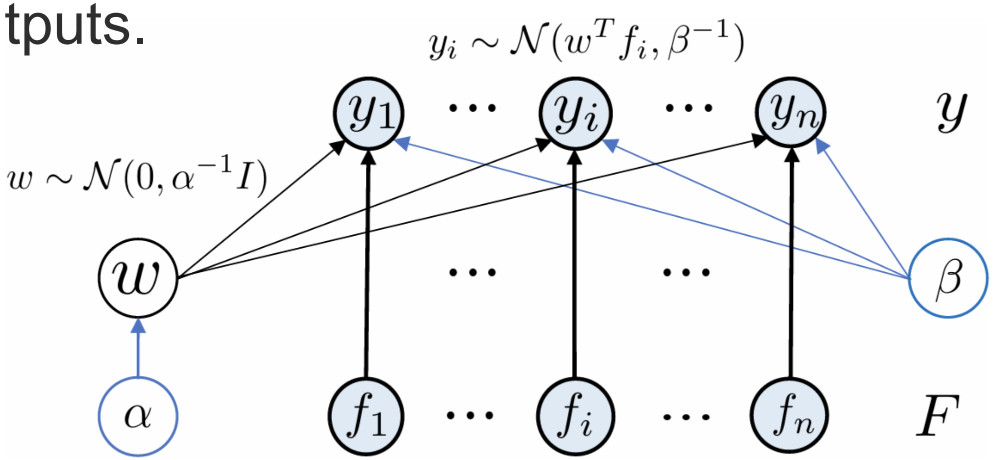
the purity of source class bin z

- The metric is based on the alignment of the model's predictions with the target task's requirements
- The soft probabilities may contain information of prediction confidence. The lower the confidence, the worse the transferability.

Cuong V. Nguyen, Tal Hassner, Matthias Seeger, Cedric Archambeau. *LEEP: A New Measure to Evaluate Transferability of Learned Representations*. ICML 2020.

LogME

LogME (logarithm of maximum evidence) measure correlation between features and outputs.


$y_i \sim \mathcal{N}(w^T f_i, \beta^{-1})$
y_1
y_i
y_n
y
$w \sim \mathcal{N}(0, \alpha^{-1} I)$
w
β
α
f_1
f_i
f_n
F

- $F \in \mathbb{R}^N \times d$ is the extracted features on the target task

$$\mathcal{L}(\alpha, \beta) = \log p\,(y|F, \alpha, \beta)$$
$$= \frac{n}{2}\log\beta + \frac{d}{2}\log\alpha - \frac{n}{2}\log 2\,\pi$$
$$-\frac{\beta}{2}\|Fm - y\| - \frac{\alpha}{2}m^T m - \frac{1}{2}\log|A|$$
$$A = \alpha I + \beta F^T F, \qquad m = \beta A^{-1} F^T y$$

Ignore the prior distributions, the objective can be simplified to

maximize feature correlation

$$\log p\,(y \mid F) = \frac{-\log|A| - y^T y + \left(F^T y\right)^T A^{-1}\left(F^T y\right)}{2}$$

$$A = F^T F$$ *minimize feature variance*

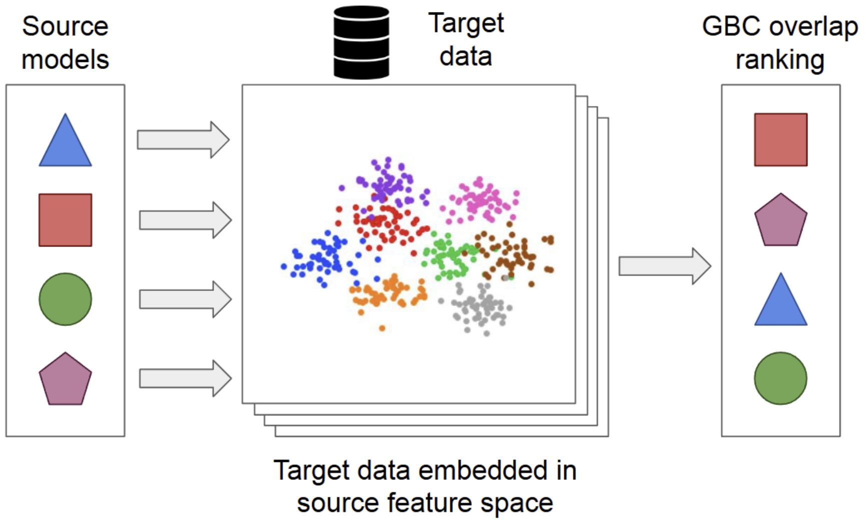
- Need to solve an optimization problem on target data. The computation cost is higher.
- Share the similar principle of H-score. Introducing the prior distribution parameters can improve generalization.
- Work for both supervised and unsupervised PTMs. Restrict to be applied on target tasks with real-valued output space.

Kaichao You, Yong Liu, Jianmin Wang, Mingsheng Long. *LogME: Practical Assessment of Pre-trained Models for Transfer Learning*. ICML 2021.

Metric-Based Methods

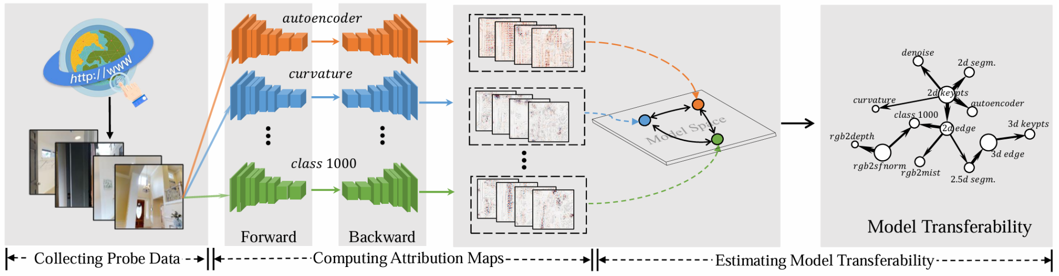
Design metrics from different aspects.

GBC measures the overlap ratio among different classes in the source model embedding space



Utilize the PTM to encode the downstream task's data, and evaluate the separability of per-class instances considering inter-class and intra-class information.

Utilize the matching degree of attribution maps to define the metric.

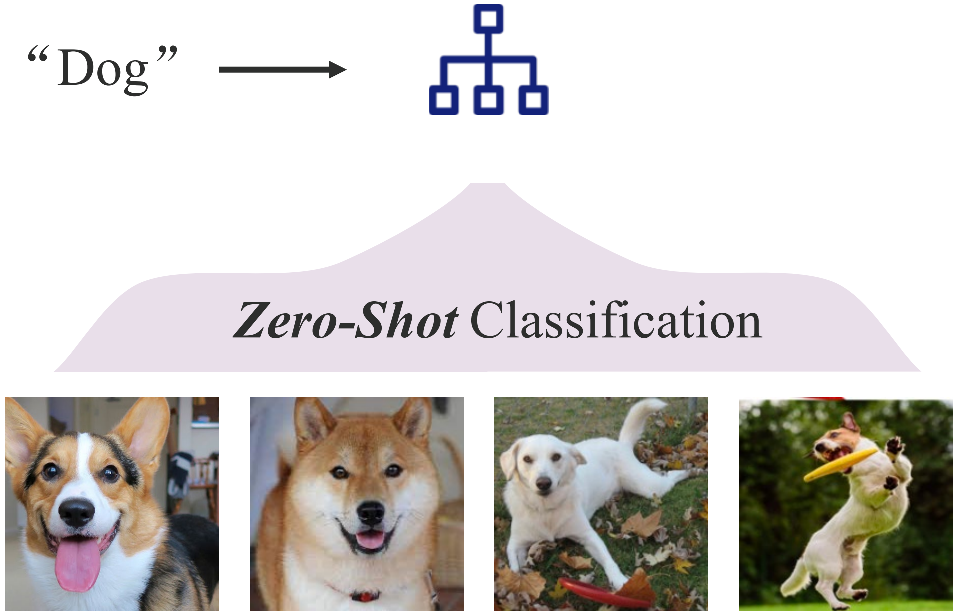


$$d(m_i, m_j) = \frac{N_p}{\sum_{k=1}^{N_p} \cos\_\mathrm{sim}\left(A_k^i, A_k^j\right)}$$

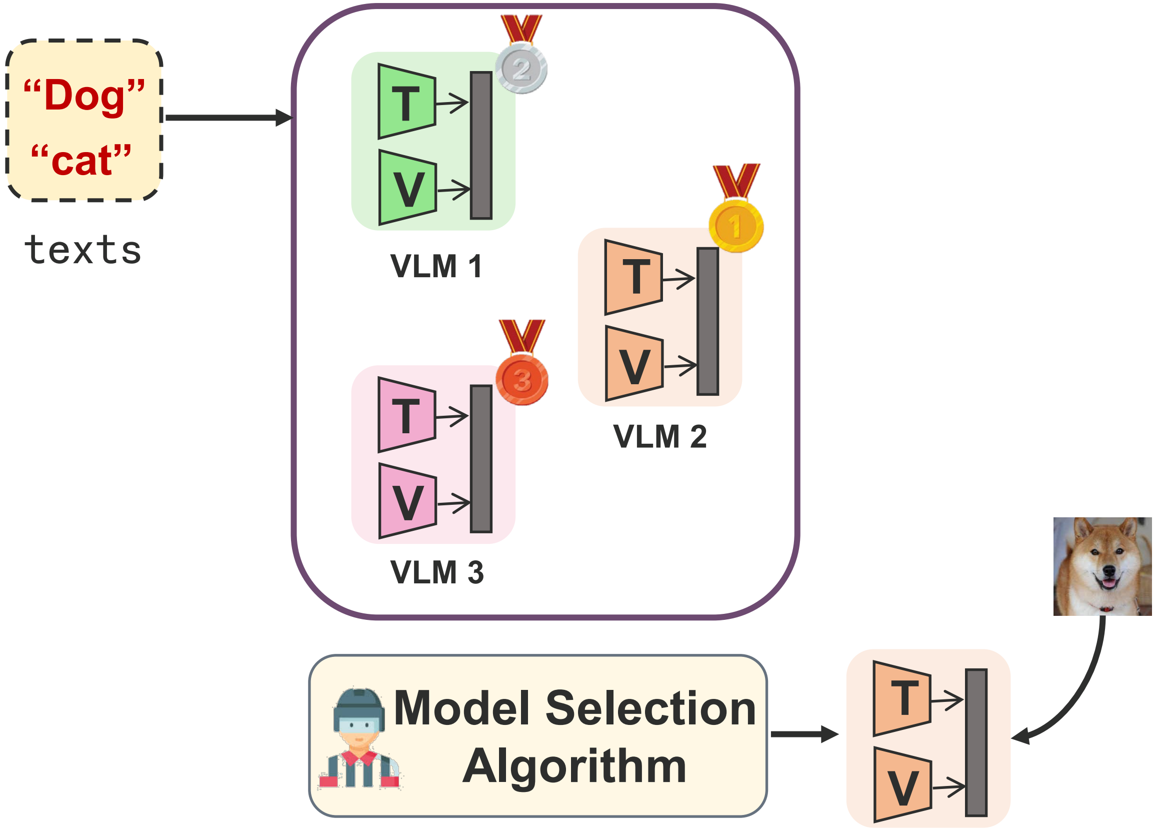
Calculate the pairwise distance between models based on the similarity of their attribution maps
Identify the source model with the smallest distance to the target in this model space.

Michal Pándy et al., *Transferability estimation using Bhattacharyya class separability*. CVPR 2022.
Song et al., *Deep Model Transferability from Attribution Maps*. NeurIPS 2019.

Selection of Vision-Language Model (VLM)



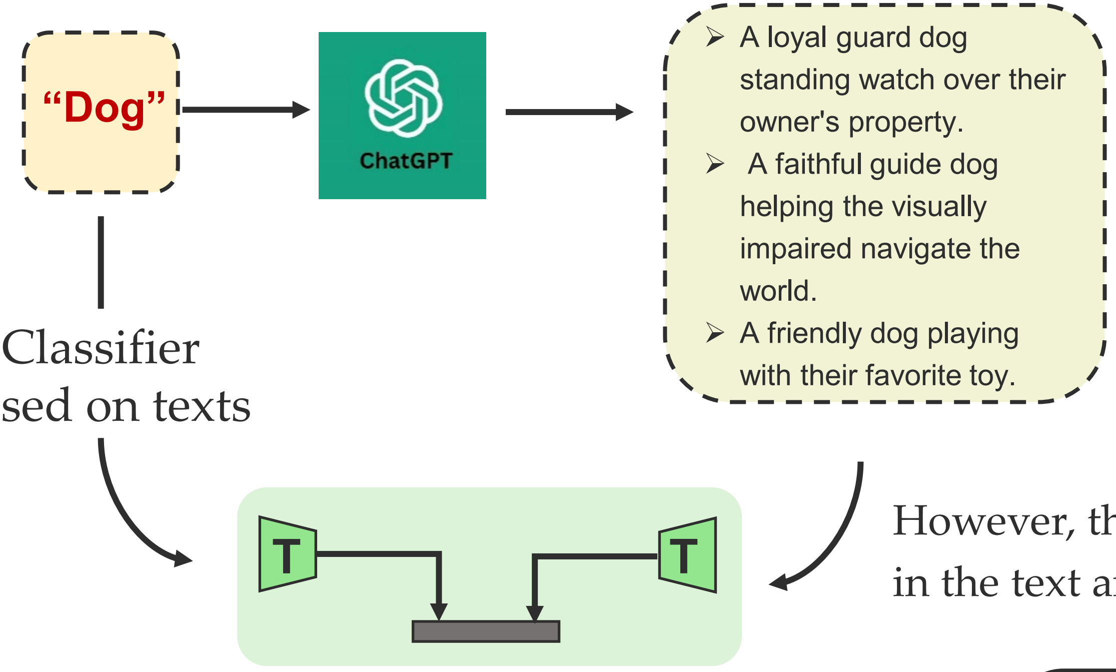
VLM models (e.g., CLIP) have strong zero-shot image classification capabilities



Select the best VLM from the model zoo ***solely based on texts***

Language-Only VLM Selection (LOVM)

Extract representations of target semantics based on a given VLM model

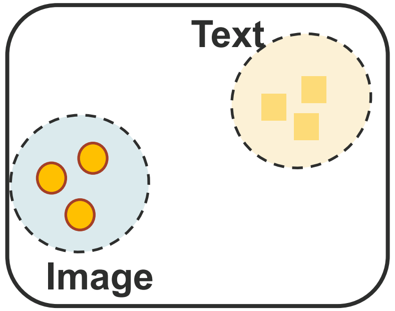


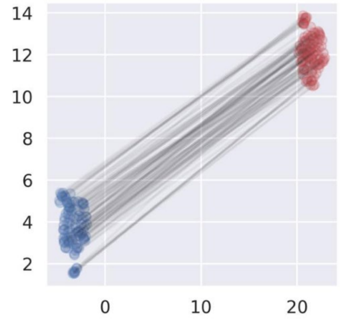
Classifier based on texts

- A loyal guard dog standing watch over their owner's property.
- A faithful guide dog helping the visually impaired navigate the world.
- A friendly dog playing with their favorite toy.

After describing the target category using language models like GPT, extract representations of target semantics based on a given VLM model

However, there is a *gap* between representations in the text and image spaces generated by VLMs.

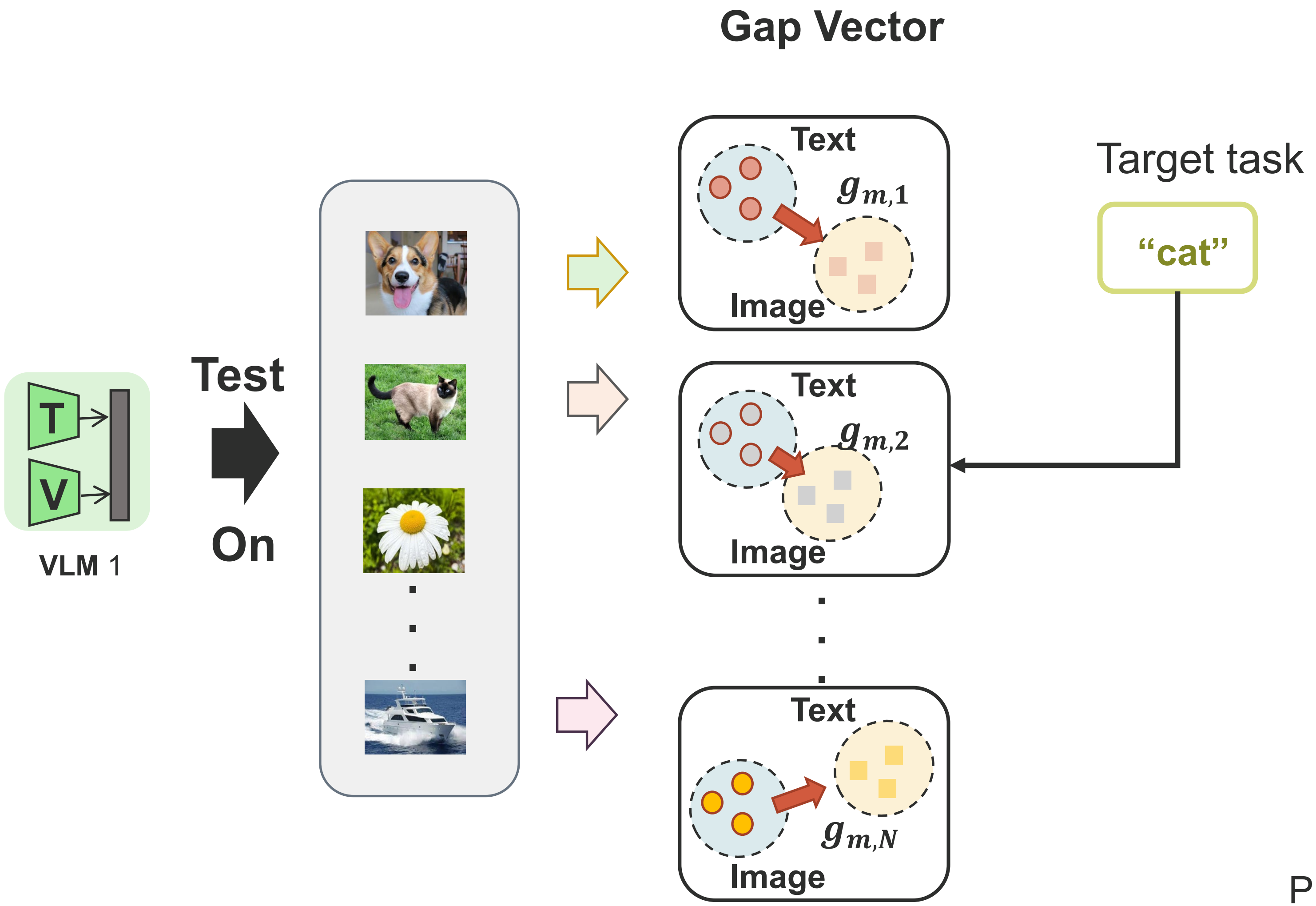




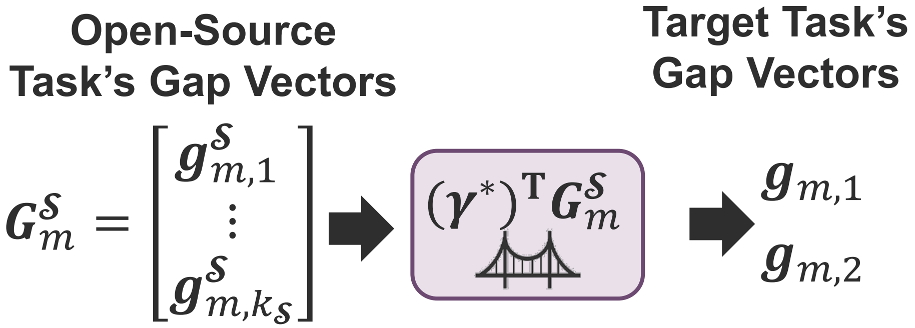
ModelGPT: convert the cross-modal matching problem between VLM and tasks into similarity calculation of in the text space.

Orr Zohar, Shih-Cheng Huang, Kuan-Chieh Wang, Serena Yeung. *LOVM: Language-only vision model selection*. NeurIPS 2023.

SWAB: Bridge the Modality Gap

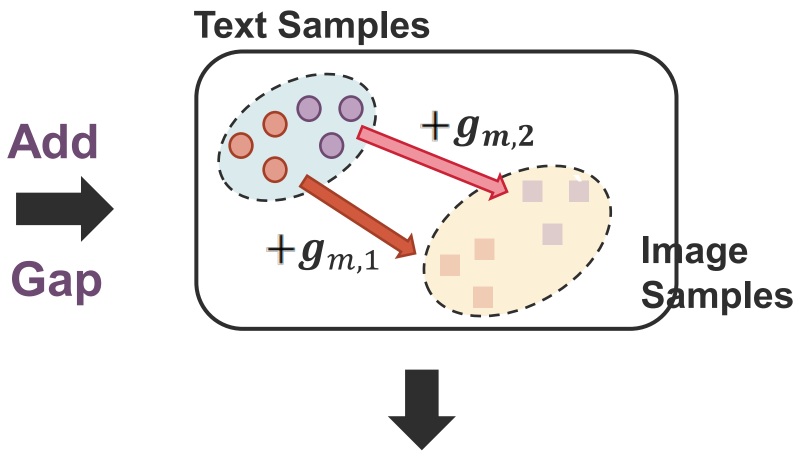


Estimate the gap vector for target class



$$G_m^{\mathcal{S}} = \begin{bmatrix} g_{m,1}^{\mathcal{S}} \\ \vdots \\ g_{m,k_{\mathcal{S}}}^{\mathcal{S}} \end{bmatrix} \Rightarrow \boxed{(\gamma^*)^{\mathrm{T}} G_m^{\mathcal{S}}} \Rightarrow \begin{matrix} g_{m,1} \\ g_{m,2} \end{matrix}$$

Rectify the embeddings of texts



Predict the ranking based on rectified embeddings

Chao Yi, Yu-Hang He, De-Chuan Zhan, Han-Jia Ye. *Bridge the Modality and Capacity Gaps in Vision-Language Model Selection*. NeurIPS 2024.

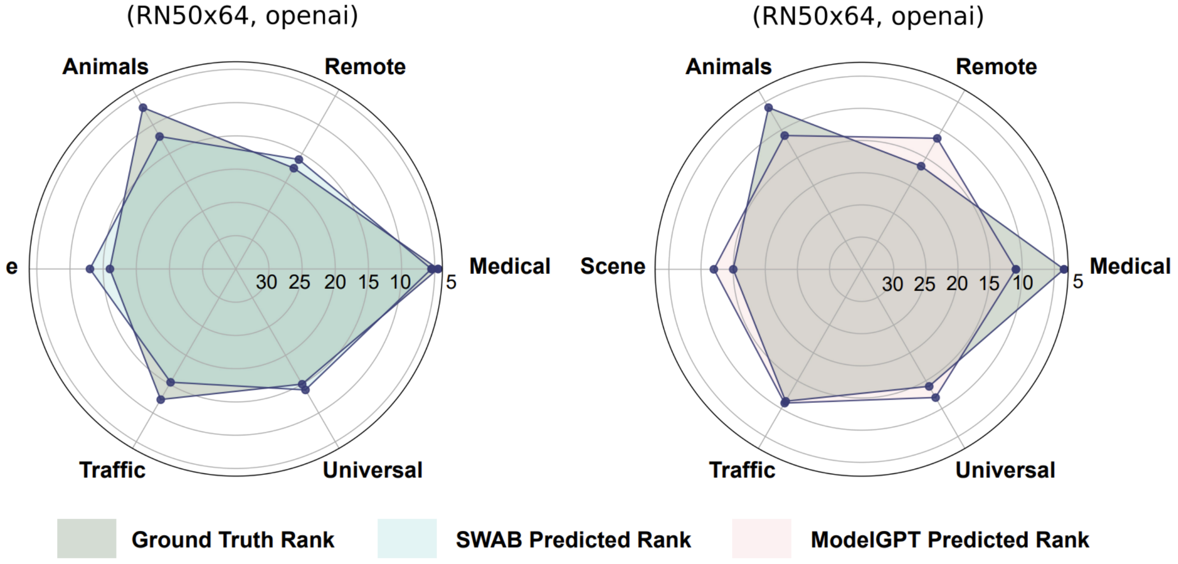
Result on VLMs Selection

text-based selection methods

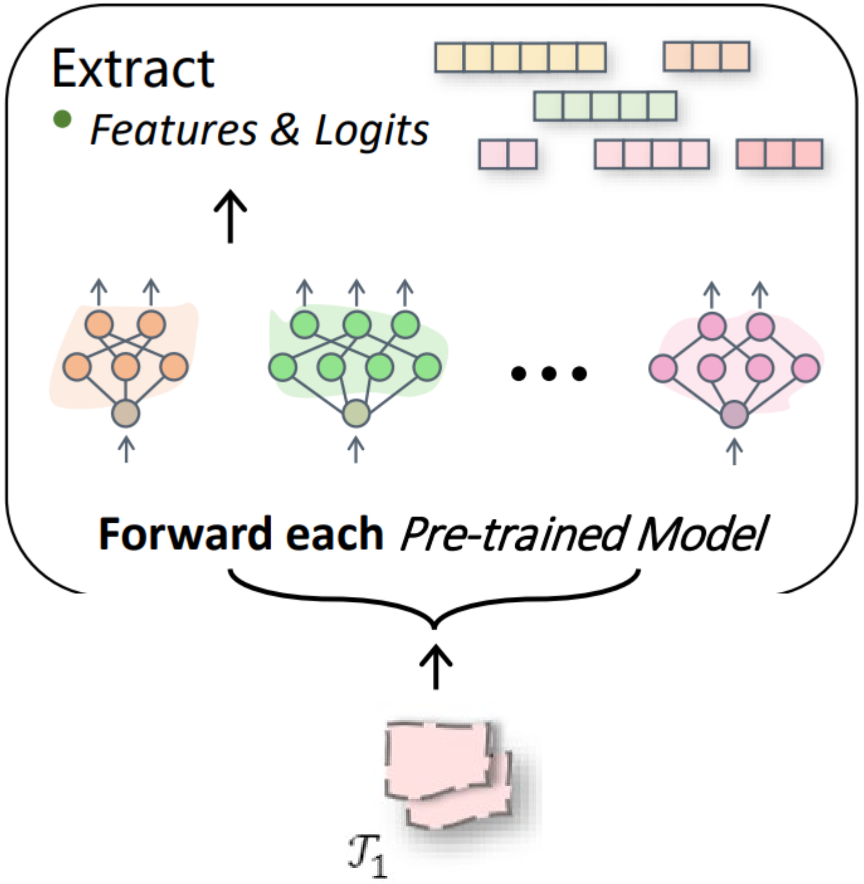
w/o considering the modality Gap

Methods	H-Score	NCE	LEEP	LogME	ModelGPT	SWAB
$R_5(\uparrow)$	0.165	0.270	0.330	0.200	0.457±0.006	**0.534**±0.023
$\tau(\uparrow)$	0.000	0.087	-0.014	-0.029	0.197±0.016	**0.260**±0.017
$R_5 + \tau(\uparrow)$	0.165	0.357	0.316	0.171	0.654±0.019	**0.794**±0.010

The model zoo contains 43 VLMs (including CLIP, BLIP, BEiT-3), and the downstream tasks include 23 zero-shot classification tasks.



Metric-Based Methods



- Provide a quantitative measure of suitability, making comparisons between models straightforward and reducing subjective bias.
- Adapt well to different types of data and tasks by adjusting the metrics to reflect the specific needs of the target task.

- The effectiveness heavily relies on the appropriateness and robustness of the chosen metric.
- Some metrics, especially those involving BP or complex statistical analyses, can be computationally intensive, limiting the use in resource constrained environments.

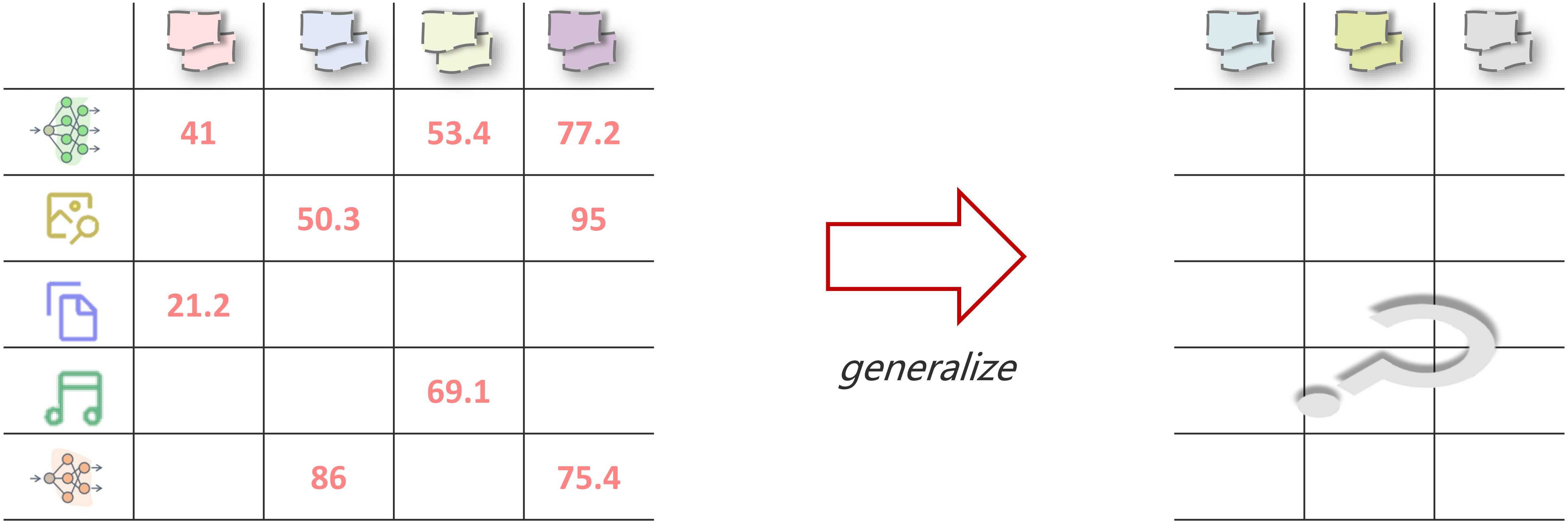
Forward-based methods require forwarding **for each pre-trained model**
which still incurs significant overhead! 😥

How to select PTM *efficiently* when the model zoo is large?

Learning-Based Methods

Main idea: automatically learn the specifications in a data-driven manner

Automatically learning the specifications. Project models, as well as datasets, into the same embedding space, where the retrieval process can be easily done by distance calculation in the unified space.



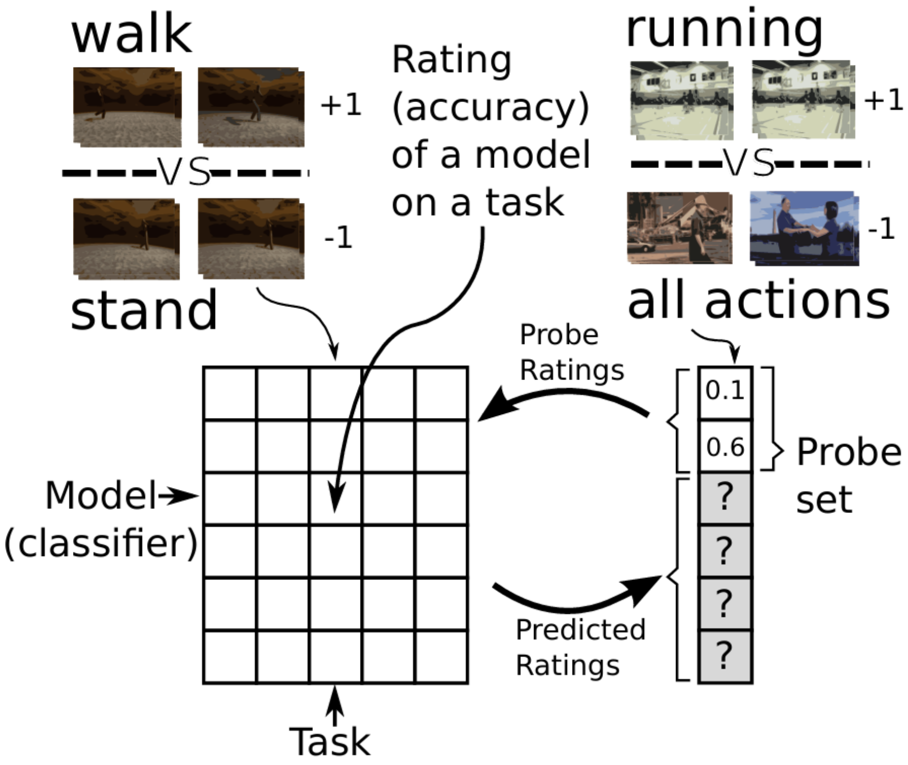
	41		53.4	77.2
		50.3		95
	21.2			
			69.1	
		86		75.4

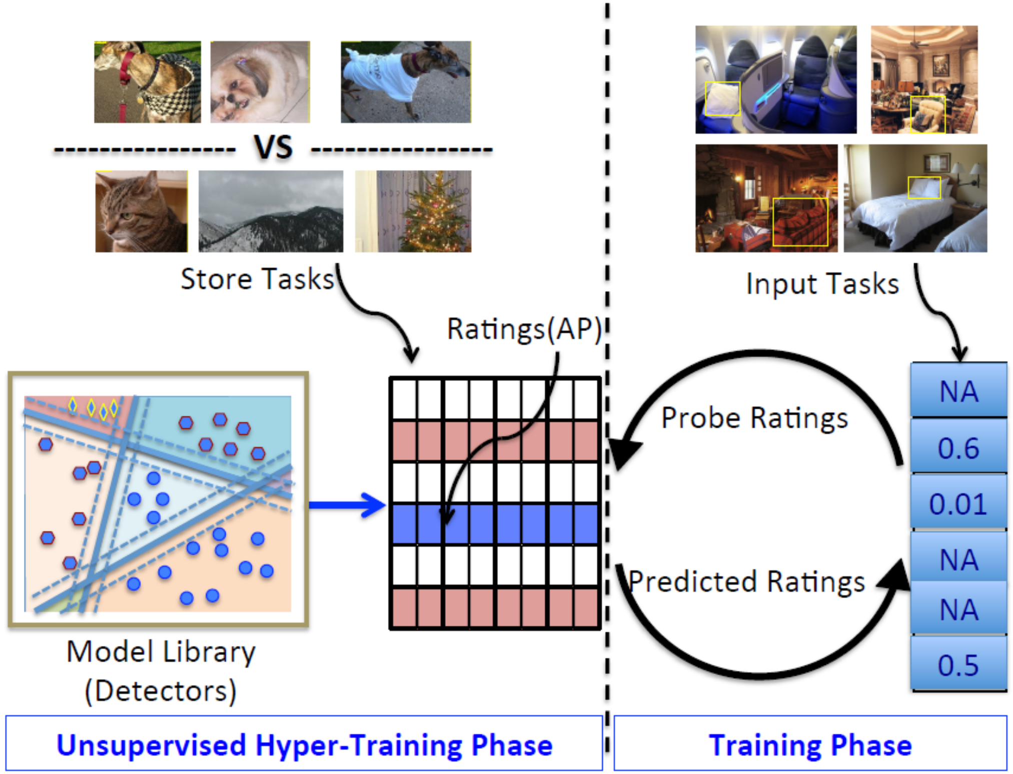
Sparse historical PTM-task interaction

Estimate the ability of the PTM on ***unseen*** *tasks*

Model Recommendation

Assume fixed set of models and tasks. Given known performance of task-model pairs, utilize collaborative filtering to predict the unknown ones.

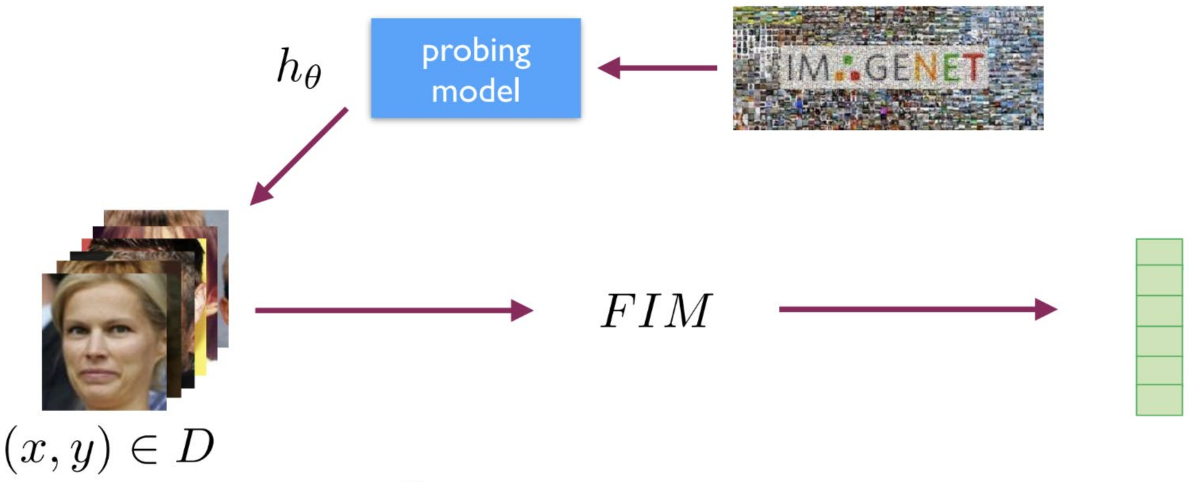




Pyry Matikainen, Rahul Sukthankar, Martial Hebert. *Model Recommendation for Action Recognition*. CVPR 2012.
Yu-Xiong Wang and Martial Hebert. *Model Recommendation: Generating Object Detectors from Few Samples*. CVPR 2015.

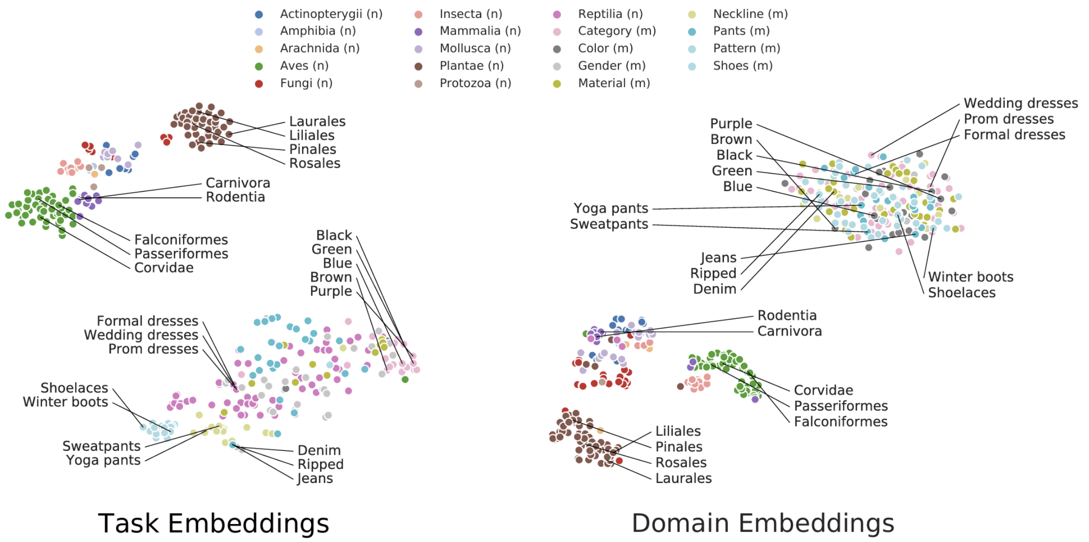
Task2Vec

Task2Vec involves passing data through a probe network to *generate a task-specific embedding that captures the complexity and characteristics of the task*.



$$FIM = \frac{1}{N}\sum_{i=1}^{N}[\nabla_\theta \log h_\theta(y_i|x_i)][\nabla_\theta \log h_\theta(y_i|x_i)]^T$$

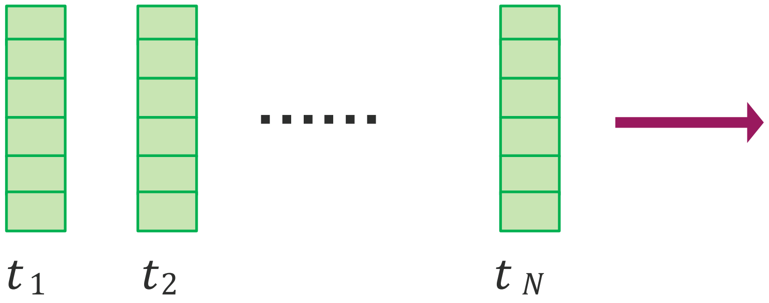
Fisher Information Matrix, measuring how sensitive/surprised when the model sees the task.



Alessandro Achille et al., *Task2vec: Task embedding for meta-learning*. ICCV 2019.

Model2Vec

Given a set of task embeddings, for PTMs $f_1, f_2, \ldots, f_K$, conduct metric learning to obtain their embeddings:



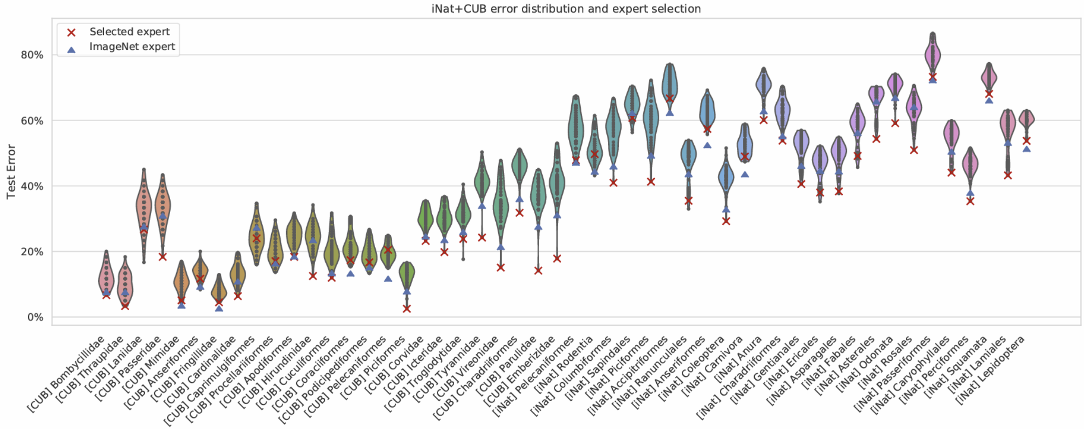
learnable parameters

$$m_k = (t_1 + b_1) \oplus (t_2 + b_2) \oplus \cdots \oplus (t_N + b_N)$$

Training objective

$$-\log p\left(m \mid d(t, m_1), d(t, m_2), \ldots, d(t, m_K)\right)$$

Metric learning: utilize the infoNCE loss to maximize the likelihood of the optimal task-model pairs



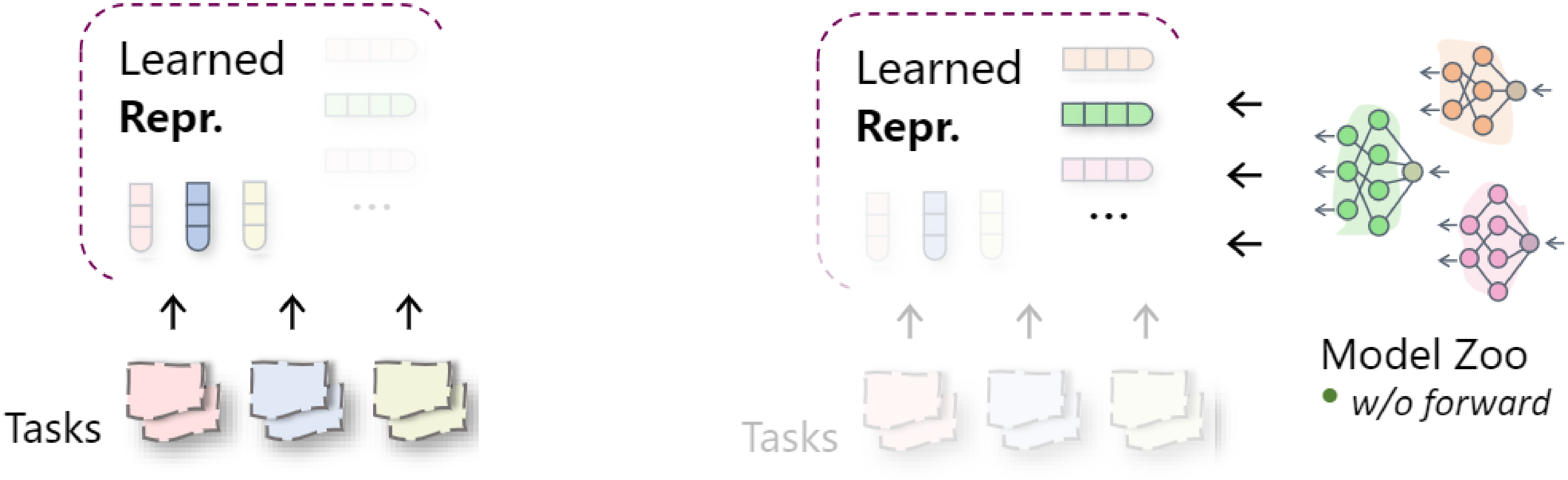
Restrictions:

- Need a strong probing model.
- The embedding space characterizes only task information.

Alessandro Achille et al., *Task2vec: Task embedding for meta-learning*. ICCV 2019.

Could We Reduce the Selection Burden?

Meta-Representation



Task Representation

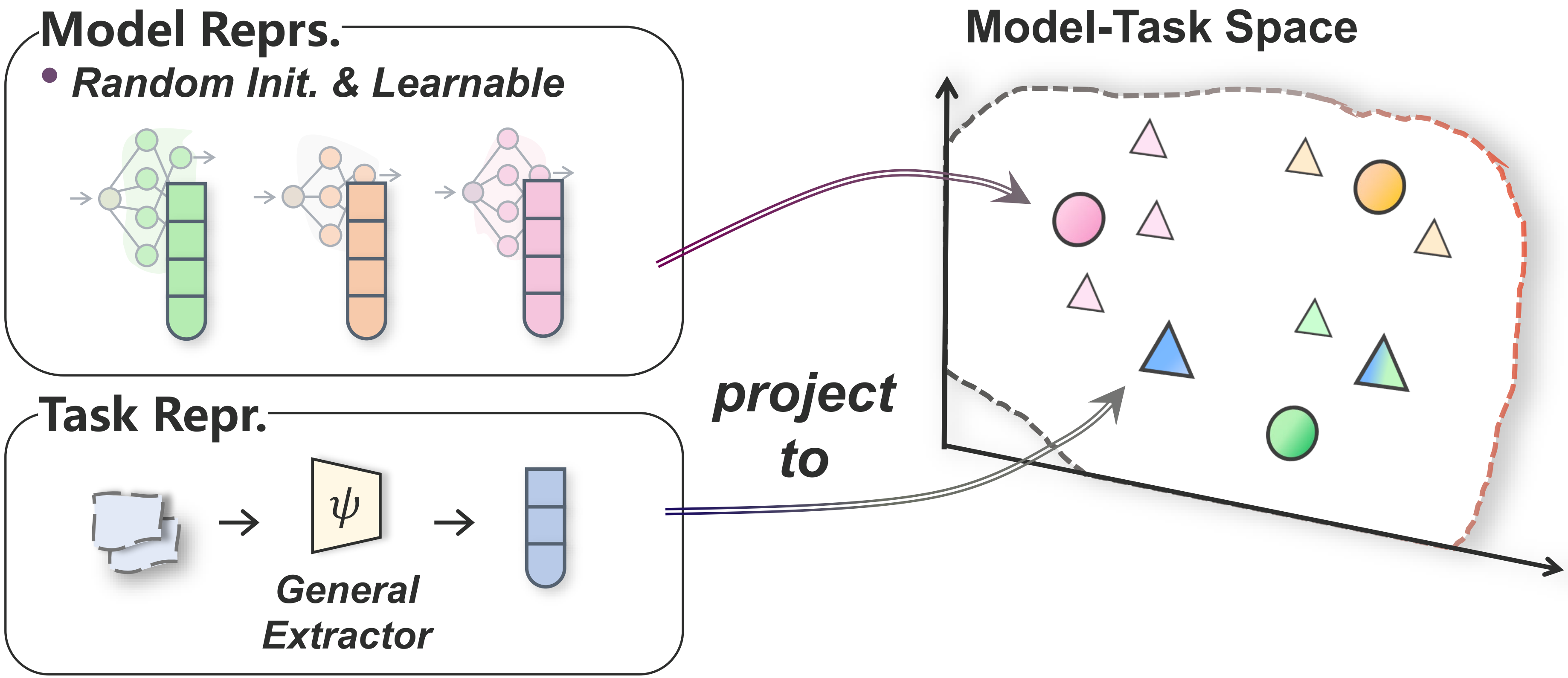
Just need one-time forward
of a ***general*** **encoder ψ**

Model Representation

Learn model representation,
one for each PTM

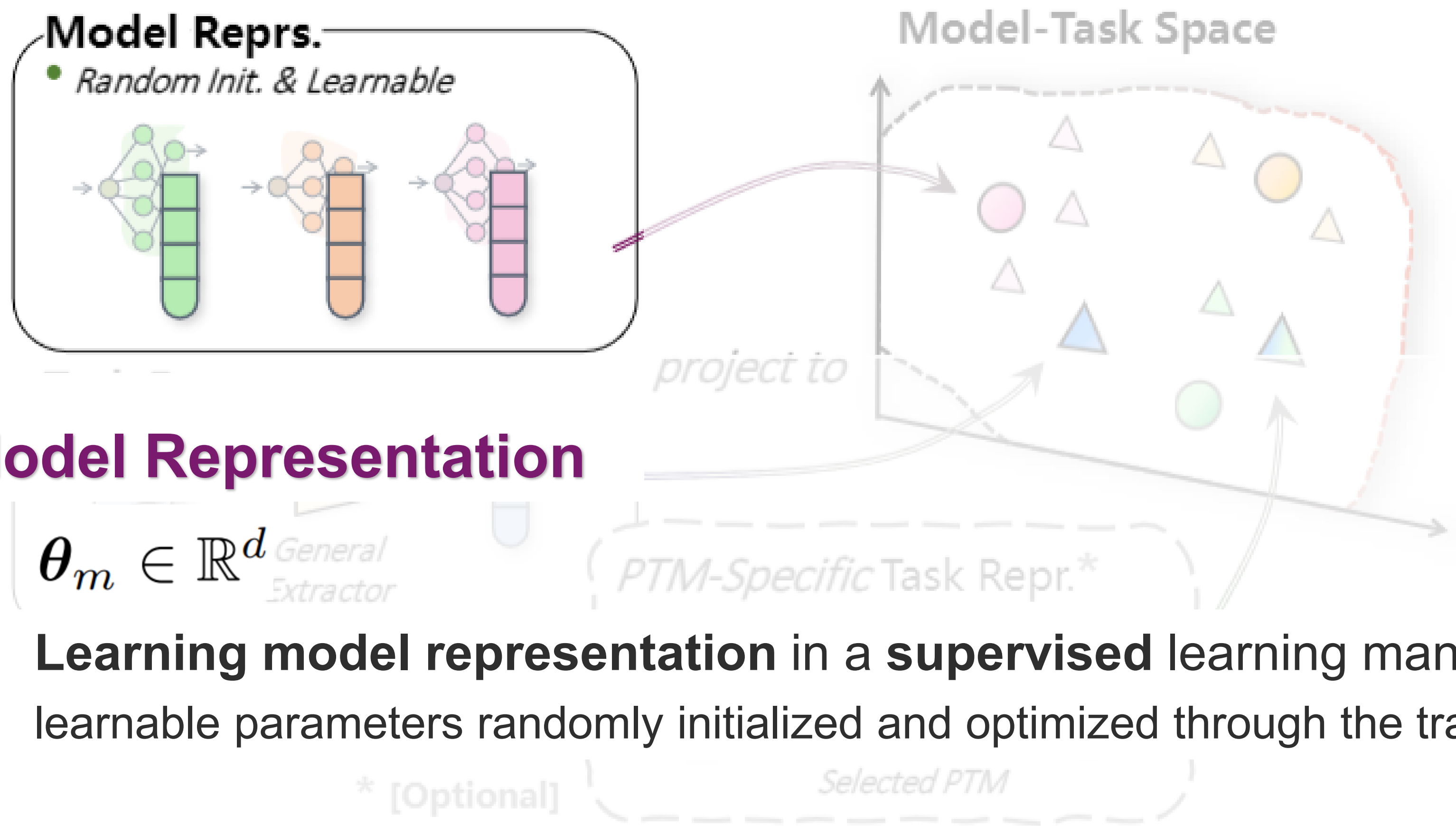
Yi-Kai Zhang, Ting-Ji Huang, Yao-Xiang Ding, De-Chuan Zhan, Han-Jia Ye. *Model Spider: Learning to Rank Pre-Trained Models Efficiently*. NeurIPS 2023.

Mapping PTMs and Tasks into a Space



Using **historical** interactions between PTMs and seen tasks to learn a selector, and apply it to **unseen tasks and new models**.

Model Representation



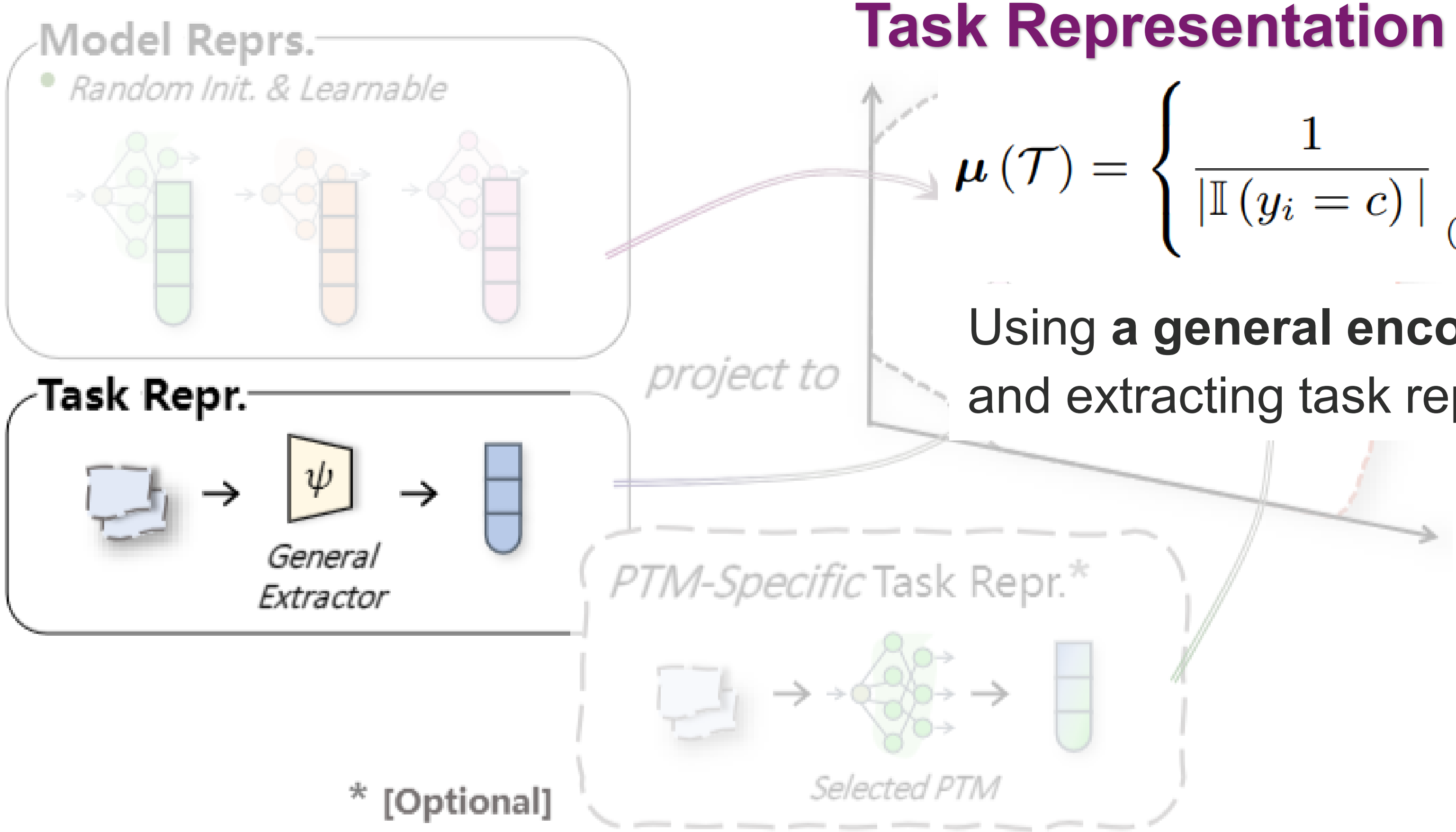
Model Representation

$$\boldsymbol{\theta}_m \in \mathbb{R}^d$$

Learning model representation in a **supervised** learning manner

learnable parameters randomly initialized and optimized through the training process

Task Representation

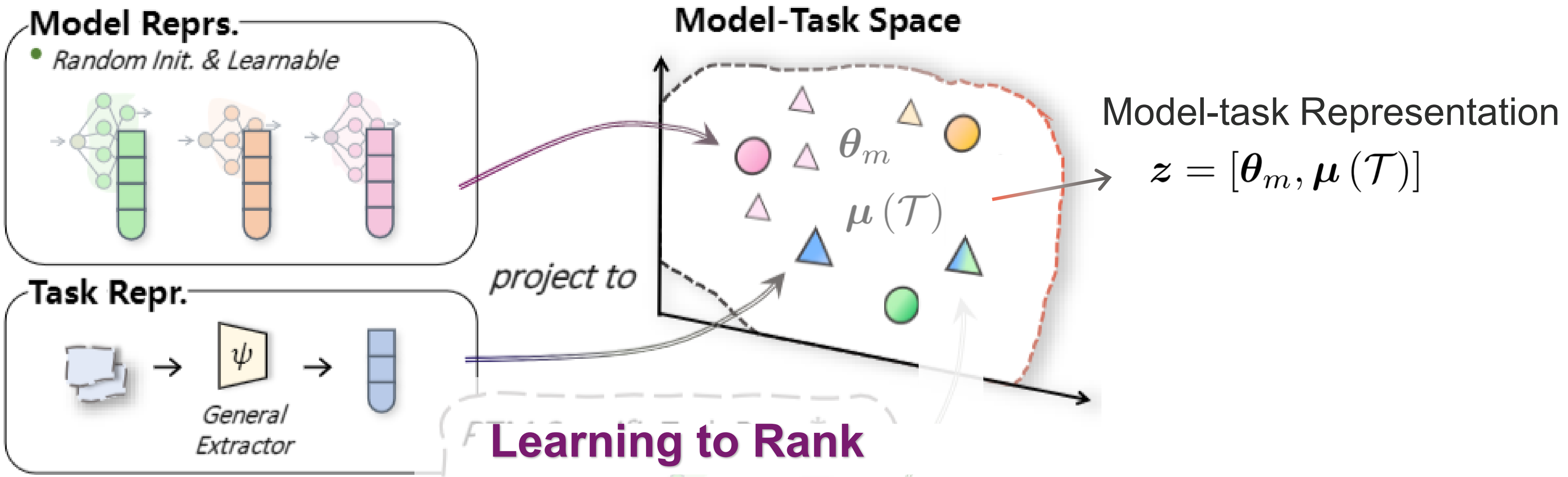


Task Representation

$$\boldsymbol{\mu}\left(\mathcal{T}\right) = \left\{ \frac{1}{\left|\mathbb{I}\left(y_i = c\right)\right|} \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{T}} \left[\psi\left(\boldsymbol{x}_i\right) \cdot \mathbb{I}\left(y_i = c\right)\right] \right\}_{c \in [C]}$$

Using **a general encoder** ψ
and extracting task repr. with **just one** forward pass.

Learning to Rank



Learning to Rank

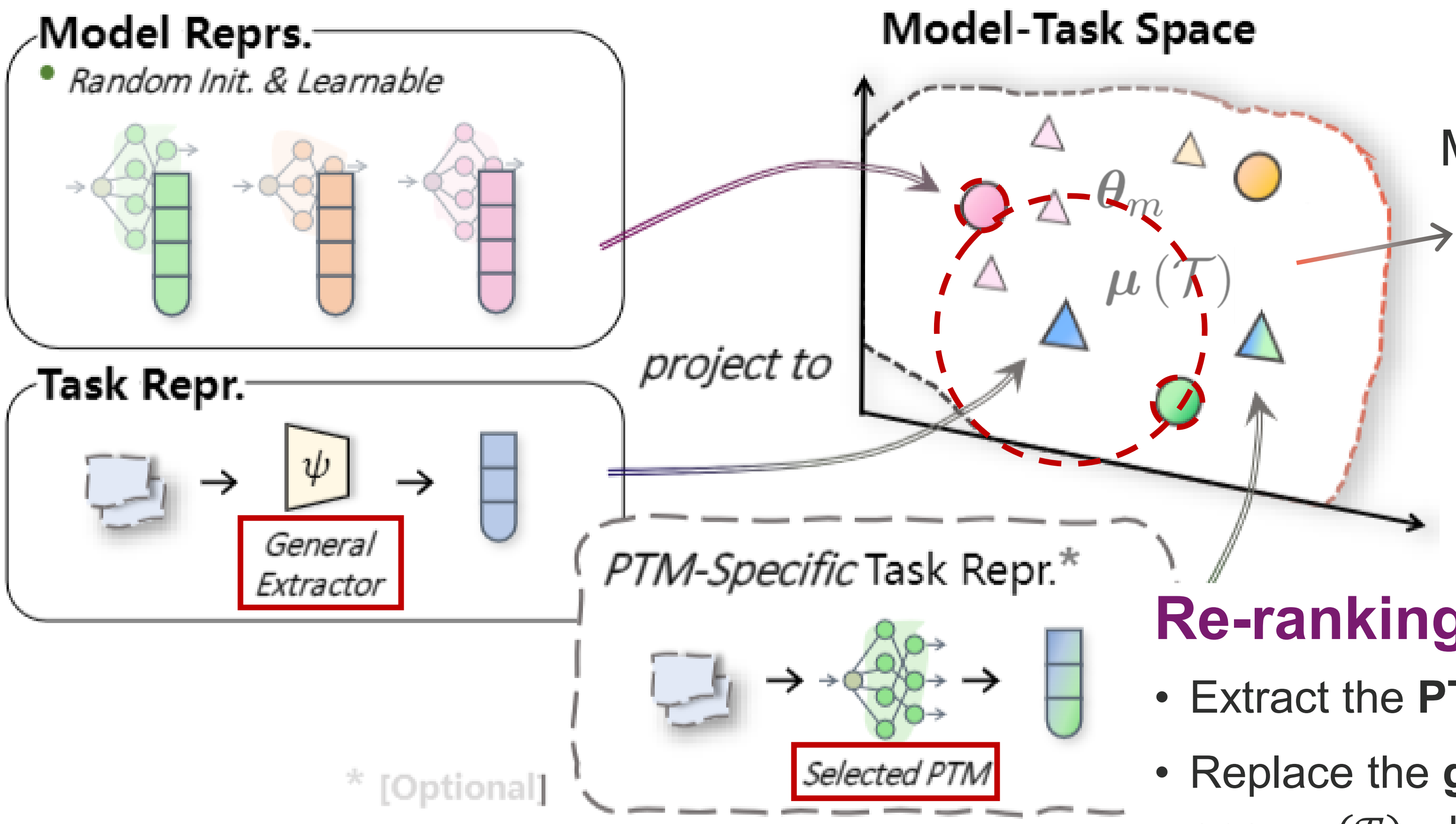
$$\mathrm{sim}(\boldsymbol{\theta}_m, \boldsymbol{\mu}(\mathcal{T})) = \mathrm{FC}\left(\mathrm{transformer}(\boldsymbol{z})[0]\right)$$

$$\min_{\Theta} \sum_{\mathcal{T}_i\sim\mathcal{D}} \ell_{\mathrm{rank}}\left(\hat{\boldsymbol{t}}_{\mathcal{T}_i} = \{\mathrm{sim}(\boldsymbol{\theta}_m, \boldsymbol{\mu}(\mathcal{T}_i))\}_{m=1}^{M}, \boldsymbol{t}_{\mathcal{T}_i}\right)$$

The ranking ground-truth is **approximated via the rank-aggregation** of various forward-based methods.

Model Spider

Model Spider: ***Learning to rank*** pre-trained models with **model** and **task representation**



Model-task Representation

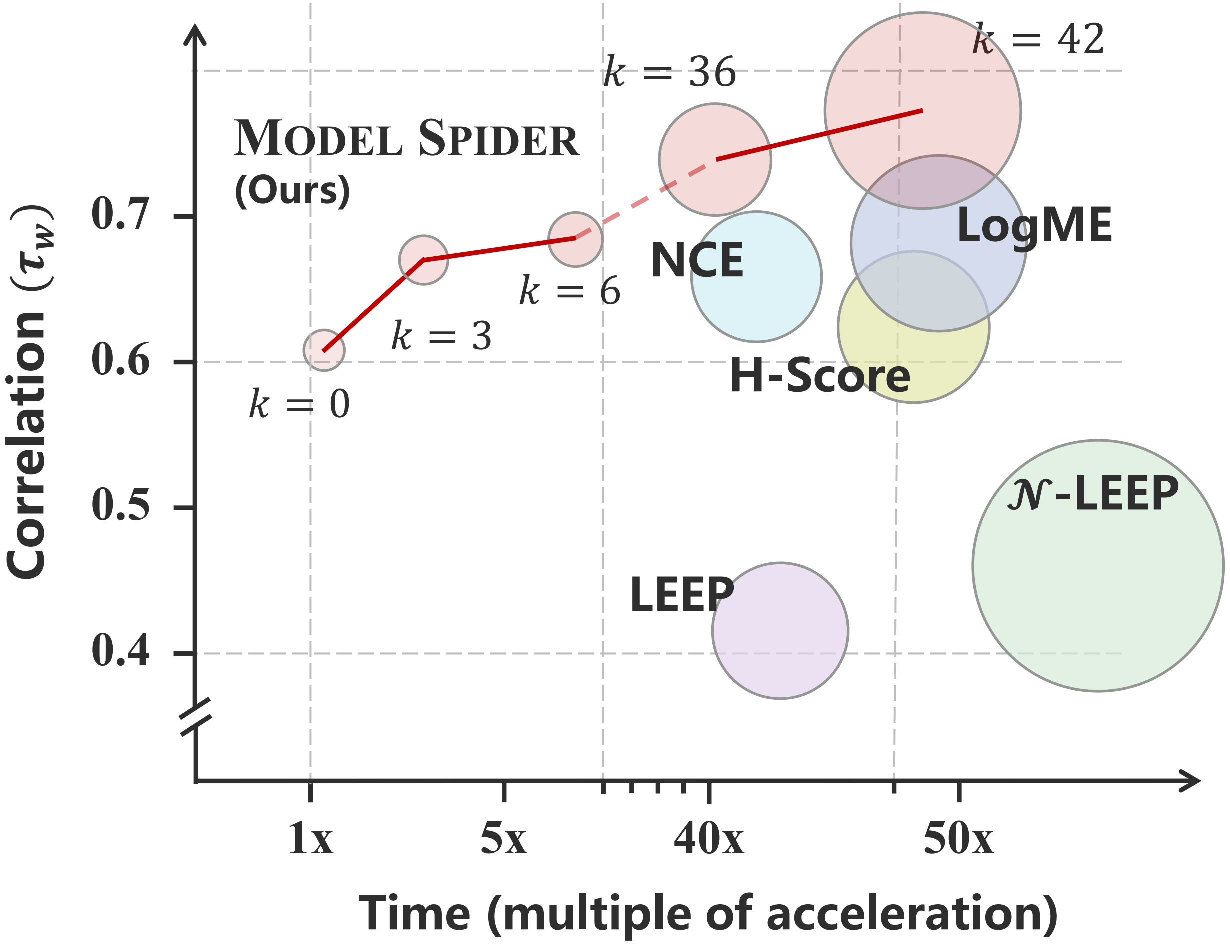
$$\boldsymbol{z} = [\boldsymbol{\theta}_m, \boldsymbol{\mu}(\mathcal{T})]$$

Re-ranking w/ Efficiency-Accuracy Trade-off

- Extract the **PTM-specific** task representation
- Replace the **general** task repr. $\mu(\mathcal{T})$ via the **specific** one $\mu_m(\mathcal{T})$ when calculating the **similarity** with the $\boldsymbol{\theta}_m$ **of the mth PTM**

Yi-Kai Zhang, Ting-Ji Huang, Yao-Xiang Ding, De-Chuan Zhan, Han-Jia Ye. *Model Spider: Learning to Rank Pre-Trained Models Efficiently*. NeurIPS 2023.

Results on Image Classification Tasks



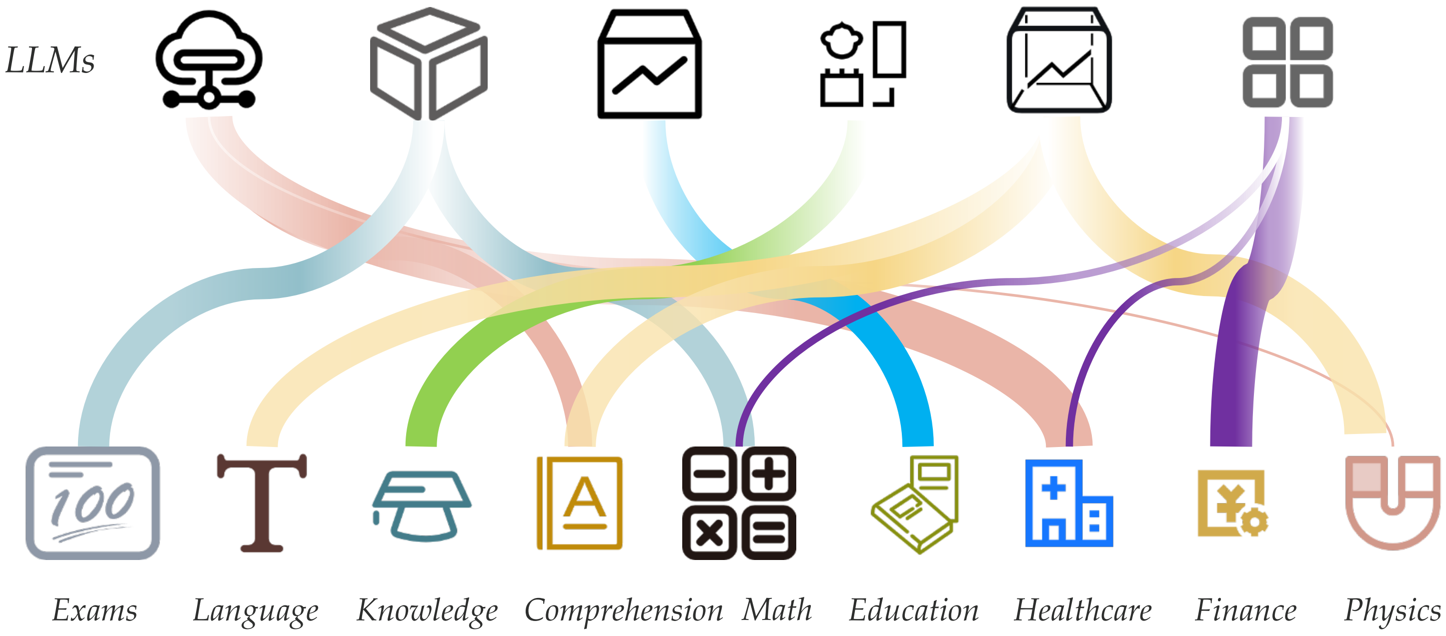
Model Spider:

***Efficiency-Accuracy* Trade-off**

The circle sizes indicate the memory footprint. **Red circles** are our method with different values of the number of **PTM-specific features** k

Large Language Model Routing

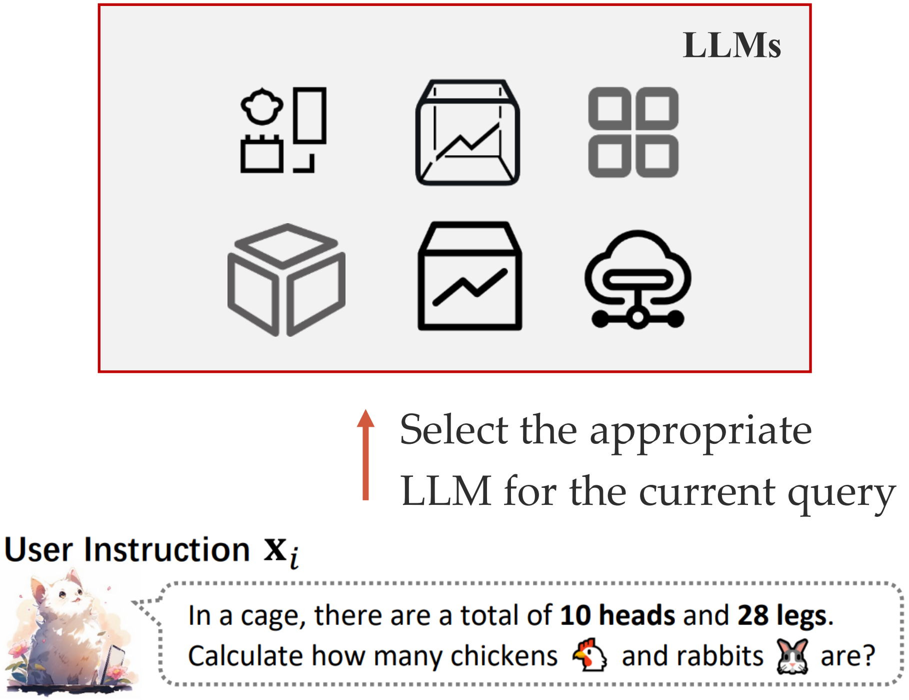
LLM routing is a learning-based decision mechanism that dynamically selects or combines models from a candidate pool $\mathcal{M} = \{M_1, \dots, M_k\}$ for each input x, in order to achieve a controllable trade-off among **output quality, latency, and computational or monetary cost**.



Large Language Model Routing

LLM routing is a learning-based decision mechanism that dynamically selects or combines models from a candidate pool $\mathcal{M} = \{M_1, \dots, M_k\}$ for each input x, in order to achieve a controllable trade-off among **output quality, latency, and computational or monetary cost**.

Formally, **a router learns a policy $p(M_k \mid x)$**, which can be instantiated as a ***hard decision*** (e.g., Top-1 or Top-K selection) or a ***soft allocation*** (a probability or weight distribution).
At inference time, the system invokes the selected model(s) and ***returns either a single output or an aggregation*** (e.g., weighted fusion or re-ranking).

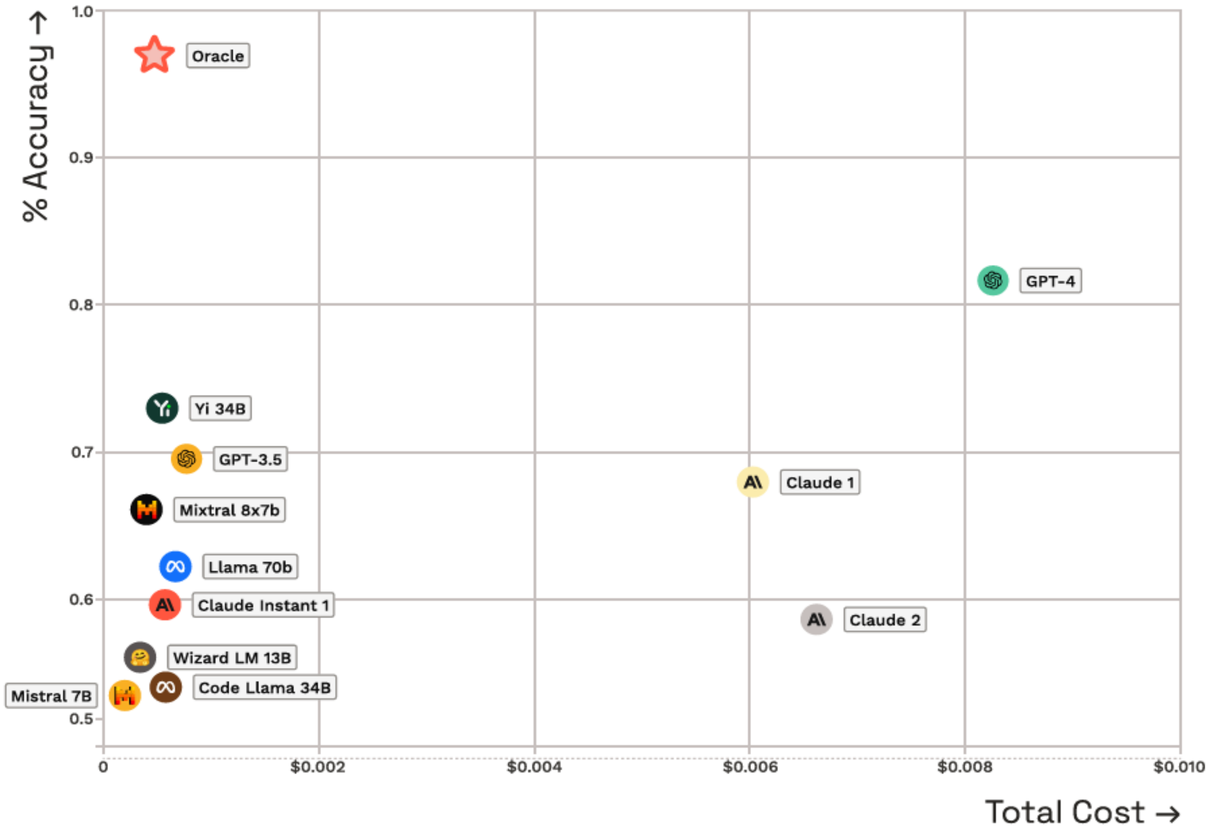


Select the appropriate LLM for the current query

User Instruction $\mathbf{x}_i$

In a cage, there are a total of **10 heads** and **28 legs**. Calculate how many chickens 🐔 and rabbits 🐰 are?

LLM Routing Benchmarks



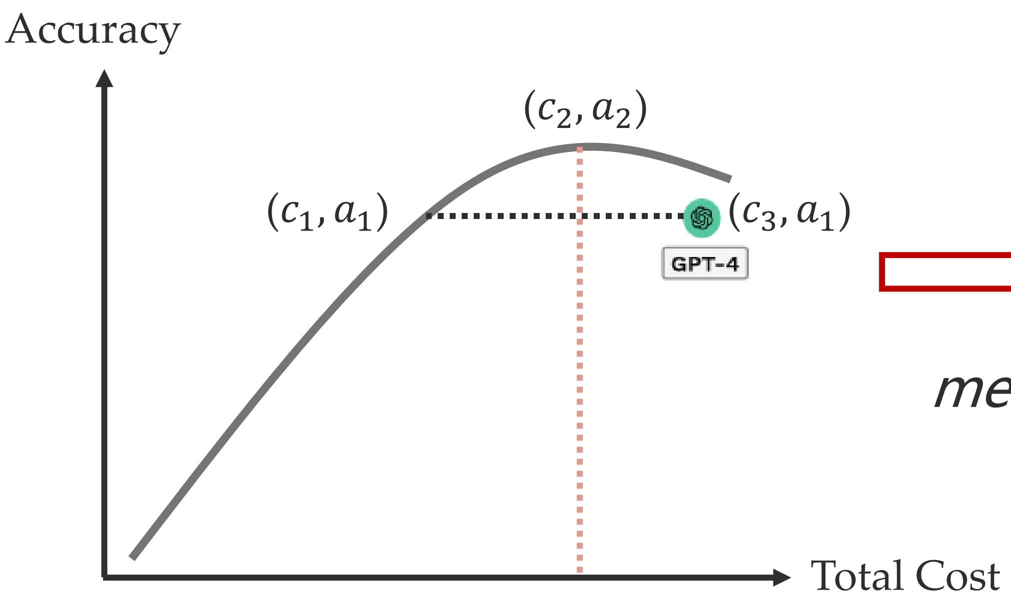
RouterBench is a standardized evaluation benchmark designed specifically for multi-LLM routing systems.

- It contains over 400K precomputed model inference records, covering the outputs of multiple LLMs on a wide range of tasks.
- The data is collected from **dozens of tasks across multiple domains** (e.g., reasoning, QA, math, coding, RAG, etc.) and from **11 representative LLMs**, including both open-source and API-based models.

Qitian Jason Hu, et al. *RouterBench: A Benchmark for Multi-LLM Routing System.* ICMLW 2024.

Metrics

By evaluating routing algorithms on a benchmark, we can obtain a deferral curve that characterizes the ***trade-off between cost and accuracy***. Based on this curve, three evaluation metrics are typically derived.



metrics

Peak Accuracy

a_2, the best performance that router can achieve

Query-Normalized Cost

QNC=$\frac{c_1}{c_3}$, the minimum relative cost required to match the performance of the most accurate tested LLM

Area

The area of the curve

Isaac Ong, et al. *RouteLLM: Learning to route LLMs from preference data*, ICLR 2025.

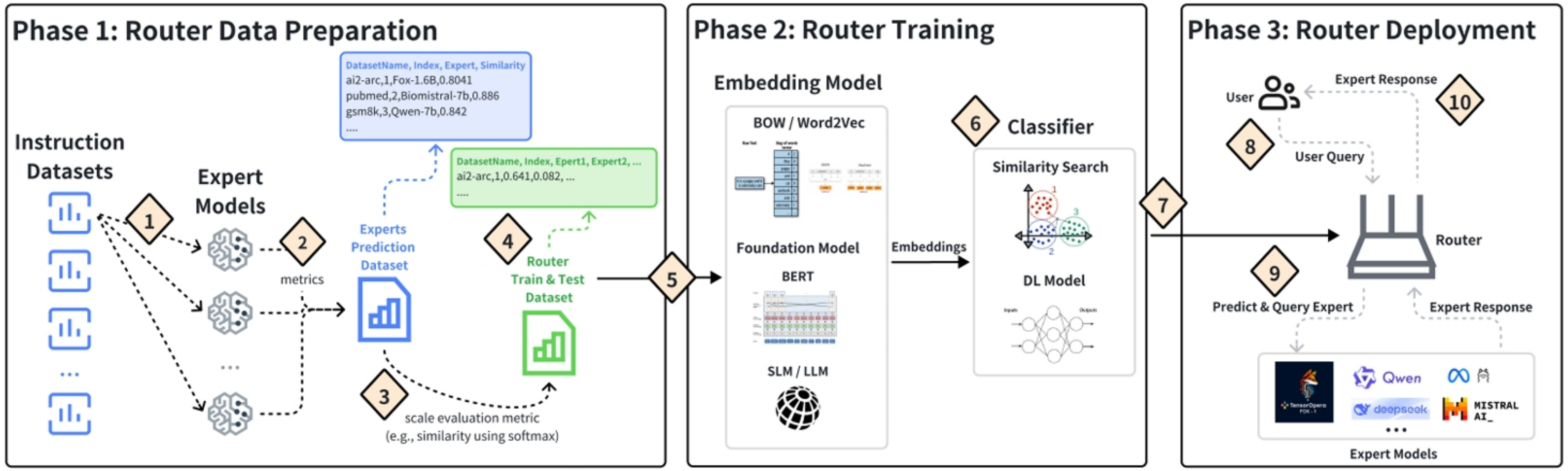
Relationship to Mixture-of-Experts

Both Routing and MoE share the idea of allocating inputs to specialized units.

	Routing	**MoE**
Granularity	The "experts" are **complete, standalone pretrained models** that may differ in architecture, scale, and training data.	Experts are usually **parameterized sub-networks** within a single model
Training paradigm	LLM routers are often **trained with indirect signals** such as offline evaluations, reward models, or user preference data	MoE gating is typically **trained end-to-end** together with the backbone model using the downstream loss

LLM Routing

Learn a mapping from query features (extracted via Word2Vec, BERT, LLM, etc.) to the target PTM.



1. Prompt expert models using instruction datasets.
2. Collect performance metrics and create expert dataset with all metrics.
3. Select metric for router training and testing and scale it to probabilities.
4. Prepare router's final training and testing datasets.
5. Pass instruction data through embedding model.
6. Fine-Tune / Train router's classifier on the embeddings.
7. Deploy trained router model.
8. Submit user query to router.
9. Router predicts expert to execute query.
10. Router replies expert's response to user.

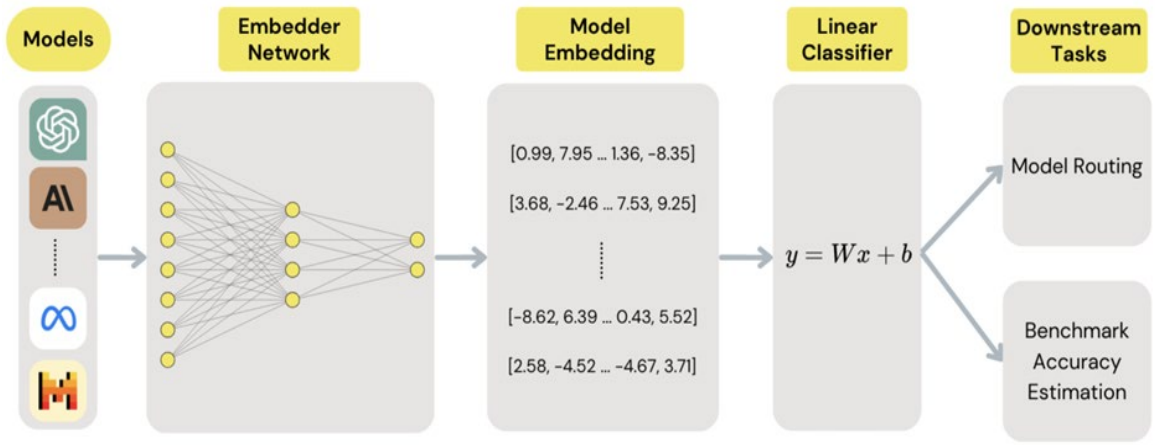
Implement the "router" with common "classifier", such as

- **MLP**: learn a parametric mapping [Stripelis et al., EMNLP'24]
- **KNN**: retrieve LLM based on the historical query-LLM results [Stripelis et al., EMNLP'24] [Li, CoRR'25]
- **Kmeans/NCM**: learn multiple centroids for a cluster [Srivatsa, Maurya, Kochmar, ACLW'24] [Jitkrittum et al., ICLRW'25] [Zhang et al., AAAI'26] [Zhang et al., DAI'25]

Dimitris Stripelis et al. *TensorOpera Router: A Multi-Model Router for Efficient LLM Inference.* EMNLP 2024.

EmbedLLM

EmbedLLM trains an encoder–decoder model to learn a low-dimensional embedding for each LLM that reconstructs or predicts its task performance.


Models

Embedder Network

Model Embedding

[0.99, 7.95 ... 1.36, -8.35]

[3.68, -2.46 ... 7.53, 9.25]

[-8.62, 6.39 ... 0.43, 5.52]

[2.58, -4.52 ... -4.67, 3.71]

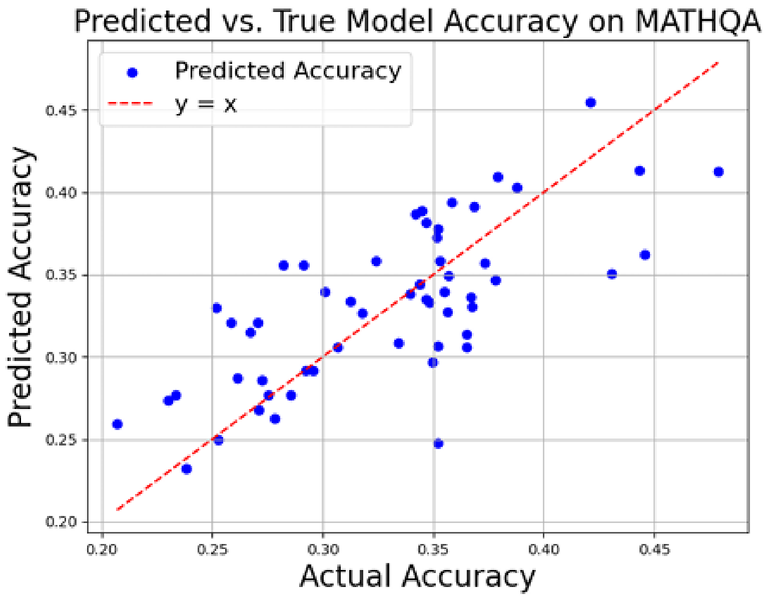
Linear Classifier

$y = Wx + b$

Downstream Tasks

Model Routing

Benchmark Accuracy Estimation



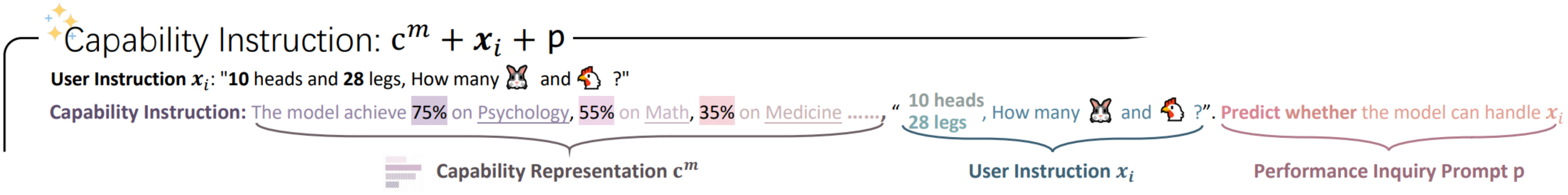
Train a lightweight router that takes the problem features and the model embeddings as input and selects the most suitable model.

Algorithm	Dataset Size						
	1K	5K	10K	15K	20K	25K	Full (29K)
KNN	0.6372	0.7078	0.7107	0.7128	0.7143	0.7146	0.7152
EmbedLLM	0.6443	0.7331	0.7362	0.7378	0.7390	0.7394	**0.7409**

- The learned embeddings are reusable across tasks, and even a simple linear classifier on top of them outperforms KNN baselines.
- A linear classifier has limited expressive power and can only represent linearly separable decision boundaries.
- Hard to generalize to new PTMs.

Richard Zhuang, et al, *EmbedLLM: Learning Compact Representations of Large Language Models*, ICLR 2025.

Capability Instruction Tuning

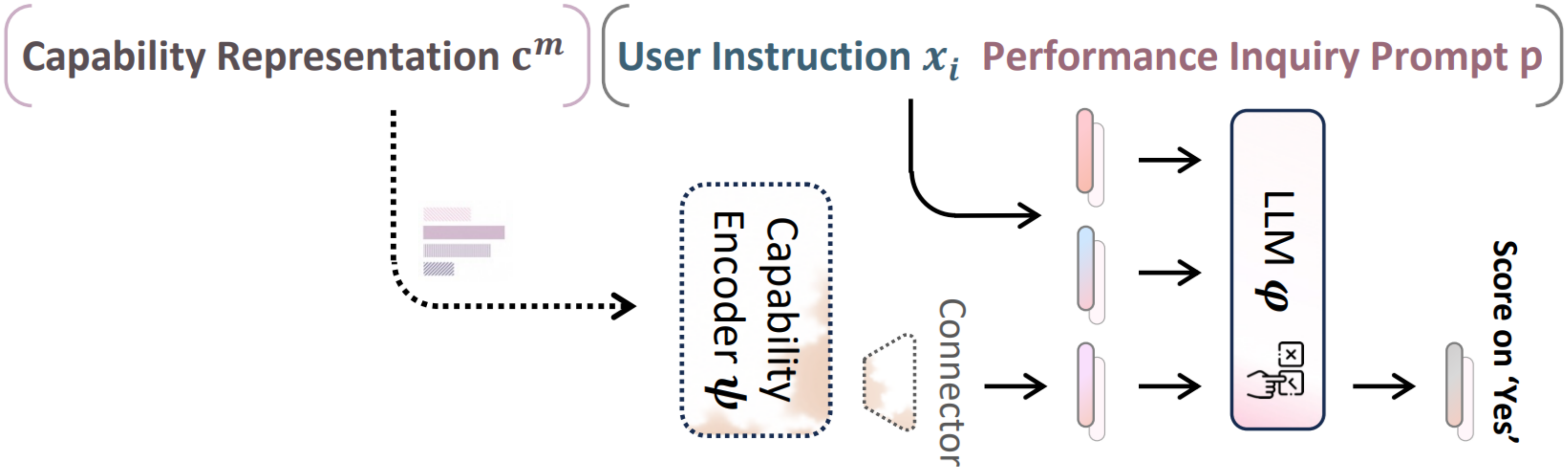


Capability Instruction: $c^m + x_i + p$

User Instruction x_i: "**10** heads and **28** legs, How many 🐰 and 🐔 ?"

Capability Instruction: The model achieve 75% on Psychology, 55% on Math, 35% on Medicine ……, " 10 heads 28 legs , How many 🐰 and 🐔 ?". **Predict whether** the model can handle x_i

Based on model descriptions and current task descriptions, estimate the model's capability on the current task



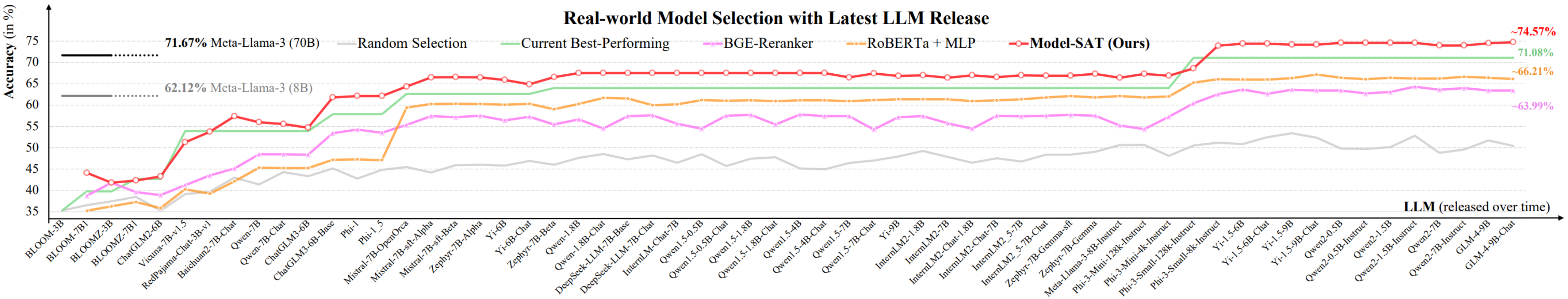
Capability Representation c^m
User Instruction x_i
Performance Inquiry Prompt p
Capability Encoder ψ
Connector
LLM φ
Score on 'Yes'

Implement the selector **in the form of an LLM**. Using instruction tuning to enable a general-purpose large model (Phi-1) to learn how to perform model-task matching.

Yi-Kai Zhang, De-Chuan Zhan, Han-Jia Ye. *Capability Instruction Tuning: A New Paradigm for Dynamic LLM Routing*. AAAI 2025.

Dynamic Selection of LLMs

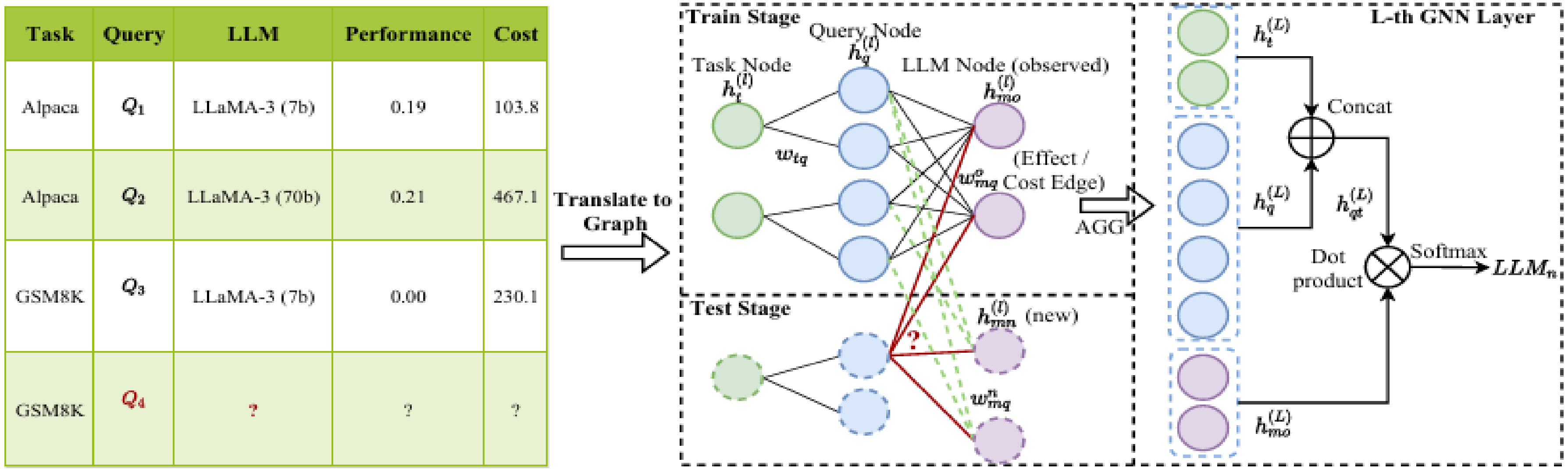
As time progresses, new LLMs are continuously released, and the model zoo is updated and expanded



The selection method can integrate the updated model zoo, select a better large model for the current user input, and achieve or ***even exceed the capabilities of the current optimal*** model across various tasks with fewer resources

GraphRouter

GraphRouter constructs a heterogeneous graph and predicts edges to select LLMs:



Task	Query	LLM	Performance	Cost
Alpaca	Q_1	LLaMA-3 (7b)	0.19	103.8
Alpaca	Q_2	LLaMA-3 (70b)	0.21	467.1
GSM8K	Q_3	LLaMA-3 (7b)	0.00	230.1
GSM8K	Q_4	?	?	?

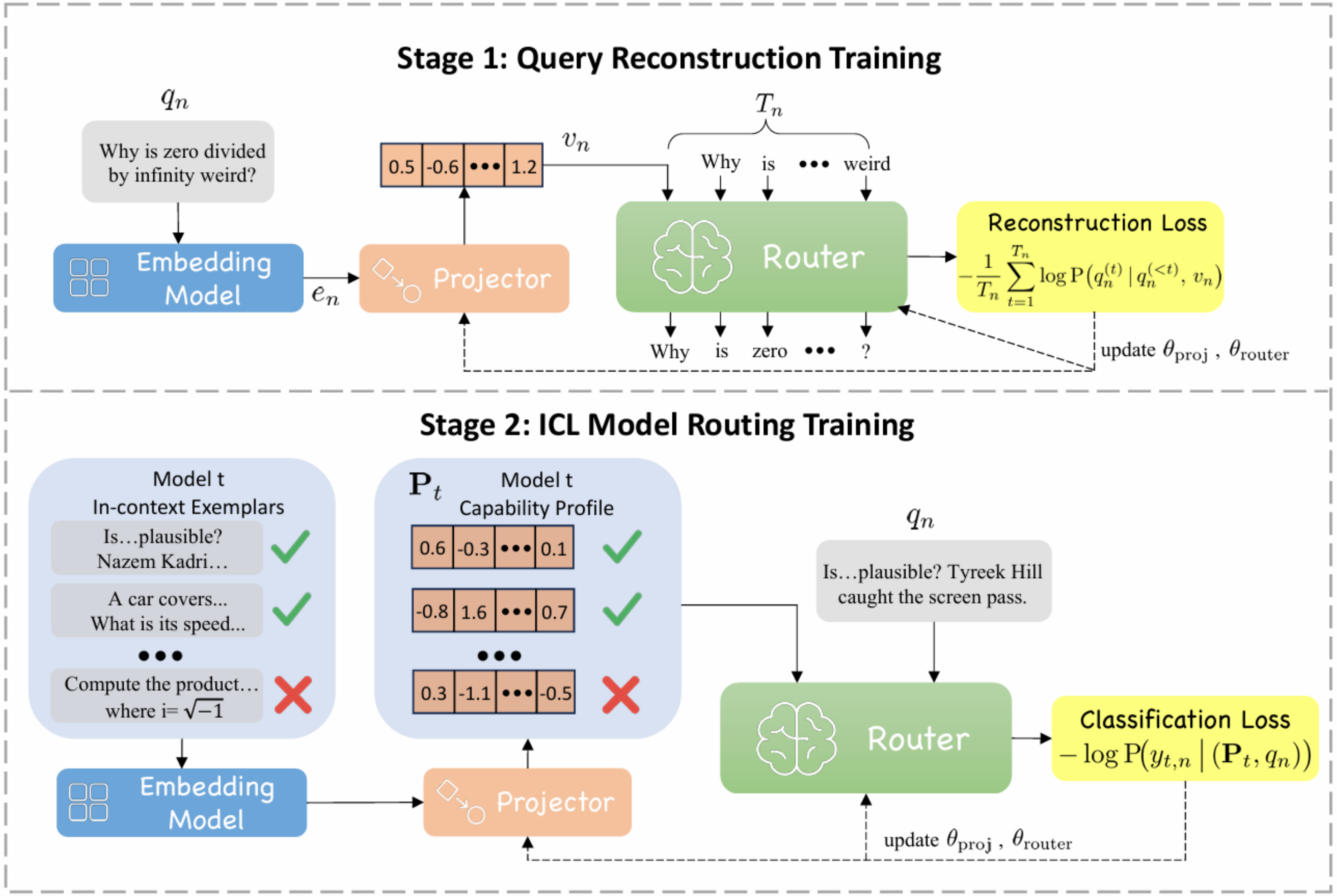
- Represent **tasks, queries, and LLMs as nodes in a heterogeneous graph**; edges encode historical interactions and performance/cost relationships.
- Apply a ***GNN to learn embeddings*** for nodes and edge features that capture contextual relationships among tasks, queries, and model capabilities.
- Perform ***edge prediction*** to estimate the expected performance and computational cost for a given query–LLM pair, and use these predicted attributes to select the LLM that best balances performance and cost.

- Model complex relational structures among queries, tasks, and models.
- Require constructing and maintaining a global graph over the entire training dataset, whose size and computational cost grow rapidly as the dataset scales.

Tao Feng, Yanzhen Shen, Jiaxuan You. *GraphRouter: A Graph-based Router for LLM Selections*. ICLR 2025.

ICL-Router

ICL-Router is a two-stage in-context routing method.


Stage 1: Query Reconstruction Training
q_n
Why is zero divided by infinity weird?
Embedding Model
e_n
Projector
0.5 | -0.6 | ••• | 1.2
v_n
T_n
Why is ••• weird
Router
Why is zero ••• ?
Reconstruction Loss
$-\frac{1}{T_n}\sum_{t=1}^{T_n}\log \mathrm{P}(q_n^{(t)} \mid q_n^{(<t)}, v_n)$
update θ_{proj} , θ_{router}
Stage 2: ICL Model Routing Training
Model t
In-context Exemplars
Is…plausible? Nazem Kadri…
A car covers... What is its speed...
•••
Compute the product… where i= $\sqrt{-1}$
$\mathbf{P}_t$
Model t
Capability Profile
0.6 | -0.3 | ••• | 0.1
-0.8 | 1.6 | ••• | 0.7
•••
0.3 | -1.1 | ••• | -0.5
Embedding Model
Projector
q_n
Is…plausible? Tyreek Hill caught the screen pass.
Router
Classification Loss
$-\log \mathrm{P}(y_{t,n} \mid (\mathbf{P}_t, q_n))$
update θ_{proj} , θ_{router}

- Encode each query into a vector representation and ***jointly train a projector and LLM-based router*** to reconstruct the original query from its vector.
- For each candidate model, profile its performance on a fixed set of queries and convert these into in-context vectors that represent model capability profiles.
- Train the router to take the query vector and a model's in-context vectors as input and predict whether the model can correctly answer the query, then select the model with highest predicted success probability.

Chenxu Wang et al. *ICL-Router: In-Context Learned Model Representations for LLM Routing*. AAAI 2026.

Analyses from the Perspective of LLM Cost

Does there exist an integer n such that $n^2 + n + 41$ is a composite number?
If yes, please provide the smallest non-negative n as evidence.

Phi-2 (≈2.7B)

Direct Output:
No, this expression is always a prime number.

Correct: ❌

Length： ≈6–10 tokens

Mistral 7B Instruct (≈7B)

Self-Consistency Output:
Sample1:n=41;
Sample2:n=40;
Sample3:n=41;
Majority vote:n=41(2/3);
Final answer: Yes,n=41

Correct： ❌

Length： ≈25–40 tokens

Qwen2.5-14B-Instruct (≈14B)

CoT Output: Substitute n=40: $n^2 + n + 41 = 1681 = 41^2$ (composite); Classic results show primes for $0 \leq n < 40$→smallest n=40

Correct： ✅

Length： ≈40–60 tokens

GPT-4.1

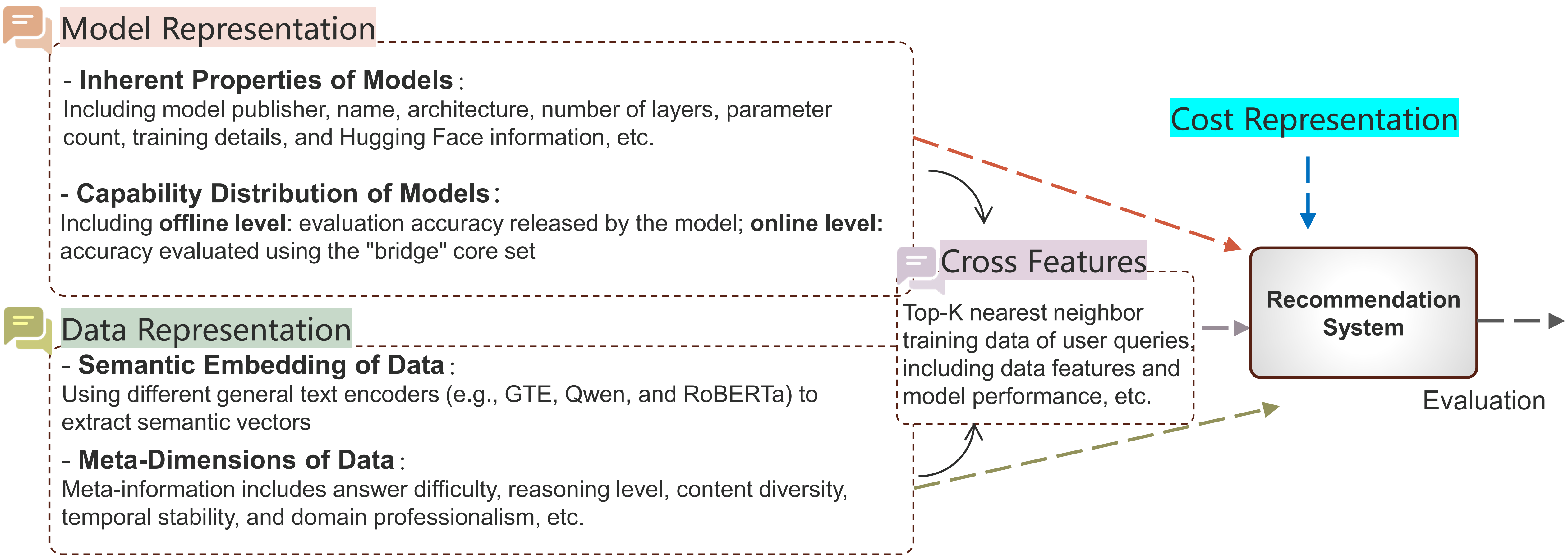
ToT Output: Layer0: Search neighborhood of n≈40→Branch A(n=39→prime)、Branch B(n=40→composite, candidate smallest)→Conclusion: Yes, smallest n=40

Correct： ✅

Length： ≈80–130 tokens

LLMs have different costs when solving complex tasks.
How to ***balance model capability and cost*** when selecting LLMs?

Fusing Model & Data Representation



Model Representation

- **Inherent Properties of Models**:
Including model publisher, name, architecture, number of layers, parameter count, training details, and Hugging Face information, etc.

- **Capability Distribution of Models**:
Including **offline level**: evaluation accuracy released by the model; **online level:** accuracy evaluated using the "bridge" core set

Data Representation

- **Semantic Embedding of Data**:
Using different general text encoders (e.g., GTE, Qwen, and RoBERTa) to extract semantic vectors

- **Meta-Dimensions of Data**:
Meta-information includes answer difficulty, reasoning level, content diversity, temporal stability, and domain professionalism, etc.

Cross Features

Top-K nearest neighbor training data of user queries, including data features and model performance, etc.

Cost Representation

Recommendation System

Evaluation

Transform the model selection problem into a representation-based “model recommendation” problem, and learn how to rank LLMs under different costs

Yi-Kai Zhang, Shiyin Lu, Qing-Guo Chen, Weihua Luo, De-Chuan Zhan, Han-Jia Ye. *Let the LLM Stick to Its Strengths: Learning to Route Economical LLM*. NeurIPS 2025.

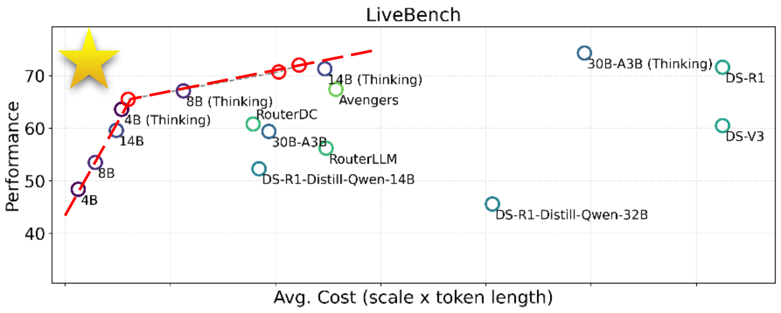
Fusing Model & Data Representation

LLM Zoo

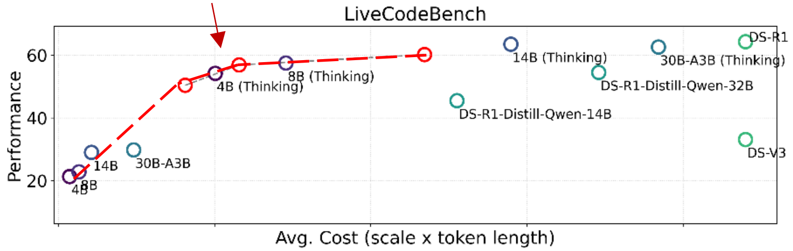
Qwen3-4B,
Qwen3-8B,
Qwen3-14B,
Qwen3-30B-A3B

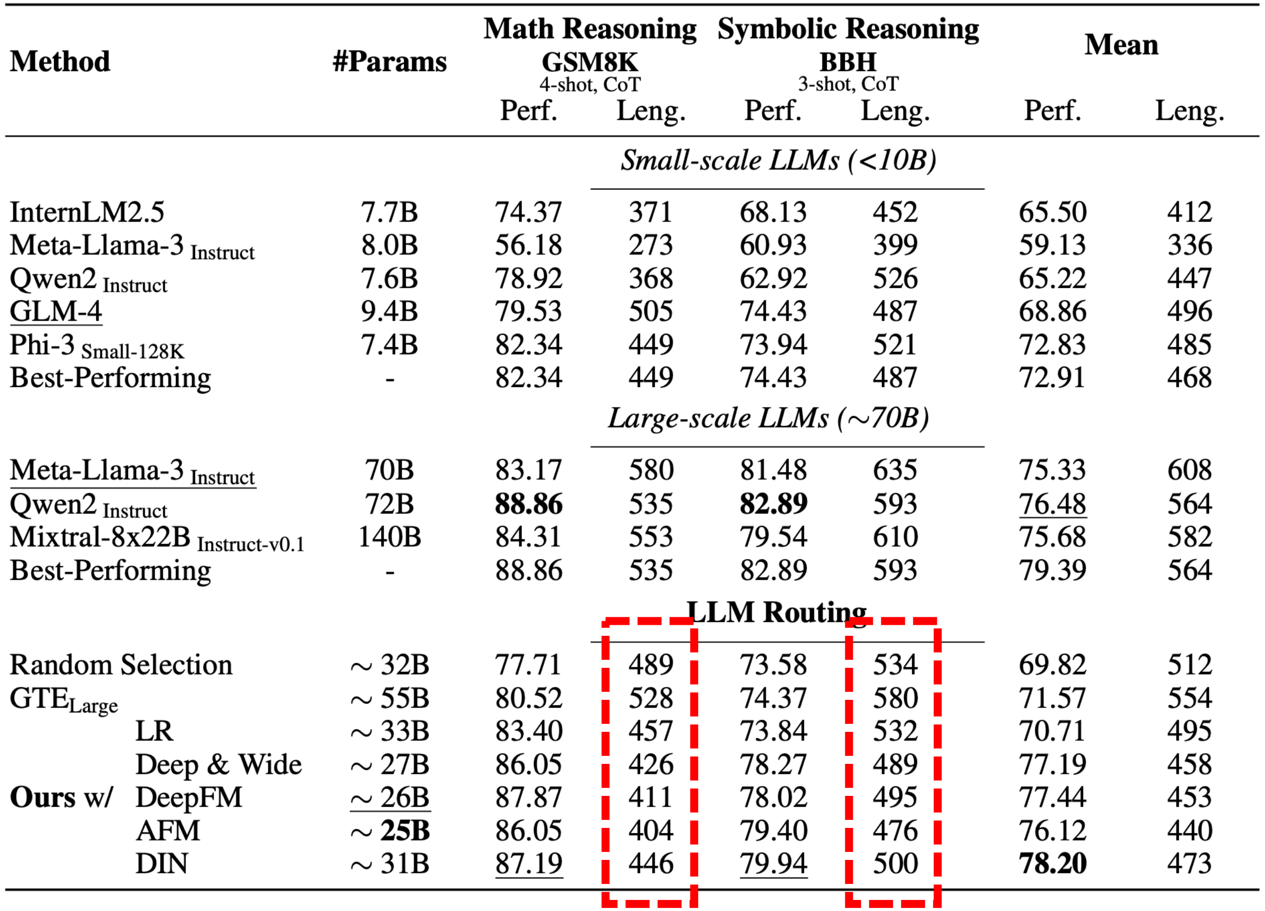
two versions w/
thinking and w/o
thinking

Oracle (w/ low cost & high perf.)



Pareto Frontier of
Performance-Cost





Method	#Params	Math Reasoning GSM8K 4-shot, CoT Perf.	Math Reasoning GSM8K 4-shot, CoT Leng.	Symbolic Reasoning BBH 3-shot, CoT Perf.	Symbolic Reasoning BBH 3-shot, CoT Leng.	Mean Perf.	Mean Leng.
			Small-scale LLMs (<10B)				
InternLM2.5	7.7B	74.37	371	68.13	452	65.50	412
Meta-Llama-3 Instruct	8.0B	56.18	273	60.93	399	59.13	336
Qwen2 Instruct	7.6B	78.92	368	62.92	526	65.22	447
GLM-4	9.4B	79.53	505	74.43	487	68.86	496
Phi-3 Small-128K	7.4B	82.34	449	73.94	521	72.83	485
Best-Performing	-	82.34	449	74.43	487	72.91	468
			Large-scale LLMs (∼70B)				
Meta-Llama-3 Instruct	70B	83.17	580	81.48	635	75.33	608
Qwen2 Instruct	72B	**88.86**	535	**82.89**	593	76.48	564
Mixtral-8x22B Instruct-v0.1	140B	84.31	553	79.54	610	75.68	582
Best-Performing	-	88.86	535	82.89	593	79.39	564
			LLM Routing				
Random Selection	∼ 32B	77.71	489	73.58	534	69.82	512
$\text{GTE}_{\text{Large}}$	∼ 55B	80.52	528	74.37	580	71.57	554
LR	∼ 33B	83.40	457	73.84	532	70.71	495
Deep & Wide	∼ 27B	86.05	426	78.27	489	77.19	458
Ours w/ DeepFM	∼ 26B	87.87	411	78.02	495	77.44	453
AFM	∼ **25B**	86.05	404	79.40	476	76.12	440
DIN	∼ 31B	87.19	446	79.94	500	**78.20**	473

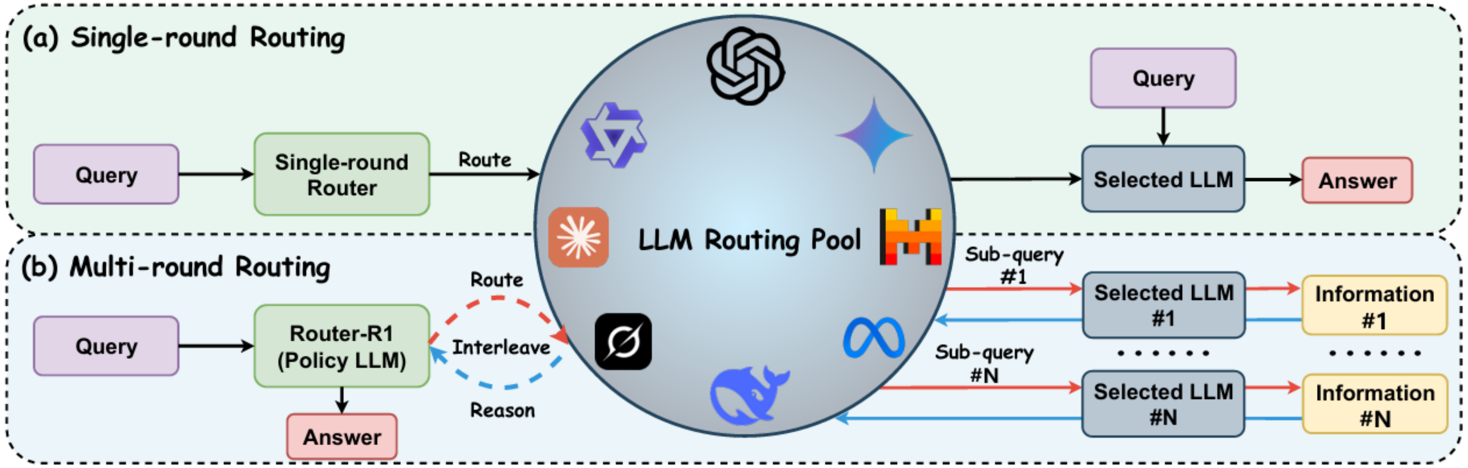
Better Accuracy Under Equal Cost

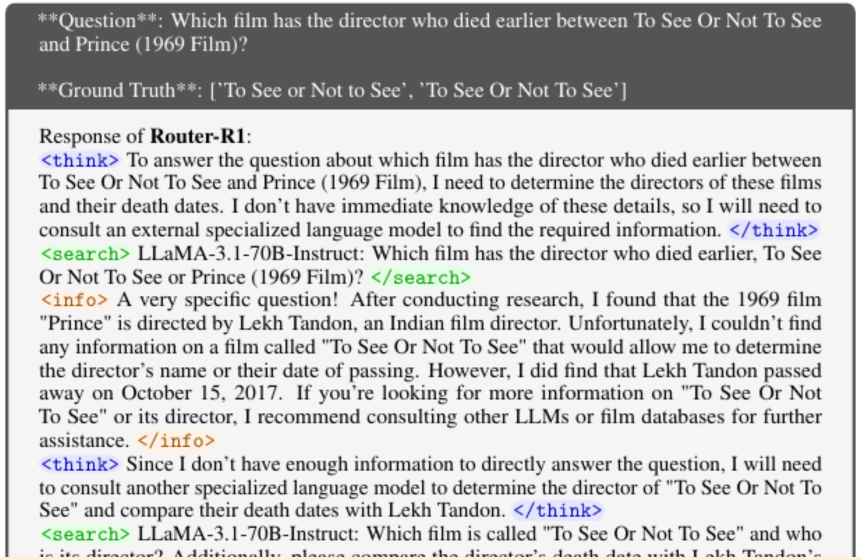
Balance Between Reasoning Accuracy & Cost:

Incorporate prior information (e.g., cost) as an input dimension to add manual control options for model selection

Router-R1

Router-R1 formulates LLM routing as a sequential decision-making problem.




Question: Which film has the director who died earlier between To See Or Not To See and Prince (1969 Film)?

Ground Truth: ['To See or Not to See', 'To See Or Not To See']

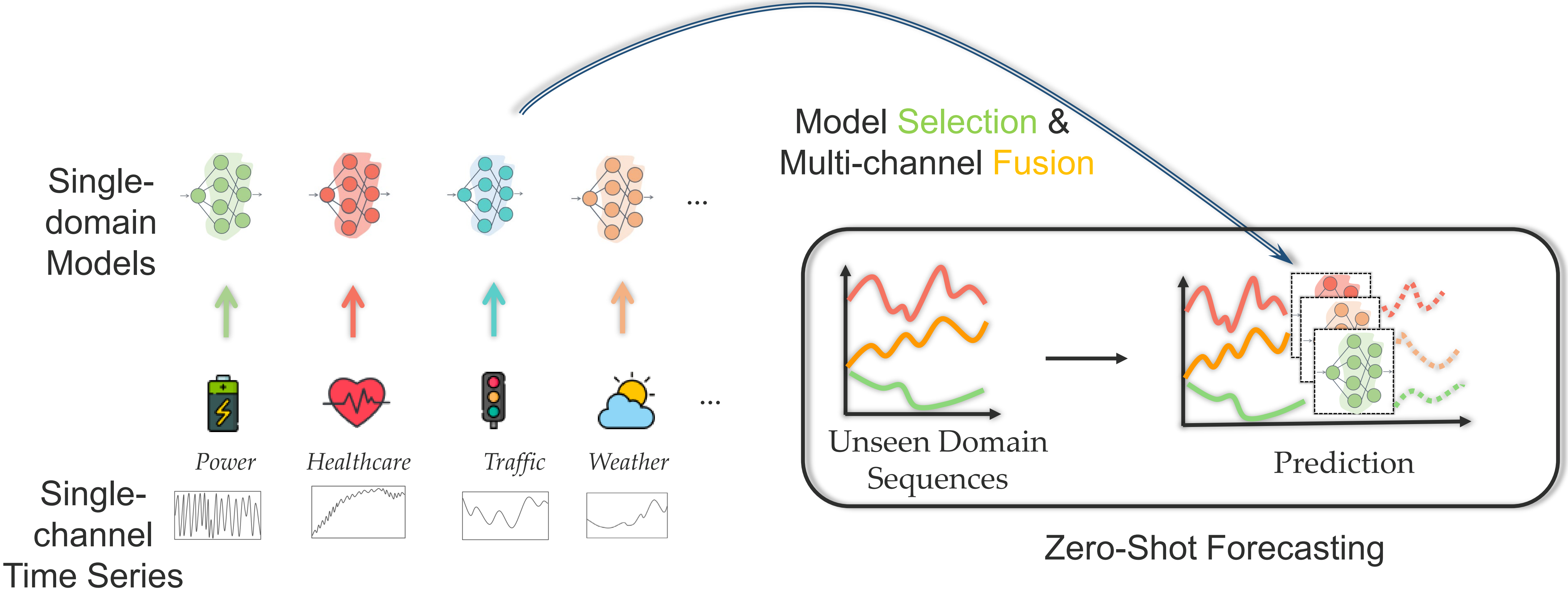
Response of **Router-R1**:
<think> To answer the question about which film has the director who died earlier between To See Or Not To See and Prince (1969 Film), I need to determine the directors of these films and their death dates. I don't have immediate knowledge of these details, so I will need to consult an external specialized language model to find the required information. </think>
<search> LLaMA-3.1-70B-Instruct: Which film has the director who died earlier, To See Or Not To See or Prince (1969 Film)? </search>
<info> A very specific question! After conducting research, I found that the 1969 film "Prince" is directed by Lekh Tandon, an Indian film director. Unfortunately, I couldn't find any information on a film called "To See Or Not To See" that would allow me to determine the director's name or their date of passing. However, I did find that Lekh Tandon passed away on October 15, 2017. If you're looking for more information on "To See Or Not To See" or its director, I recommend consulting other LLMs or film databases for further assistance. </info>
<think> Since I don't have enough information to directly answer the question, I will need to consult another specialized language model to determine the director of "To See Or Not To See" and compare their death dates with Lekh Tandon. </think>
<search> LLaMA-3.1-70B-Instruct: Which film is called "To See Or Not To See" and who

- Represent the router as a capable LLM that alternates "think" actions (internal reasoning) and "route" actions (invoking external LLMs).
- Train the router with reinforcement learning ***using a reward function that combines format correctness, final task success, and a cost penalty***, encouraging both accuracy and efficiency.

- Multi-round routing allows the router to iteratively perform internal reasoning, *dynamically choose* which LLMs to query, and aggregate partial outputs over several steps.
- Enables the system to handle complex, multi-step tasks more effectively, refine decisions based on intermediate results, and balance performance with computational cost.

Haozhen Zhang, Tao Feng, Jiaxuan You. *Router-R1: Teaching LLMs Multi-Round Routing and Aggregation via Reinforcement Learning*, NeurIPS 2025.

Selection of Time-Series Forecasting Models



Single-domain Models

…

Single-channel Time Series

Power

Healthcare

Traffic

Weather

…

Model Selection & Multi-channel Fusion

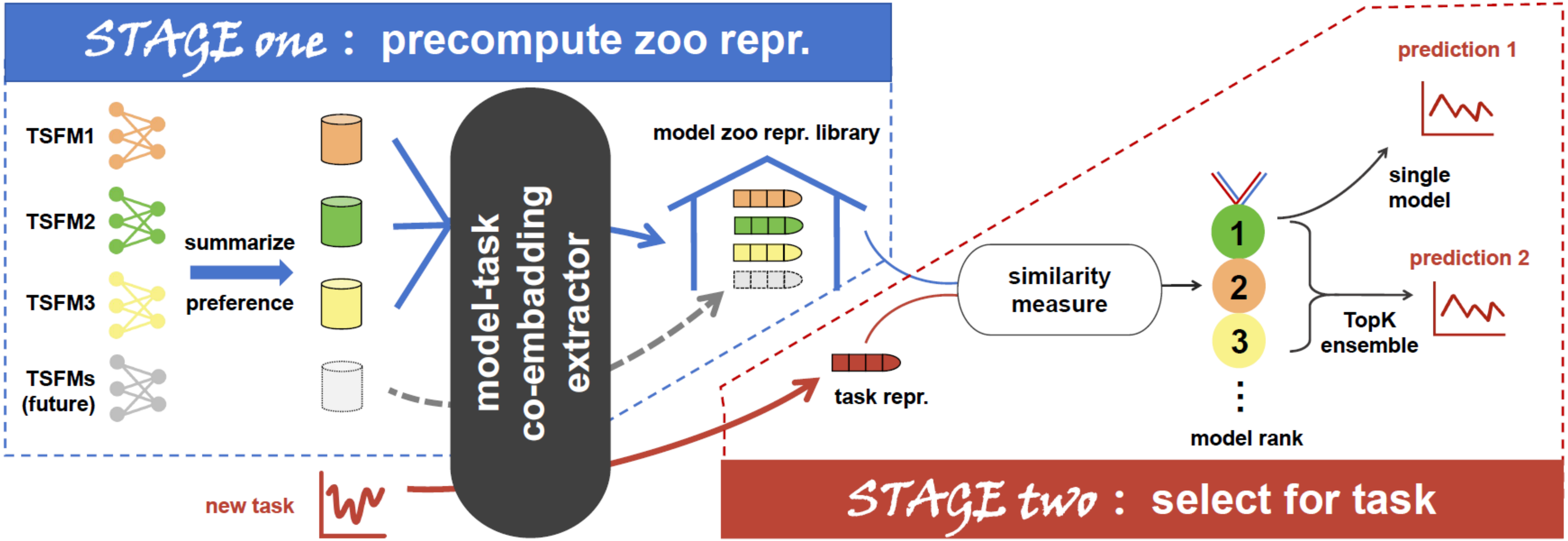
Unseen Domain Sequences

Prediction

Zero-Shot Forecasting

Ting-Ji Huang, Xu-Yang Chen, Han-Jia Ye. *SeqFusion: Sequential Fusion of Pre-Trained Models for Zero-Shot Time-Series Forecasting*. CoRR 2025.

Selection of Time-Series Foundation Models



1. Train a general representation model (General Extractor) to generate sequence meta-representations, using sequence reconstruction self-supervision + transferability loss for training.

2. Based on model meta-representations and sequence meta-representations, perform ranking and selection of models according to their domain adaptability on the current sequence.

Hao-Nan Shi, Ting-Ji Huang, Lu Han, De-Chuan Zhan, Han-Jia Ye. *One-Embedding-Fits-All: Efficient Zero-Shot Time Series Forecasting by a Model Zoo*. CoRR 2025.

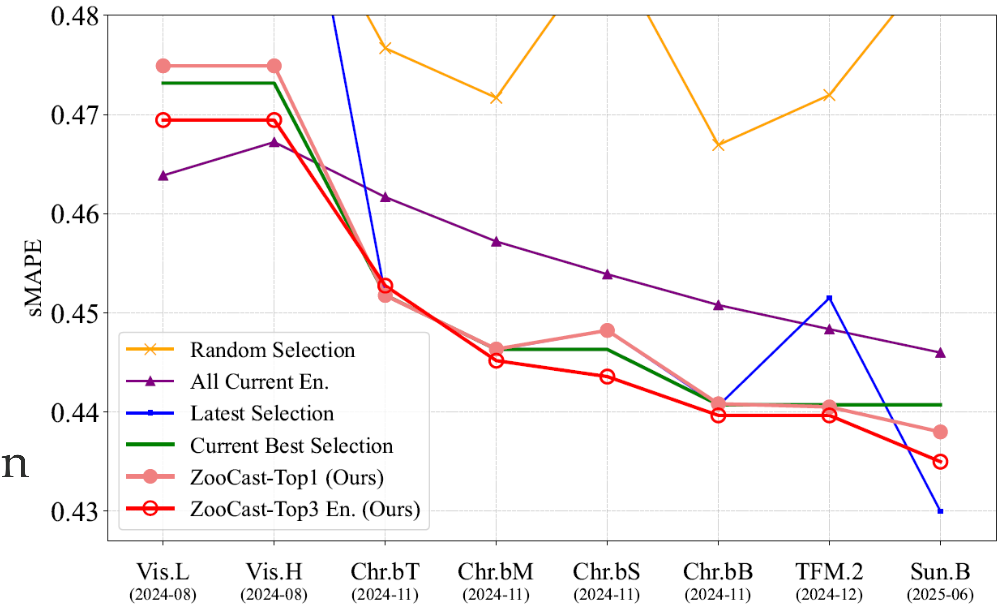
Selection of Time-Series Foundation Models

GIFT-Eval experimental results, including 23 large-scale time-series datasets and 97 configurations

	Single Model Prediction													Ensemble Prediction		
Metric	Chr.bT	Chr.bM	Chr.bS	Chr.bB	Moi.S	Moi.B	Moi.L	TFM.1	TFM.2	Vis.B	Vis.L	Vis.H	Sun.B	All-13	Top-3 Z.C.(**ours**)	Top-5 Z.C.(**ours**)
sMAPE	0.452	0.446	0.448	0.441	0.488	0.474	0.474	0.474	0.452	0.513	0.512	0.511	**0.430**	0.445	0.437	0.431
Rank	6.856	5.753	6.113	4.856	9.753	8.371	8.031	8.598	4.949	11.381	11.258	10.979	4.845	5.062	3.688	**3.158**

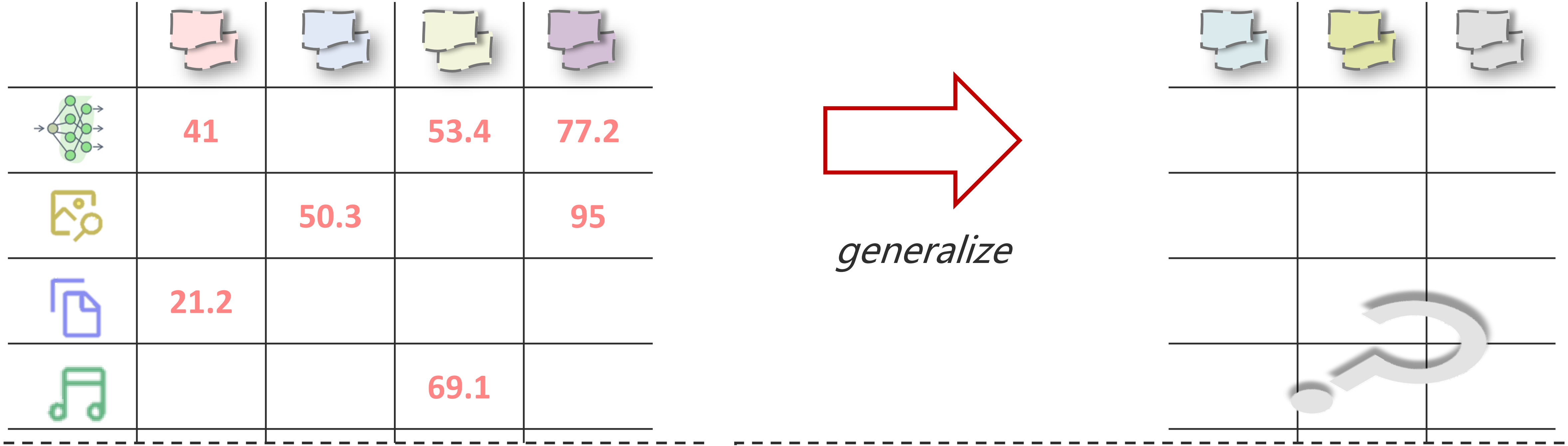
The time-series model zoo includes the following foundation models: Chronos (Chr.bT/bM/bS/bB), Moirai (Moi.S/B/L), VisionTS (Vis.B/L/H), TimesFM (TFM.1/2), Sundial (Sun.B).

Time-series foundation models emerge dynamically →



Learning-Based Methods

Main idea: automatically learn the specifications in a data-driven manner



	41		53.4	77.2
		50.3		95
	21.2			
			69.1	

- Automate the specification learning process.
- Usually achieve better PTM selection performance.
- Enable scalability by adapting to new models and tasks in an end-to-end manner.

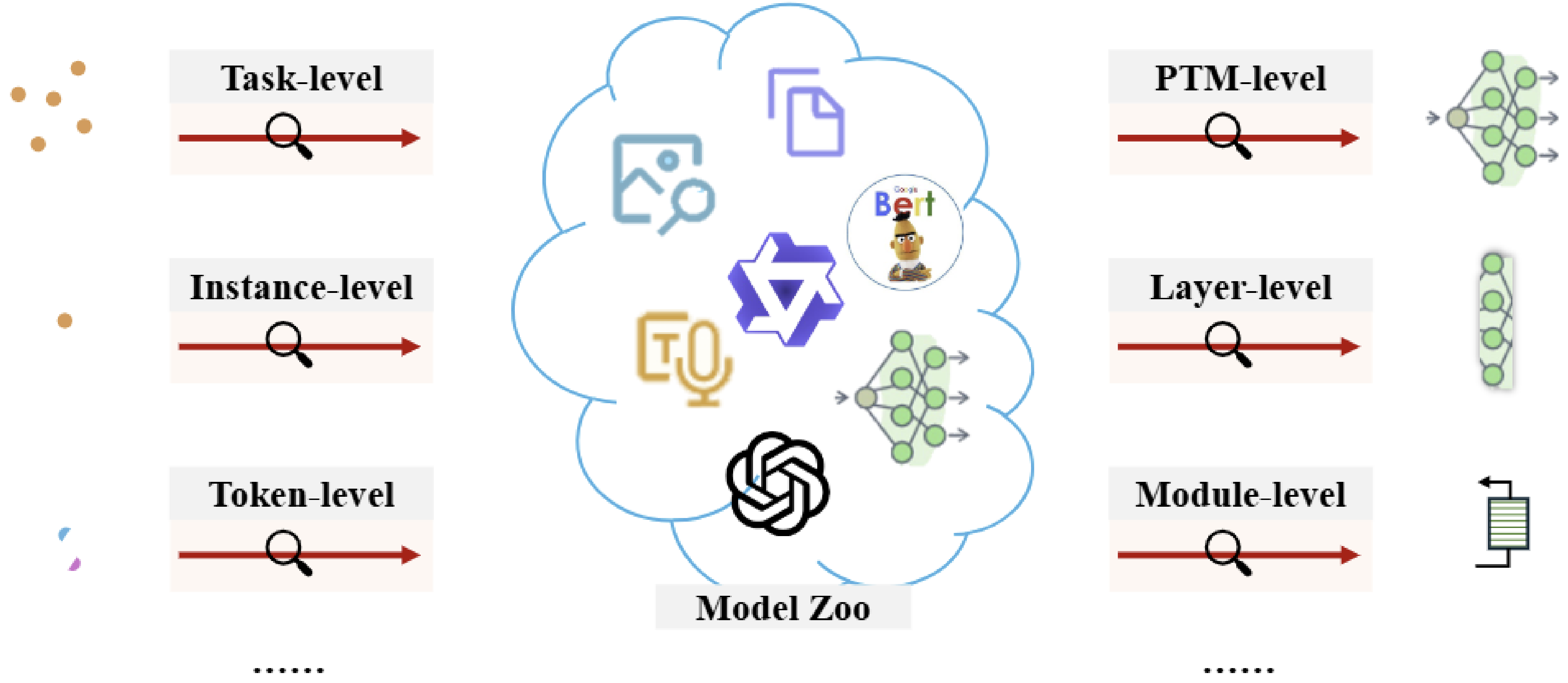
- The effectiveness heavily depends on the quality, diversity, and the scale of the training data.
- The process of training and deploying these systems, especially for large models, can be resource-intensive, requiring substantial computational power and time.

Related Topics

Multi-Level Selection

Rather than treating each PTM as a whole, model reuse can operate at a finer granularity by assembling models from components such as network blocks or parameter-efficient tuning modules.

Queries and targets over different levels and granularities.



Task-level

Instance-level

Token-level

......

Model Zoo

PTM-level

Layer-level

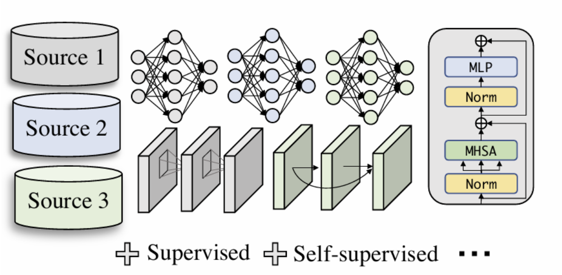
Module-level

......

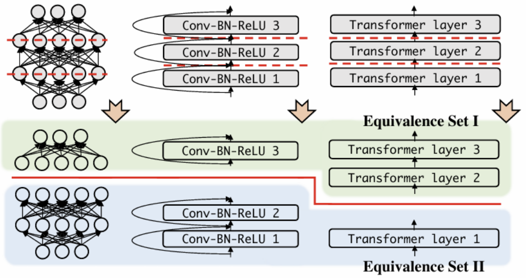
Model Assembly

Enable fine-grained model selection targets, where reusable parts are identified and recombined to build models tailored to specific tasks.

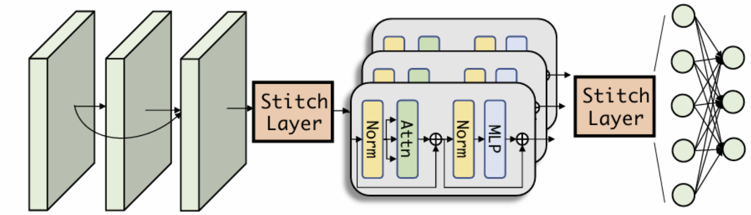
Deep Model Reassembly



Model Zoo

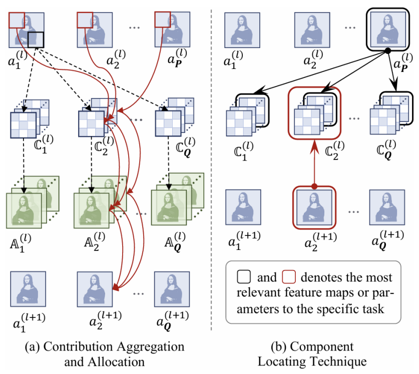


Network Partition

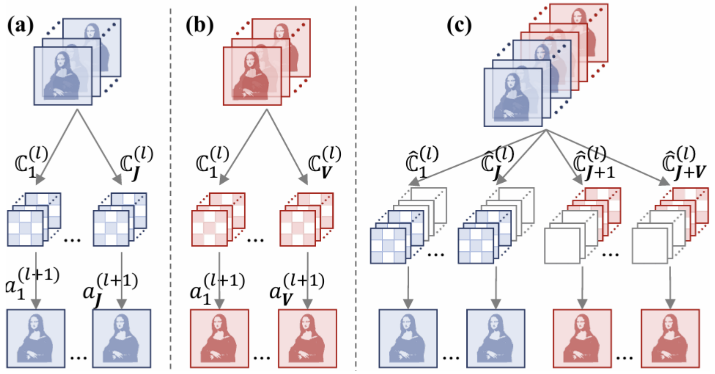


Network Reassembly

Model LEGO



Disassembling process

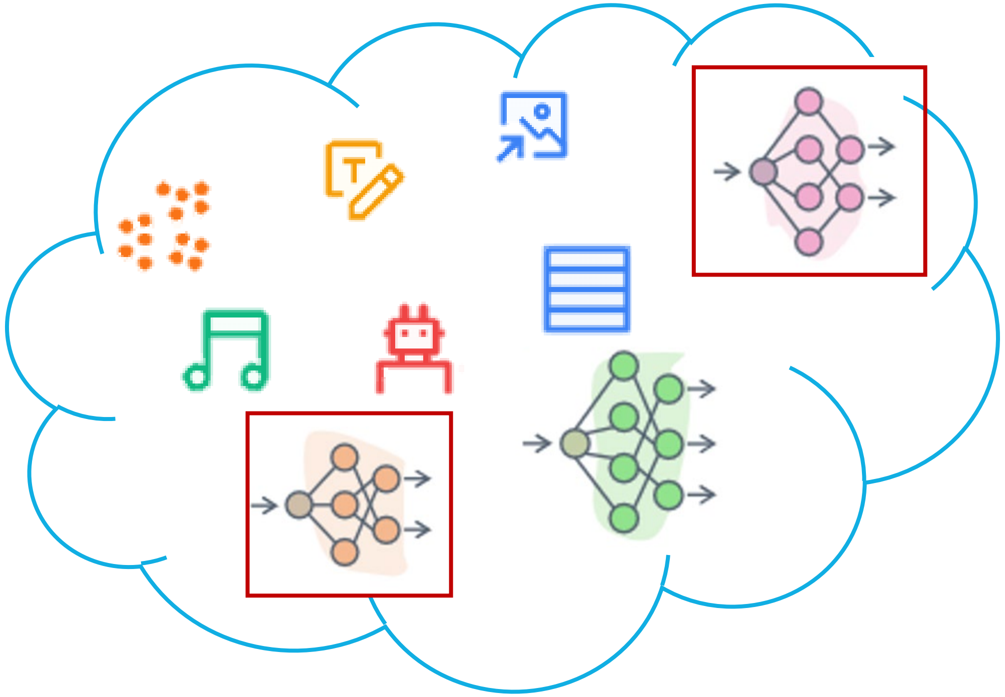


Assembling process

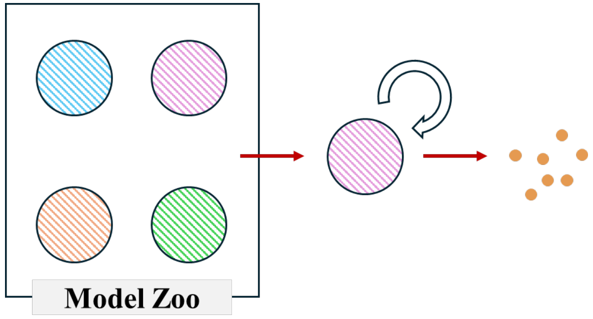
Xingyi Yang, Daquan Zhou, Songhua Liu, Jingwen Ye, Xinchao Wang. *Deep Model Reassembly*. NeurIPS 2022
Jiacong Hu et al. *Model LEGO: Creating Models Like Disassembling and Assembling Building Blocks*. NeurIPS 2024

Multi-PTM Selection

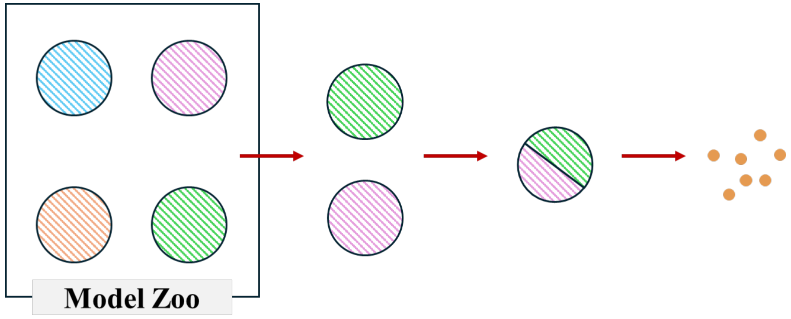
Selecting a set of PTMs may not equal to selecting top-K PTMs.



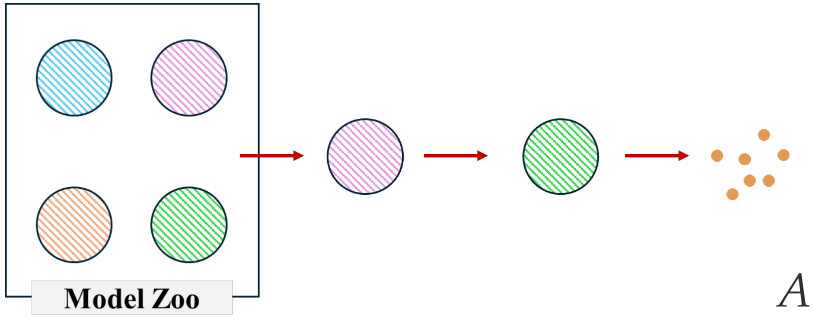
Obtaining the ground-truth of the model set is a combinatorial optimization problem.



(a) Single PTM Selection



(b) Parallel Multi-PTM Selection



(c) Sequential Multi-PTM Selection

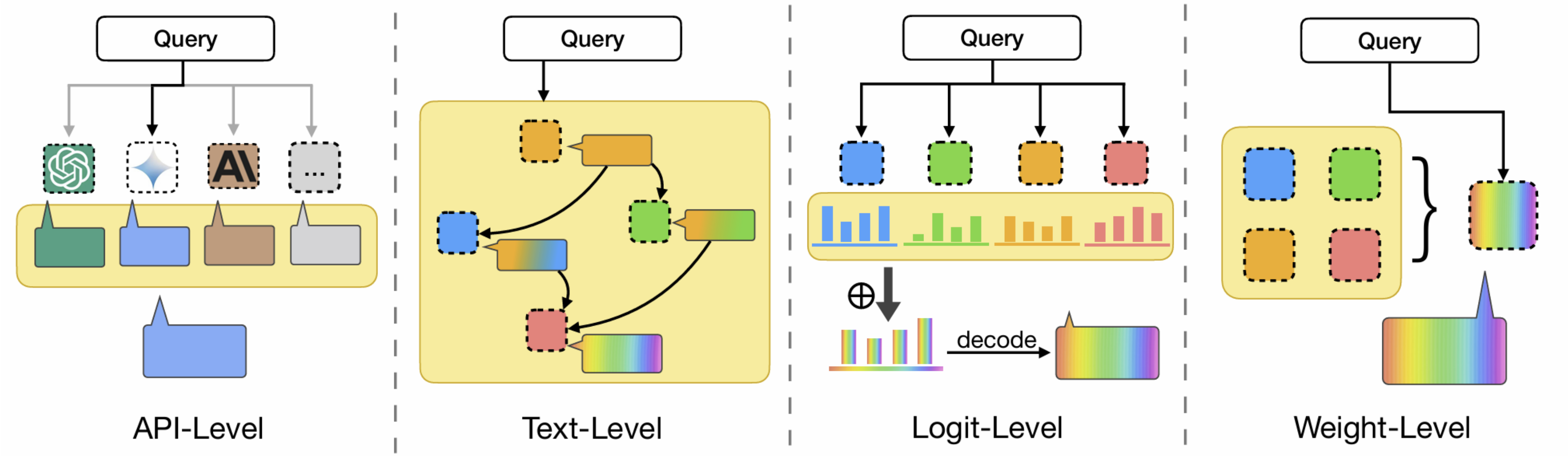
Also depend on the usage of multiple PTMs.

Andrea Agostinelli, Jasper R. R. Uijlings, Thomas Mensink, Vittorio Ferrari. *Transferability Metrics for Selecting Source Model Ensembles*. CVPR 2022
Vimal K. B., et al. *Building a Winning Team: Selecting Source Model Ensembles using a Submodular Transferability Estimation Approach*. ICCV 2023

Model Collaboration

Instead of selecting a single PTM, aggregating multiple PTMs (*with zero-shot ability*) to collaborate in parallel or sequentially can enhance model robustness and accuracy.

Complex target tasks can often be decomposed into multiple sub-tasks, each handled by a different PTM with distinct expertise.

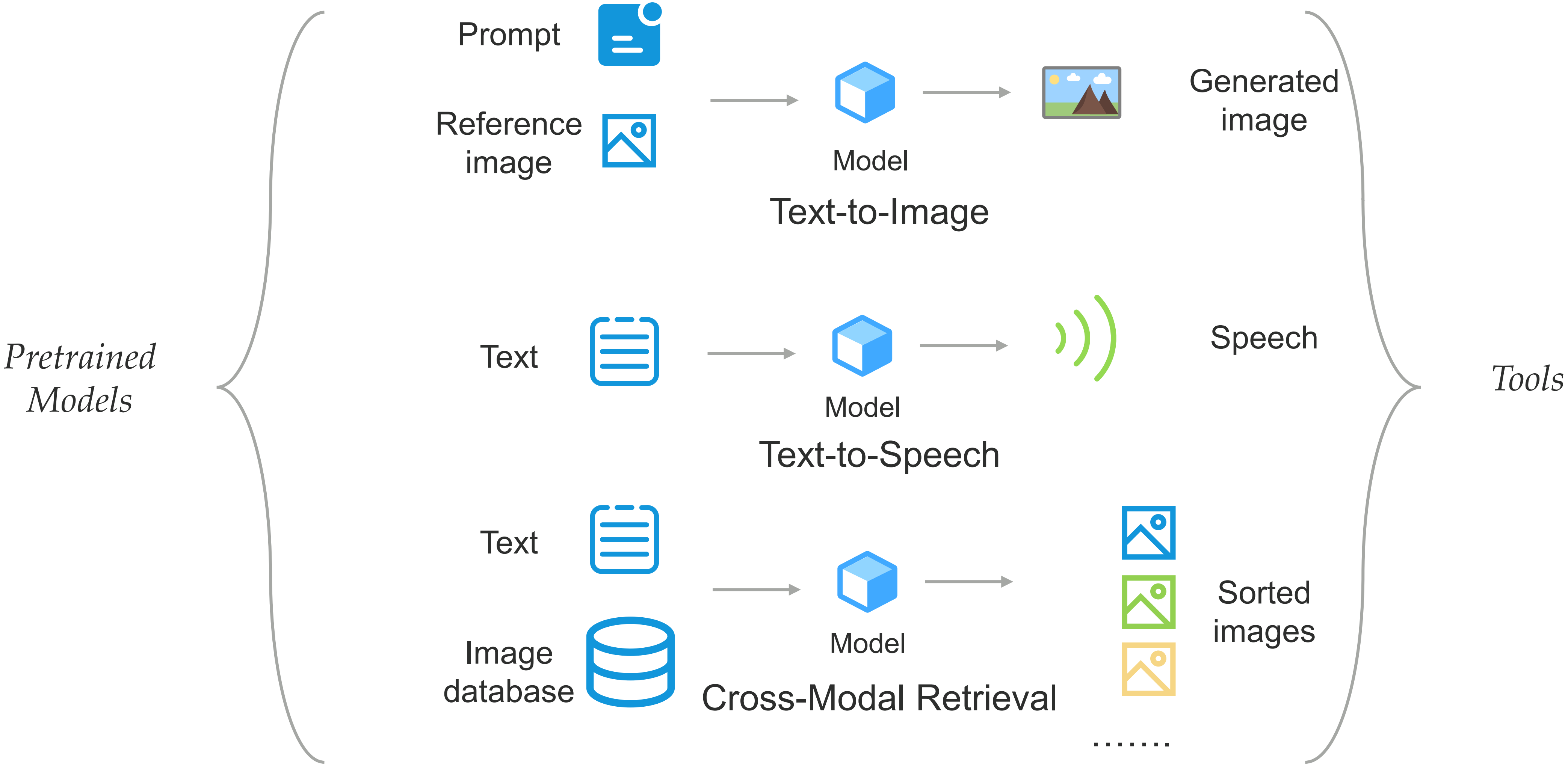


- ***Sequential collaboration***: *small models can tackle easy tasks, while large models can tackle what small models cannot handle. Furthermore, some tasks can be decomposed into sub-tasks, where different PTMs handle each step sequentially.*
- ***Parallel collaboration***: *multiple PTMs contribute jointly to task performance through techniques such as weighted averaging or model merging*

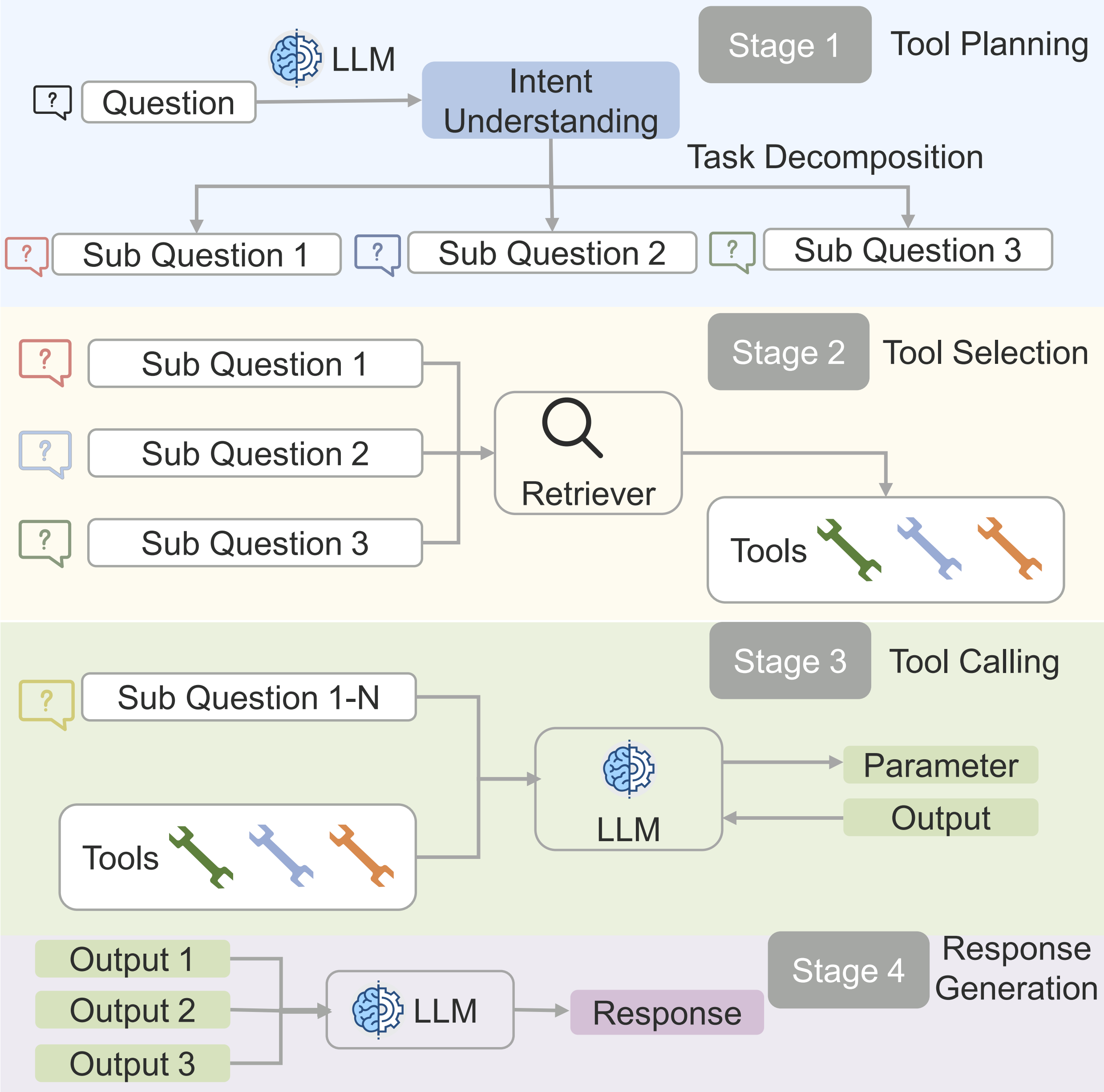
Lihu Chen, Gaël Varoquaux. *What is the Role of Small Models in the LLM Era: A Survey*. CoRR 2024
Shangbin Feng, et al. *When One LLM Drools, Multi-LLM Collaboration Rules*. CoRR 2025

Tool Learning

Pre-trained models form individual tools through specific workflows.



Tool Learning



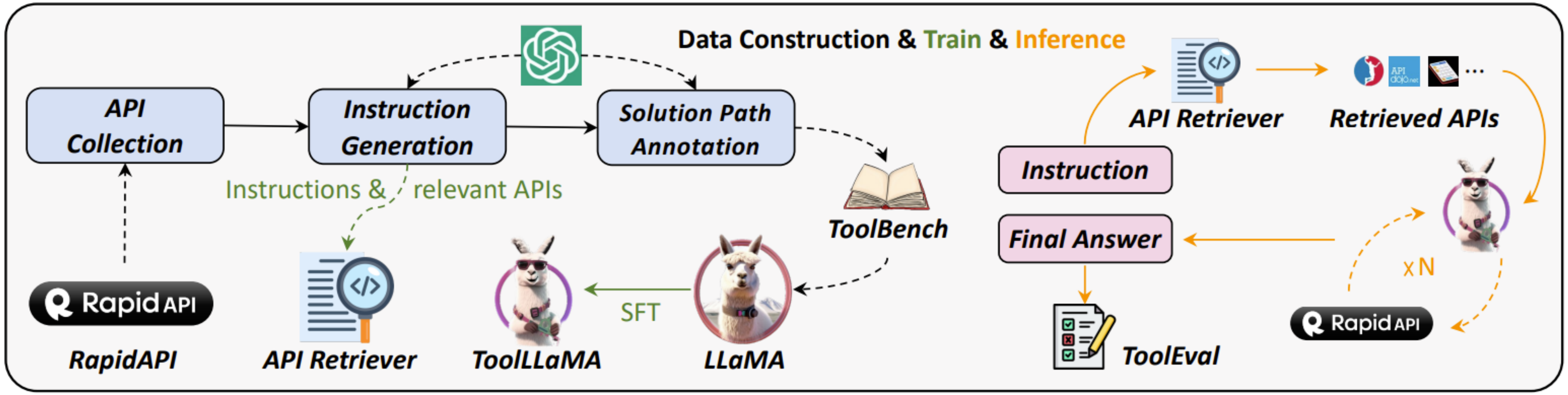
In the field of tool learning, the underlying implementation of many tools is based on deployed general-purpose models.

Consequently, the boundary between tools and models is becoming increasingly blurred.

In the traditional four-stage setting of tool learning, ***tool selection*** *and* ***tool calling*** *can be analogized to* ***model selection*** *and* ***hyperparameter setting****, respectively.*

Tool Learning

ToolLLM systematically addresses the issues of insufficient instruction diversity and low robustness of complex calls by constructing the ToolBench dataset covering 16,000 real APIs. This has successfully unlocked the significant potential of open-source models to match the performance of ChatGPT in zero-sample tool invocation scenarios.



Data Construction & Train & Inference

API Collection

Instruction Generation

Solution Path Annotation

Instructions & relevant APIs

ToolBench

RapidAPI

API Retriever

ToolLLaMA

SFT

LLaMA

API Retriever

Retrieved APIs

Instruction

Final Answer

ToolEval

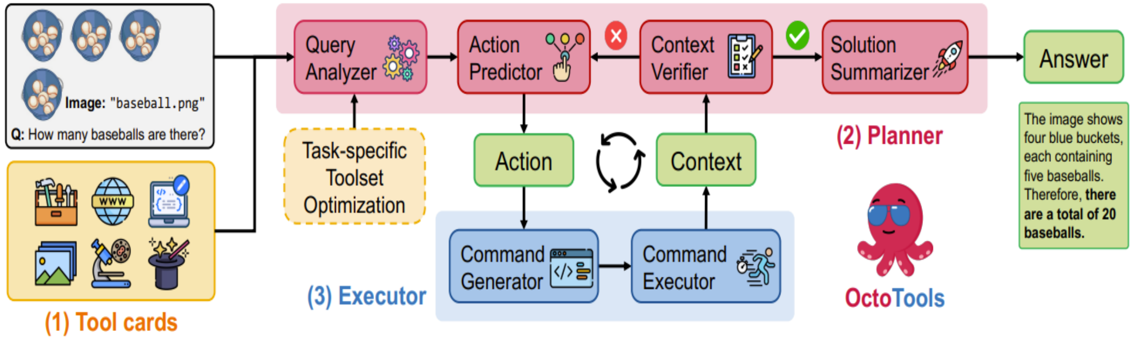
x N

The model relies on a static training set, whereas real-world APIs are dynamic and prone to frequent interface changes or deprecations.

Yujia Qin, Shihao Liang, Yining Ye, et al. *ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs*. ICLR 2024.

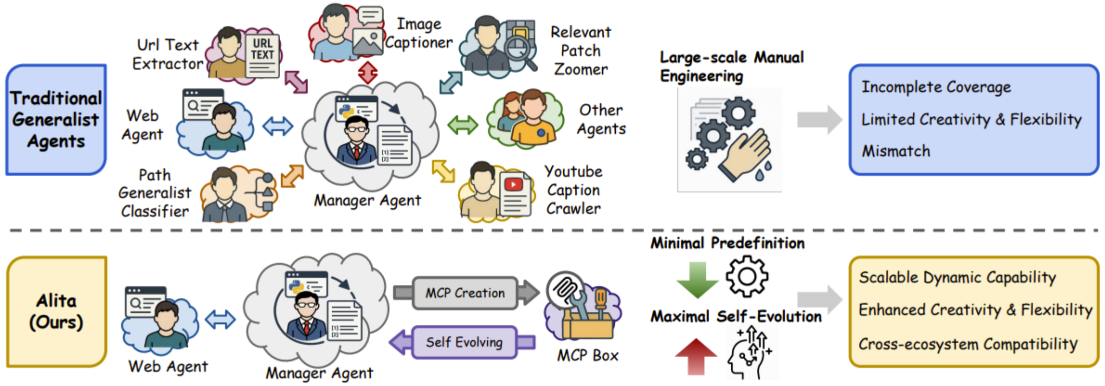
Tool Learning

OctoTools is a training-free agent framework. It achieves zero-cost immediate expansion of tools through standardized "Tool Cards", and with a "planner-executor" dual-layer architecture and an automated toolset optimization algorithm.



It enhances the performance of LLMs.

The Alita framework achieves the self-evolution capability by independently writing code and encapsulating it using the MCP protocol into new tools.



This enables the generation and maintenance of an exclusive subset of tools that can be dynamically created based on task requirements.

Pan Lu, Bowen Chen, Sheng Liu, Rahul Thapa, Joseph Boen, James Zou. *OctoTools: An Agentic Framework with Extensible Tools for Complex Reasoning*. CoRR 2025.
Jiahao Qiu, Xuan Qi, Tongcheng Zhang, et al. *ALITA: Generalist Agent Enabling Scalable Agentic Reasoning with Minimal Predefinition and Maximal Self-Evolution*. CoRR 2025.

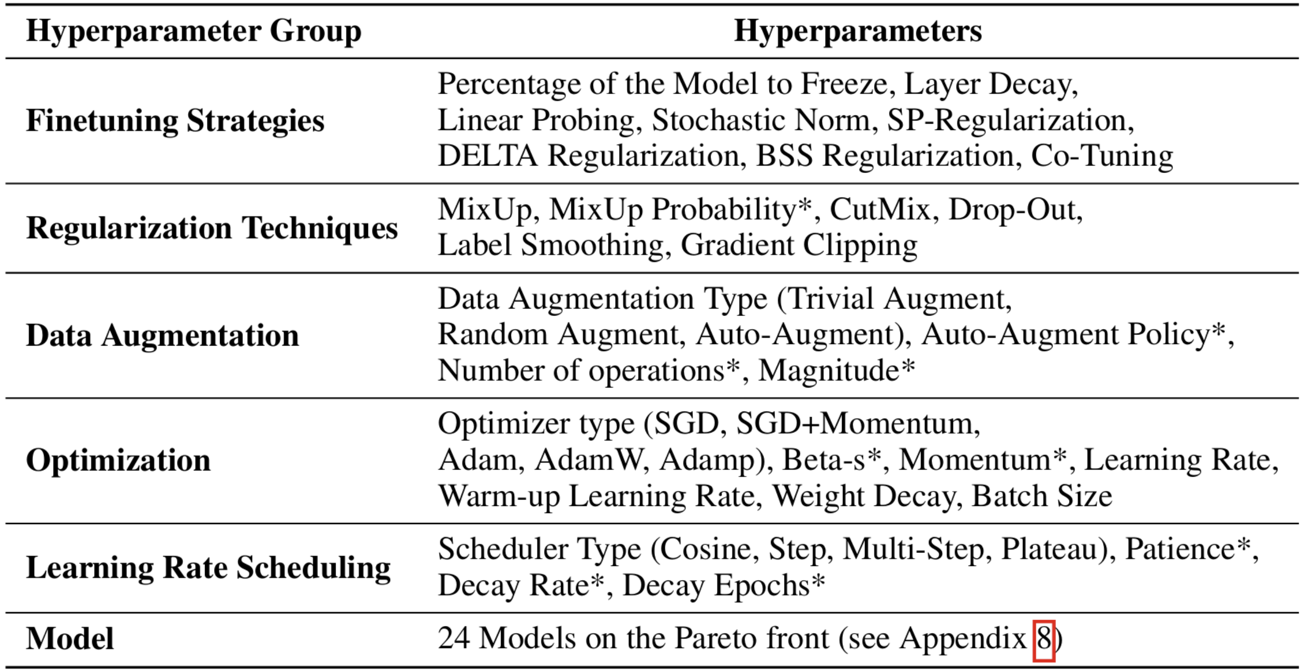
Discussions

Coupling of Selection and Adaptation

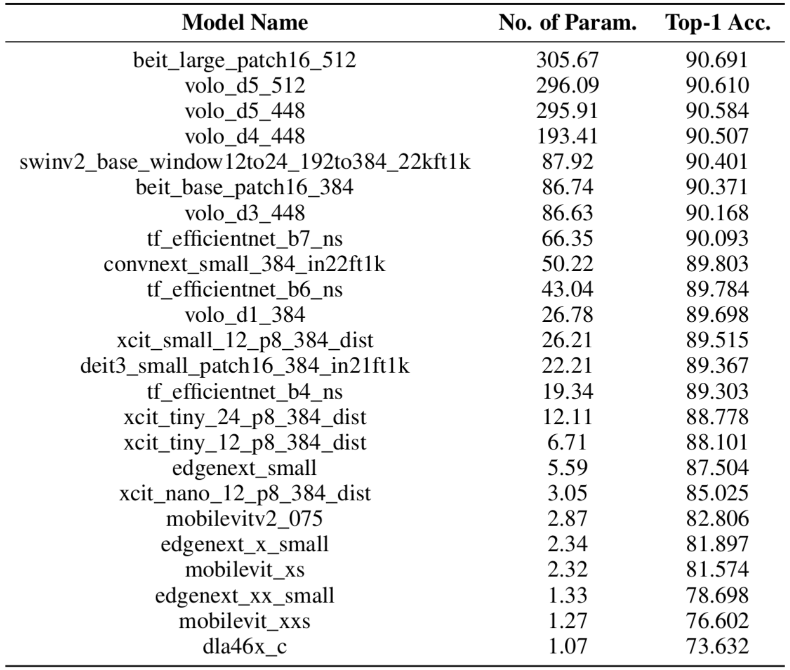
The generalization ability of an adapted model depends not only on the selected PTM but also on the adaptation strategy employed. Thus, PTM selection and adaptation are inherently ***coupled***

While some methods, such as transferability estimation, attempt to improve selection, they often fail to account for the mismatches between fine-tuning and advanced adaptation techniques.

Search both the PTM and corresponding fine-tuning strategies (hyperparameters) with meta-learning.



Hyperparameter Group	**Hyperparameters**
Finetuning Strategies	Percentage of the Model to Freeze, Layer Decay, Linear Probing, Stochastic Norm, SP-Regularization, DELTA Regularization, BSS Regularization, Co-Tuning
Regularization Techniques	MixUp, MixUp Probability*, CutMix, Drop-Out, Label Smoothing, Gradient Clipping
Data Augmentation	Data Augmentation Type (Trivial Augment, Random Augment, Auto-Augment), Auto-Augment Policy*, Number of operations*, Magnitude*
Optimization	Optimizer type (SGD, SGD+Momentum, Adam, AdamW, Adamp), Beta-s*, Momentum*, Learning Rate, Warm-up Learning Rate, Weight Decay, Batch Size
Learning Rate Scheduling	Scheduler Type (Cosine, Step, Multi-Step, Plateau), Patience*, Decay Rate*, Decay Epochs*
Model	24 Models on the Pareto front (see Appendix 8)

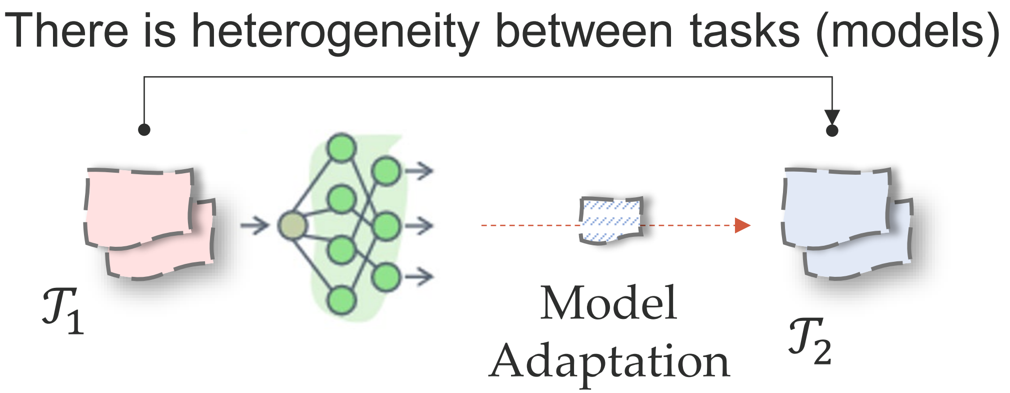


Model Name	**No. of Param.**	**Top-1 Acc.**
beit_large_patch16_512	305.67	90.691
volo_d5_512	296.09	90.610
volo_d5_448	295.91	90.584
volo_d4_448	193.41	90.507
swinv2_base_window12to24_192to384_22kft1k	87.92	90.401
beit_base_patch16_384	86.74	90.371
volo_d3_448	86.63	90.168
tf_efficientnet_b7_ns	66.35	90.093
convnext_small_384_in22ft1k	50.22	89.803
tf_efficientnet_b6_ns	43.04	89.784
volo_d1_384	26.78	89.698
xcit_small_12_p8_384_dist	26.21	89.515
deit3_small_patch16_384_in21ft1k	22.21	89.367
tf_efficientnet_b4_ns	19.34	89.303
xcit_tiny_24_p8_384_dist	12.11	88.778
xcit_tiny_12_p8_384_dist	6.71	88.101
edgenext_small	5.59	87.504
xcit_nano_12_p8_384_dist	3.05	85.025
mobilevitv2_075	2.87	82.806
edgenext_x_small	2.34	81.897
mobilevit_xs	2.32	81.574
edgenext_xx_small	1.33	78.698
mobilevit_xxs	1.27	76.602
dla46x_c	1.07	73.632

Sebastian Pineda Arango, Fabio Ferreira, Arlind Kadra, Frank Hutter, Josif Grabocka. *Quick-Tune: Quickly Learning Which Pretrained Model to Finetune and How*. ICLR 2024.

The Necessity of the Adaptation Stage

Most adaptation techniques are task-specific and require extensive hyper-parameter tuning, making them difficult to generalize. Lacking a universal adaptation method slows down the development. Most modern PTMs have the zero-shot ability which generalizes to new tasks without additional training. This property decouples the selection and adaptation stages, but may also weaken the ability of the selected PTMs.



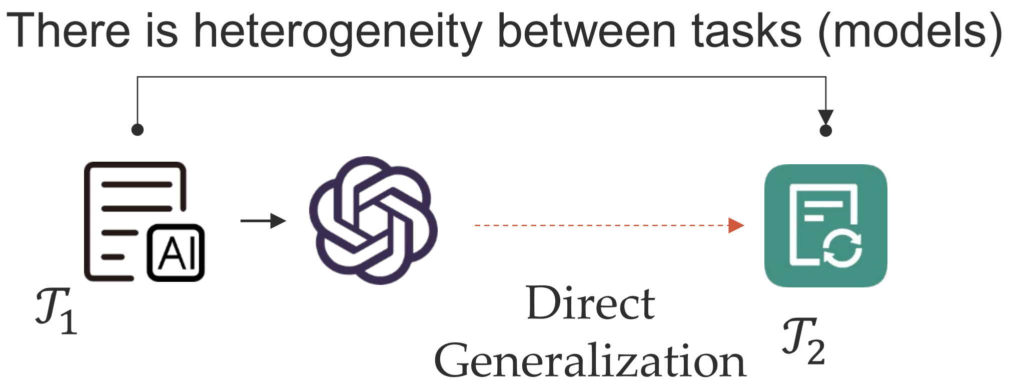
There is heterogeneity between tasks (models)

$\mathcal{T}_1$

Model Adaptation

$\mathcal{T}_2$

w/ an explicit *adaptation* phase



There is heterogeneity between tasks (models)

$\mathcal{T}_1$

Direct Generalization

$\mathcal{T}_2$

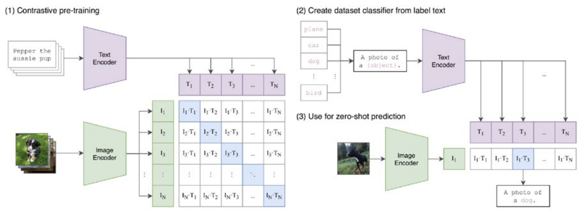
w/ *zero-shot generalization* ability

What is the right way to adapt a selected (foundation) PTM and further boost its ability?
Such as in-context-learning, model reprogramming, etc.

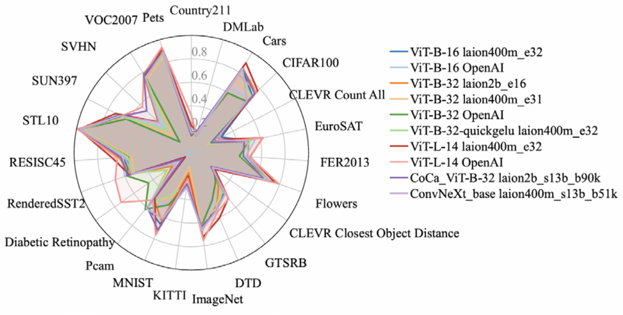
Diversity and Strength of the Model Zoo

The success of model reuse depends on the quality, diversity, and generalization of PTMs in the model zoo. A well-curated collection improves the chance of selecting suitable models—even at fine-grained levels such as instances or tokens. Therefore, improving the curation and management of model zoos is essential for maximizing the impact and utility of model reuse.

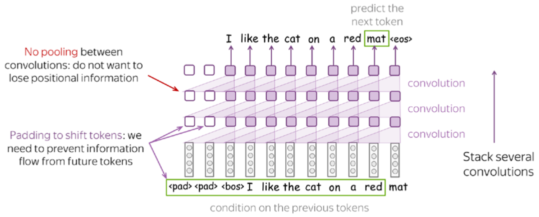
Image-Text Multimodal PTM


(1) Contrastive pre-training
Pepper the aussie pup
Text Encoder
Image Encoder
T_1 T_2 T_3 ... T_N
I_1 I_2 I_3 ... I_N
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
$I_2 \cdot T_1$ $I_2 \cdot T_2$ $I_2 \cdot T_3$ ... $I_2 \cdot T_N$
$I_3 \cdot T_1$ $I_3 \cdot T_2$ $I_3 \cdot T_3$ ... $I_3 \cdot T_N$
$I_N \cdot T_1$ $I_N \cdot T_2$ $I_N \cdot T_3$ ... $I_N \cdot T_N$
(2) Create dataset classifier from label text
plane
car
dog
bird
A photo of a {object}.
Text Encoder
(3) Use for zero-shot prediction
Image Encoder
I_1
A photo of a dog.

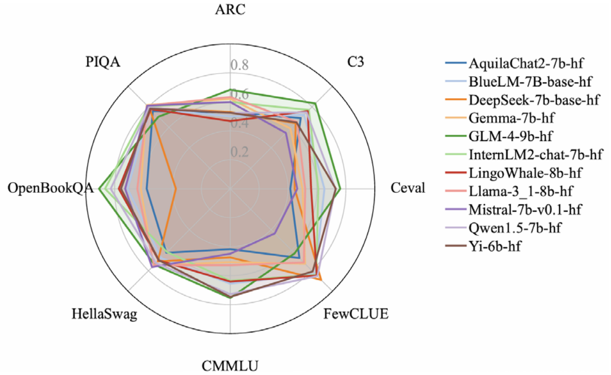
[Radford et al., ICML’21]



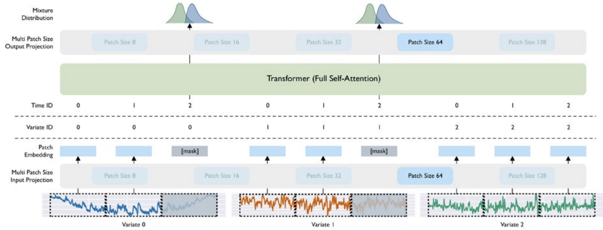
Language Generation PTM


predict the next token
I like the cat on a red mat <eos>
No pooling between convolutions: do not want to lose positional information
Padding to shift tokens: we need to prevent information flow from future tokens
convolution
convolution
convolution
Stack several convolutions
<pad> <pad> <bos> I like the cat on a red mat
condition on the previous tokens

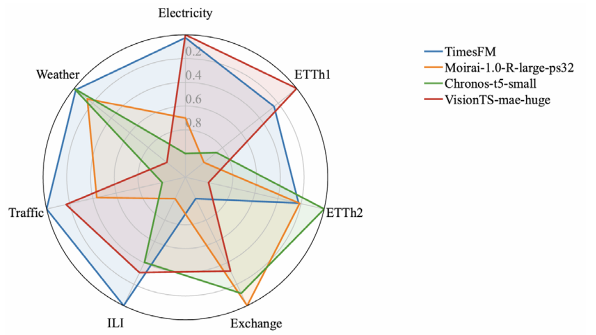
[Brown et al., NeurIPS’20]



Time Series Forecasting PTM



[Ansari et al., TMLR’24]

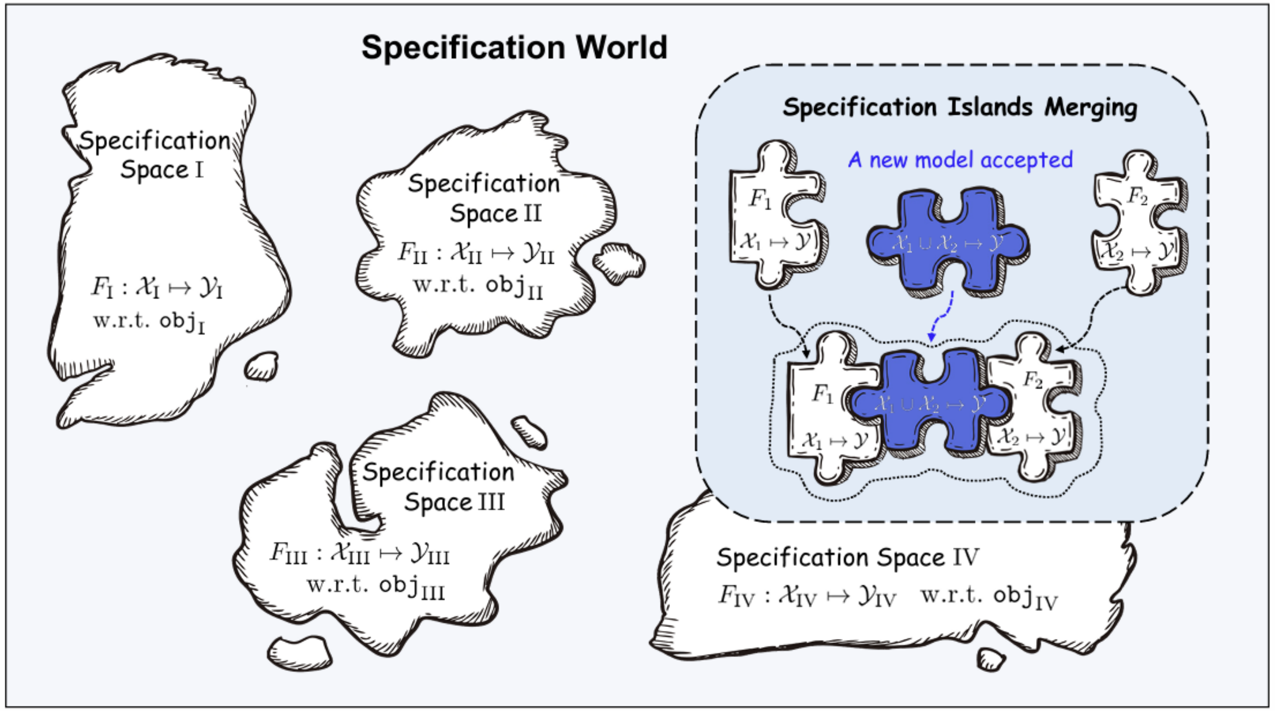


Whether reusing model zoo treats the diversity of PTM as a prerequisite?

Manage A Huge Model Zoo

The challenge: select the right PTM given a huge model zoo (like HuggingFace).

- How to manage the PTMs in the model zoo to facilitate the selection and adaptation.
- How to link different components of PTMs in a model zoo?


Specification World
Specification Space I
$F_{\mathrm{I}} : \mathcal{X}_{\mathrm{I}} \mapsto \mathcal{Y}_{\mathrm{I}}$ w.r.t. $\mathrm{obj}_{\mathrm{I}}$
Specification Space II
$F_{\mathrm{II}} : \mathcal{X}_{\mathrm{II}} \mapsto \mathcal{Y}_{\mathrm{II}}$ w.r.t. $\mathrm{obj}_{\mathrm{II}}$
Specification Space III
$F_{\mathrm{III}} : \mathcal{X}_{\mathrm{III}} \mapsto \mathcal{Y}_{\mathrm{III}}$ w.r.t. $\mathrm{obj}_{\mathrm{III}}$
Specification Islands Merging
A new model accepted
F_1
$\mathcal{X}_1 \mapsto \mathcal{Y}$
$\mathcal{X}_1 \cup \mathcal{X}_2 \mapsto \mathcal{Y}$
F_2
$\mathcal{X}_2 \mapsto \mathcal{Y}$
Specification Space IV
$F_{\mathrm{IV}} : \mathcal{X}_{\mathrm{IV}} \mapsto \mathcal{Y}_{\mathrm{IV}}$ w.r.t. $\mathrm{obj}_{\mathrm{IV}}$

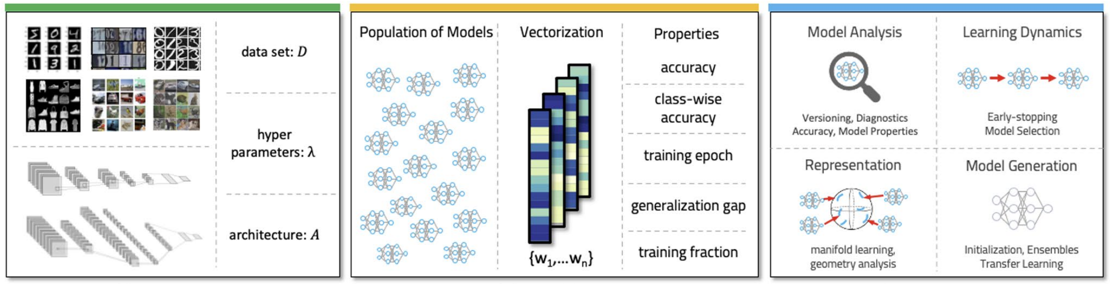
- Tree of PTMs
- Graph of PTMs
- Island of PTMs
- …

Ameya Prabhu et al., *Efficient Lifelong Model Evaluation in an Era of Rapid Progress*. NeurIPS 2024.

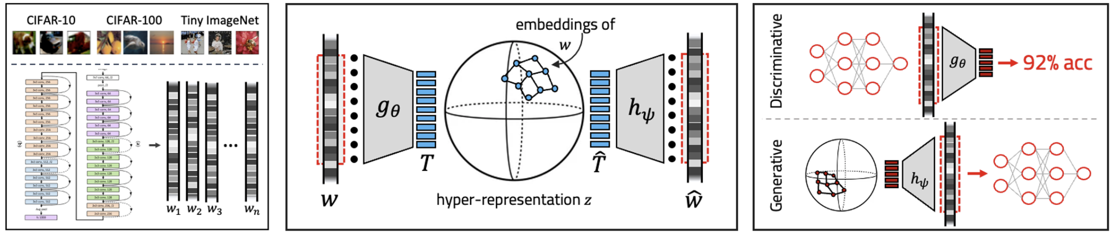
Model Representation Learning

Model Representation Learning vectorizes PTMs to encode their characteristics, forming hyper-representations that allow model selection without direct access to the original model parameters.

These representations can be learned based on intrinsic permutation-invariant properties of a PTM and can even be decoded back into model weights [Schürholt et al., NeurIPS’22][Schürholt et al., ICML’24].

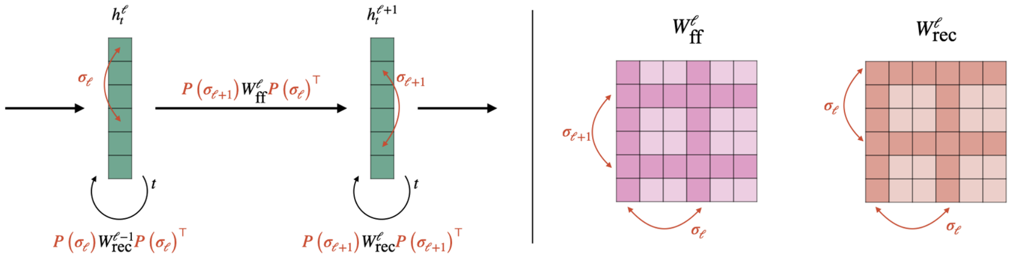


(I) Generation (II) Model Zoo (III) Use Cases



Model Zoos SANE: Sequential Autoencoder for Neural Embeddings Downstream Tasks

Learned representations serve multiple purposes: they act as model specifications within a model zoo, assist in efficient retrieval, and serve as auxiliary modalities during adaptation [Zhou et al., CoRR’24].



h_t^ℓ
$h_t^{\ell+1}$
σ_ℓ
$\sigma_{\ell+1}$
$P(\sigma_{\ell+1}) W_{\text{ff}}^\ell P(\sigma_\ell)^\top$
t
$P(\sigma_\ell) W_{\text{rec}}^{\ell-1} P(\sigma_\ell)^\top$
$P(\sigma_{\ell+1}) W_{\text{rec}}^\ell P(\sigma_{\ell+1})^\top$
W_{ff}^ℓ
W_{rec}^ℓ

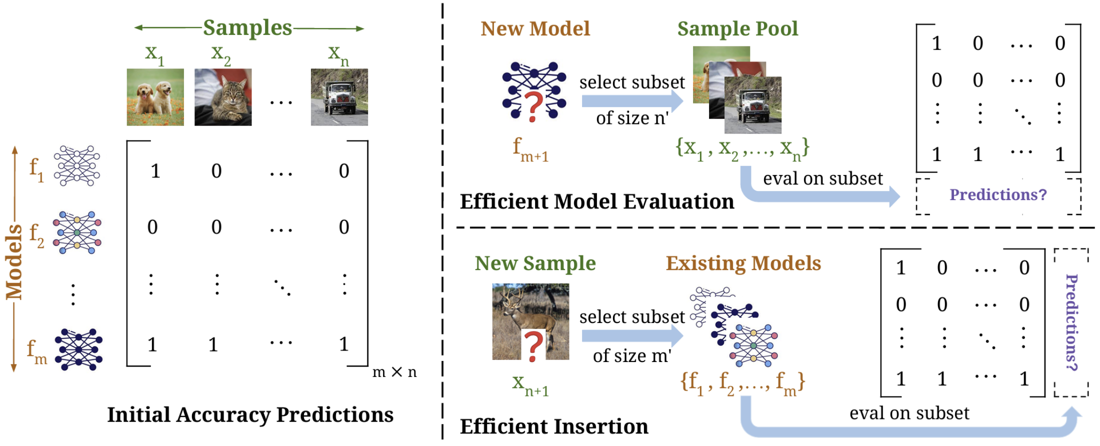
Automatically construct permutation equivariant models, which is referred to as universal neural functionals (UNFs), for any weight space.

Konstantin Schürholt, Michael W. Mahoney, Damian Borth. *Towards Scalable and Versatile Weight Space Learning*. ICML 2024.
Allan Zhou, Chelsea Finn, James Harrison. *Universal Neural Functionals*. CoRR 2024.

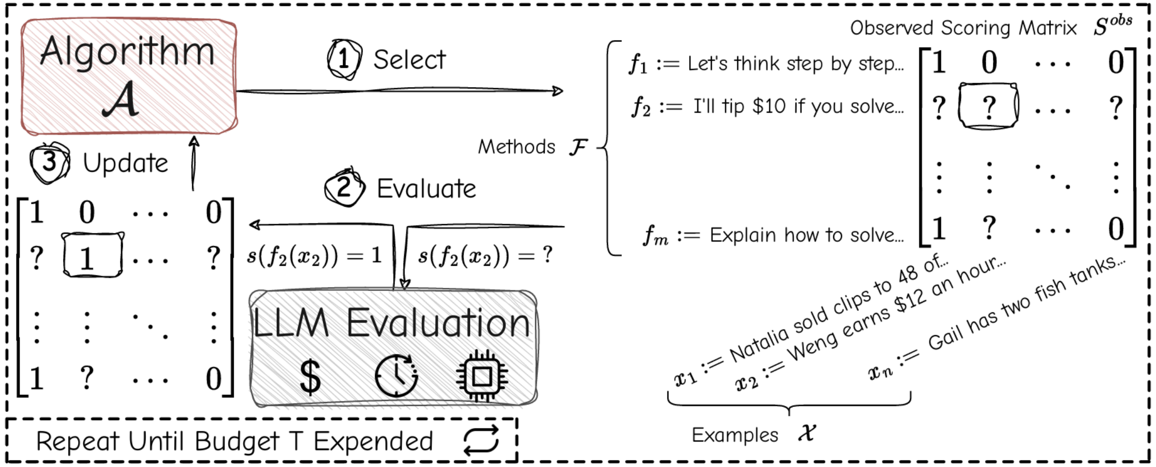
Manage A Huge Model Zoo

The *dual* problem: how to evaluate the ability of a PTM with limited data efficiently?

Sort & Search (S&S) reuses previously evaluated models by leveraging dynamic programming algorithms to selectively rank and sub-select test samples.



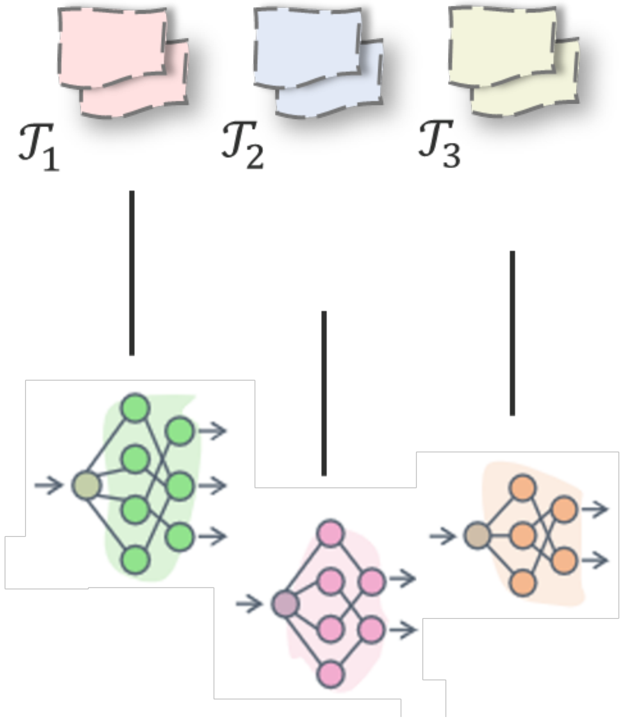
Use multi-armed bandits to sequentially identify the next (method, validation sample)-pair to evaluate and utilize low-rank matrix factorization to fill in missing evaluations.


Algorithm $\mathcal{A}$
① Select
② Evaluate
③ Update
$s(f_2(x_2)) = 1$
$s(f_2(x_2)) = ?$
LLM Evaluation
Repeat Until Budget T Expended
Observed Scoring Matrix S^{obs}
$f_1 :=$ Let's think step by step...
$f_2 :=$ I'll tip $10 if you solve...
$f_m :=$ Explain how to solve...
Methods $\mathcal{F}$
$x_1 :=$ Natalia sold clips to 48 of...
$x_2 :=$ Weng earns $12 an hour...
$x_n :=$ Gail has two fish tanks...
Examples $\mathcal{X}$

Ameya Prabhu, Vishaal Udandarao, Philip Torr, Matthias Bethge, Adel Bibi, Samuel Albanie. *Efficient Lifelong Model Evaluation in an Era of Rapid Progress*. NeurIPS 2024.
Jin Peng Zhou, Christian K Belardi, Ruihan Wu, Travis Zhang, Carla P Gomes, Wen Sun, Kilian Q Weinberger. *On Speeding Up Language Model Evaluation*. ICLR 2025.

Conclusion

Effectively taking advantage of PTM resources, can significantly enhance the sample and time efficiency of the machine learning paradigm. And the key lies in the efficient selection (and adaptation) of models.



- PTM selection requires capturing the characteristic of PTMs as well as tasks.
- Select the right PTM based on semantic/rules, forward-based metrics, or learned selectors/routers.
- The PTM selection notion could be generalized to image classification PTMs, LLMs, VLMs, time-series prediction models, and even LLM tools.

Thank you

Q&A

For more discussions, please contact yehj@nju.edu.cn