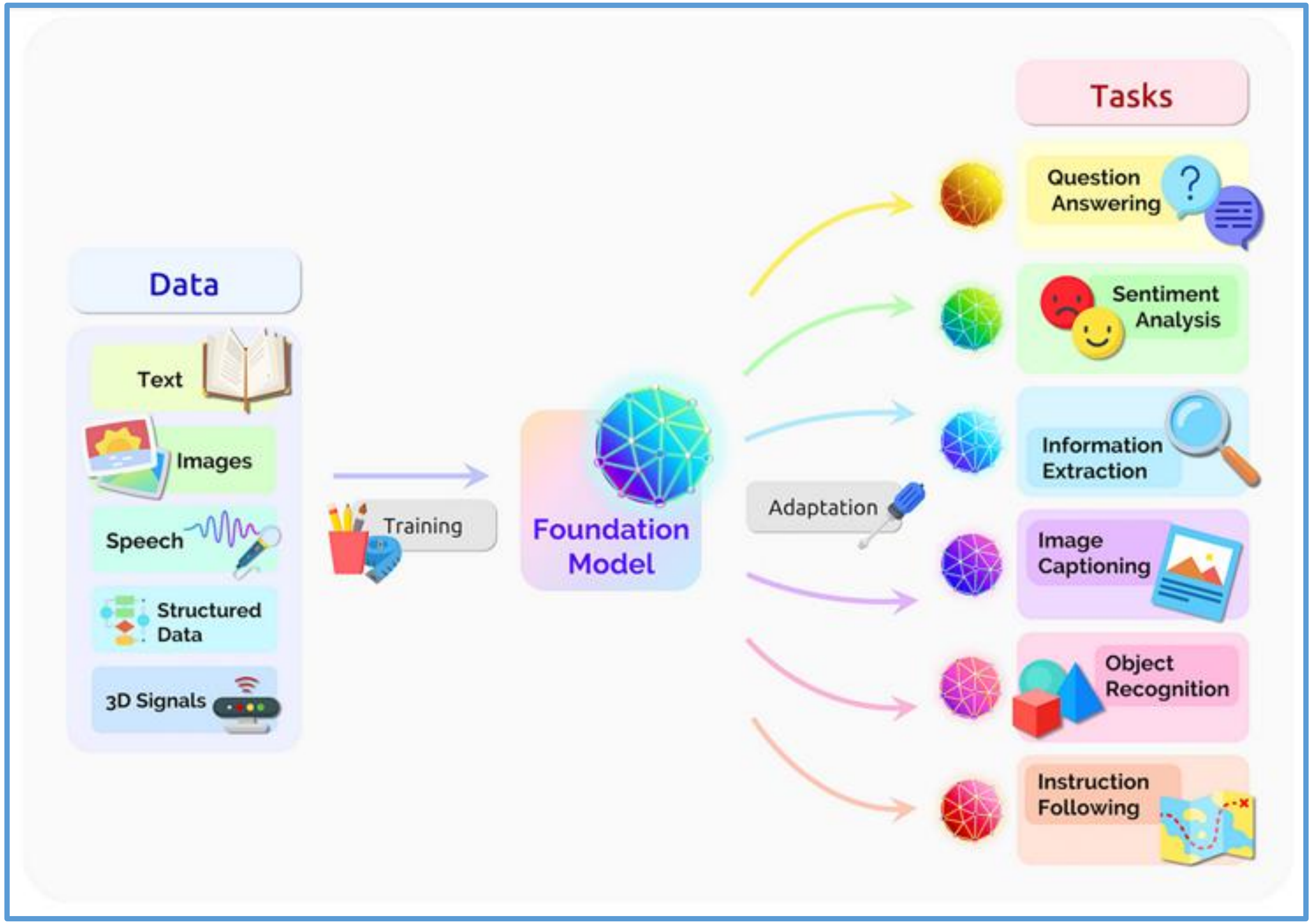
Towards Better Adaptation of Foundation Models

Zhuoyan Xu

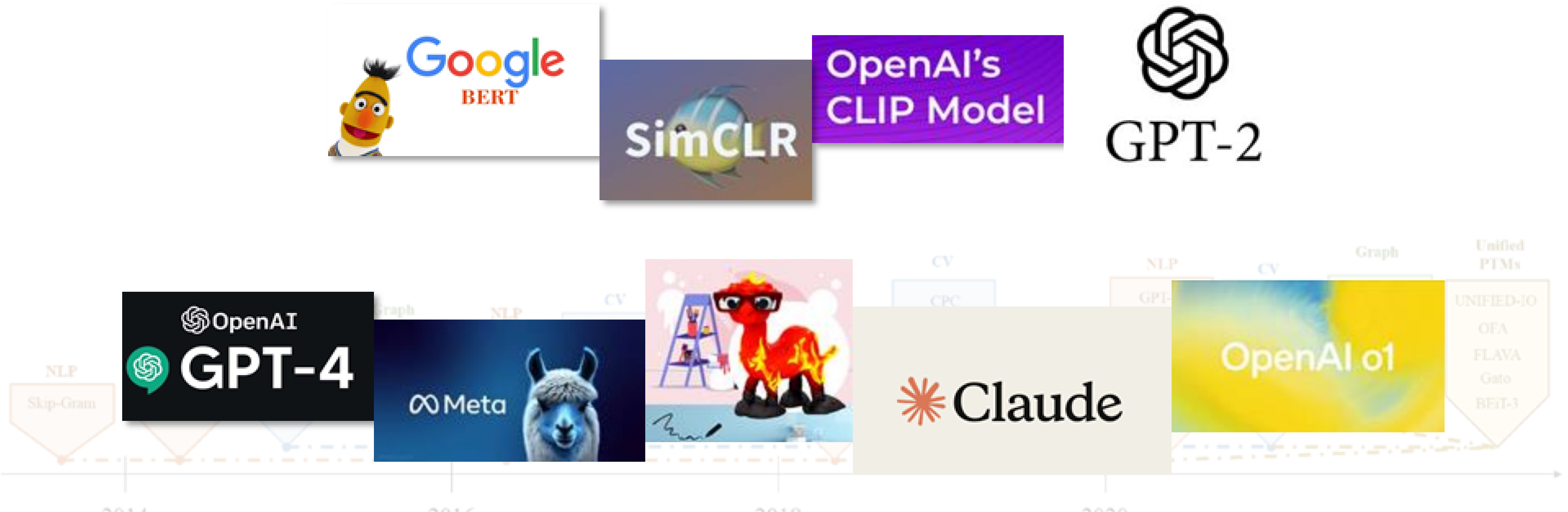
Committee: Yingyu Liang, Yin Li, Yiqiao Zhong, Junjie Hu

Foundation Models



Figures from: *On the opportunities and risks of foundation models, 2021.*

Evolution of Foundation Models



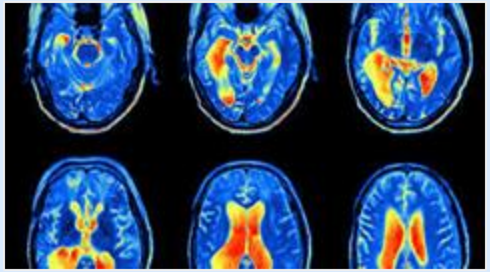
The history and evolution of foundation models

Figures from: *A Comprehensive Survey on Pretrained Foundation Models: A History from BERT to ChatGPT, 2023.*

Pretrained FMs are generalists:

There are gaps between these general models and specialized tasks.

Same task, different data



New tasks require reasoning

- 56 + 33 -> 89
- 67 -> sixty-seven
- …
- 31 + 25 -> fifty-six (?)

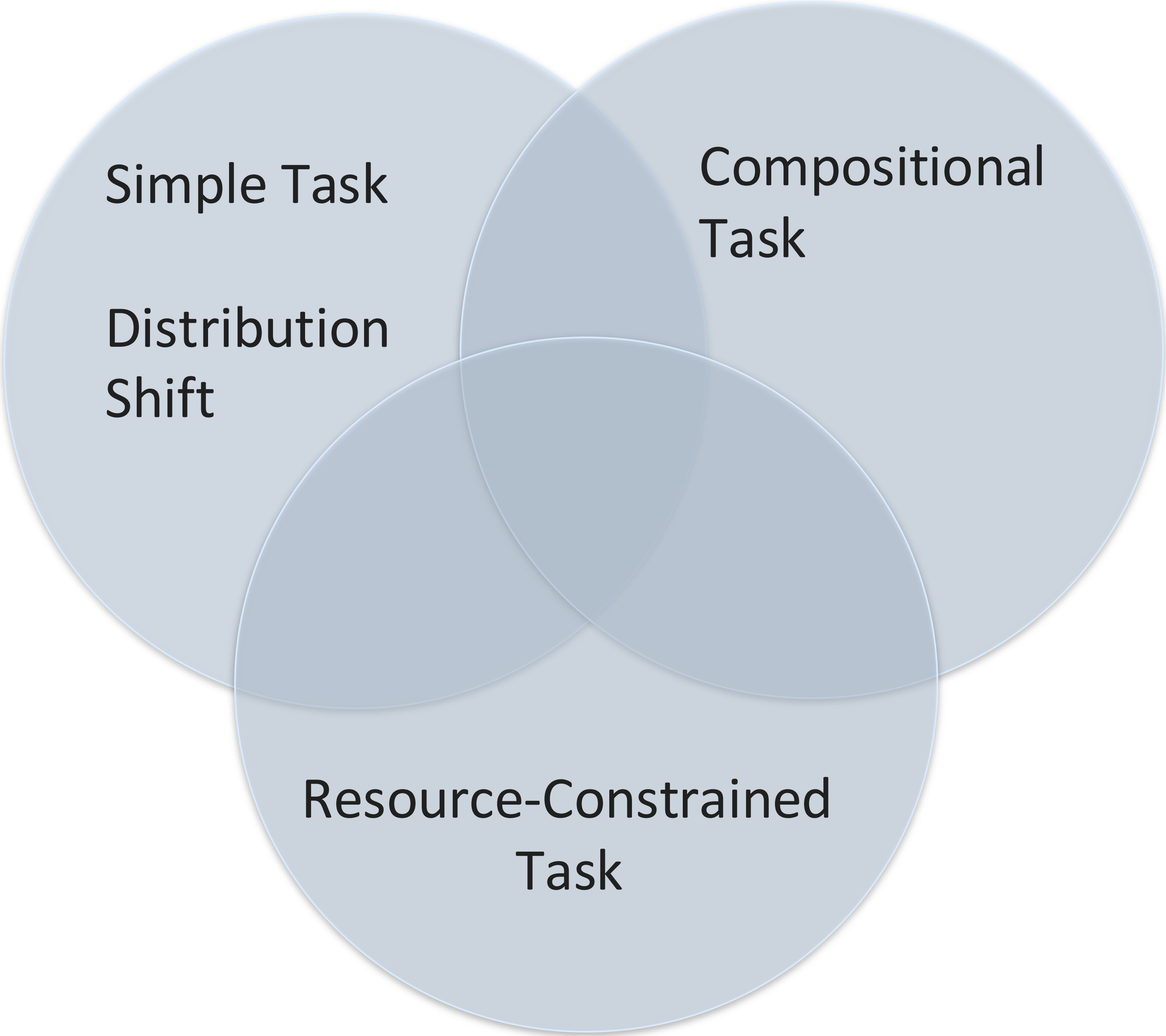
Tasks with resource constrain

FP32	6,480 TFLOPS	180 TFLOPS
FP64	3,240 TFLOPS	90 TFLOPS
FP64 Tensor Core	3,240 TFLOPS	90 TFLOPS
GPU Memory | Bandwidth	Up to 13.5 TB HBM3e | 576 TB/s	Up to 384 GB HBM3e | 16 TB/s
NVLink Bandwidth	130TB/s	3.6TB/s

GB200 NVL72[1] Specs

https://www.nvidia.com/en-us/data-center/gb200-nvl72/

Adaptation to new tasks:

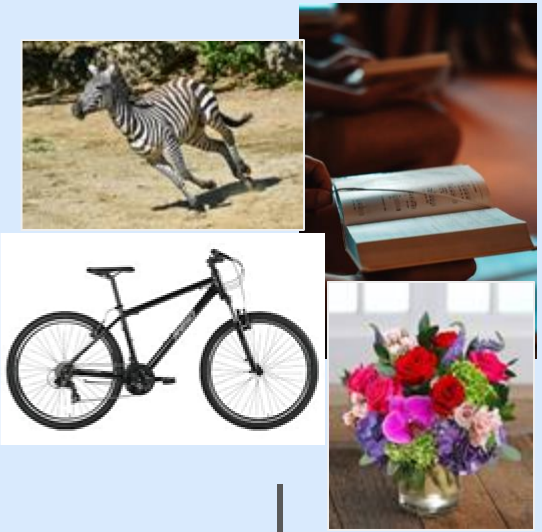


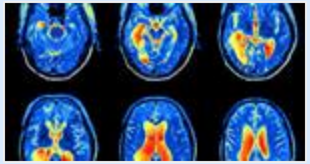
My Work

Foundation models (FMs) are trained as generalists, my research:

1. Enables FMs to better specialize in tasks in different domains
2. Advances FMs' ability to handle complex problems by combining simple tasks
3. Make FMs more deployable by reducing computational overhead

Data Distribution Shift





Compositional Ability

* bird

BIRD

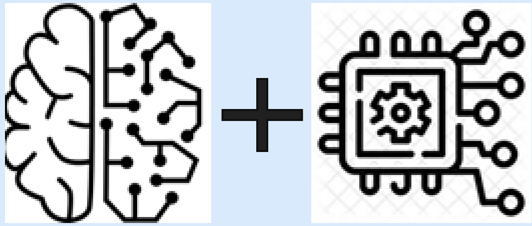
(house hat)

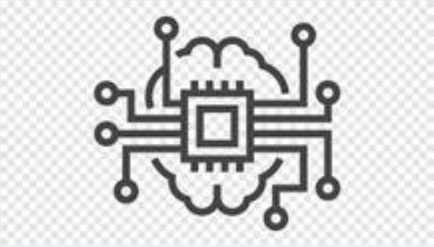
hat house

(* pie * sports)

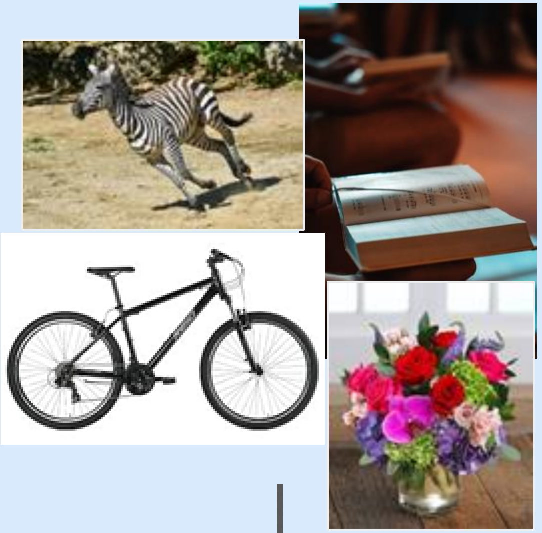
?

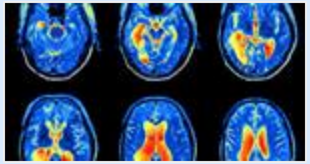
Efficient Inference





Data Distribution Shift





Compositional Ability

* bird

BIRD

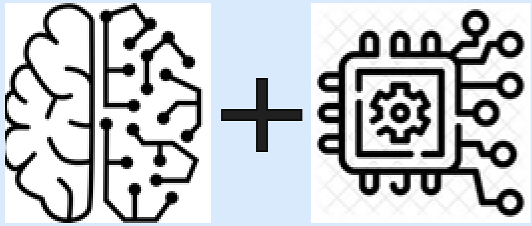
(house hat)

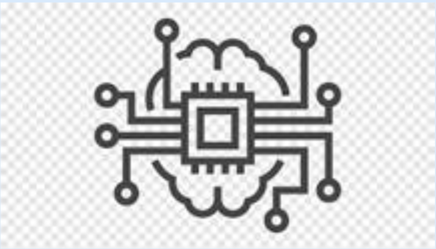
hat house

(* pie * sports)

?

Efficient Inference



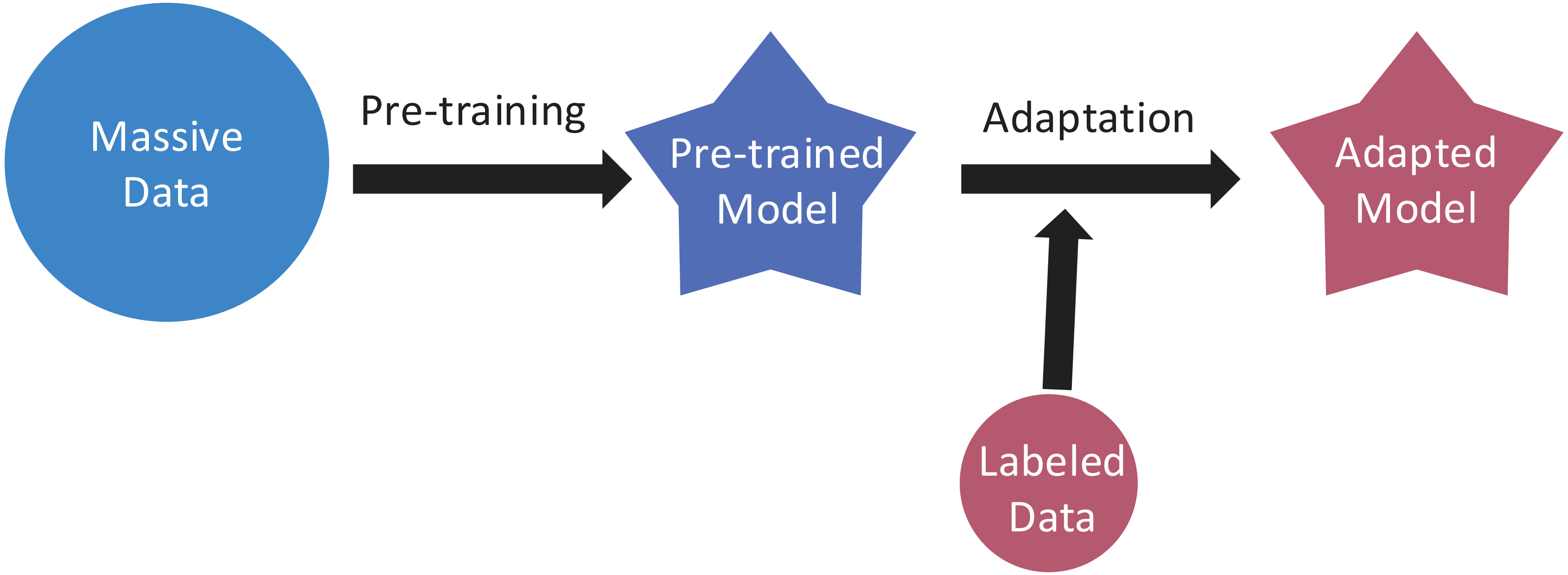


Few-Shot Adaptation

Multitask Finetuning

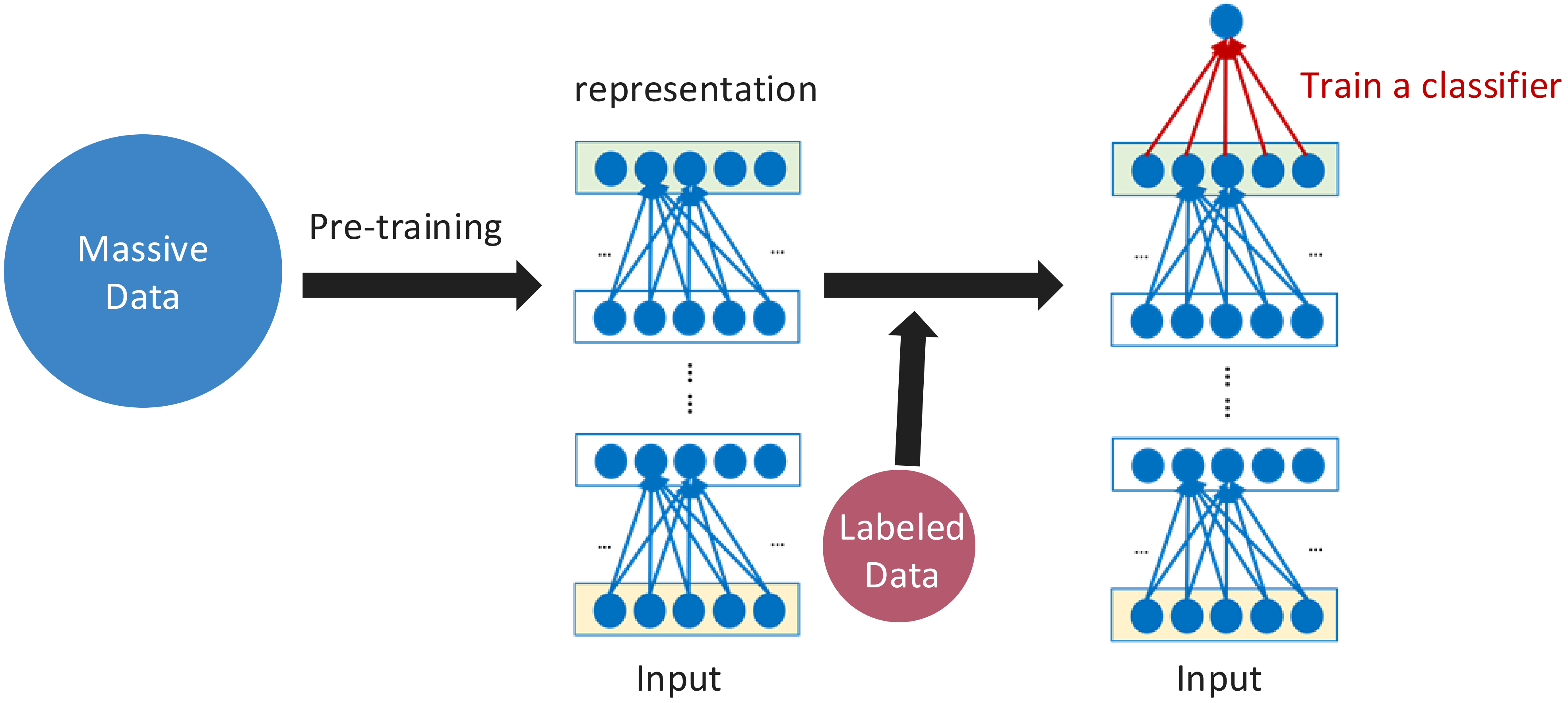
New Paradigm: Pre-trained Representations

Paradigm shift: supervised learning → pre-training + adaptation



New Paradigm: Pre-trained Representations

Paradigm shift: supervised learning → pre-training + adaptation



New Paradigm: Pre-trained Representations

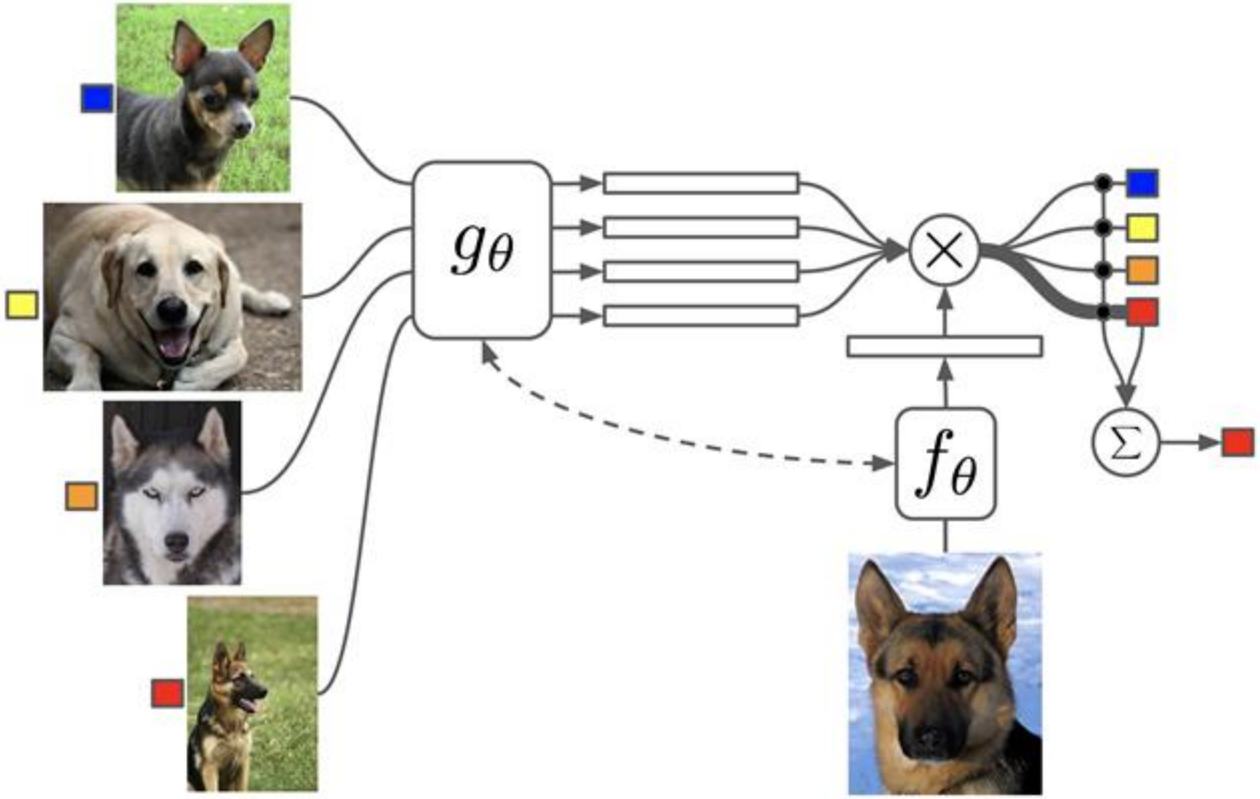


Figure 1: Matching Networks architecture

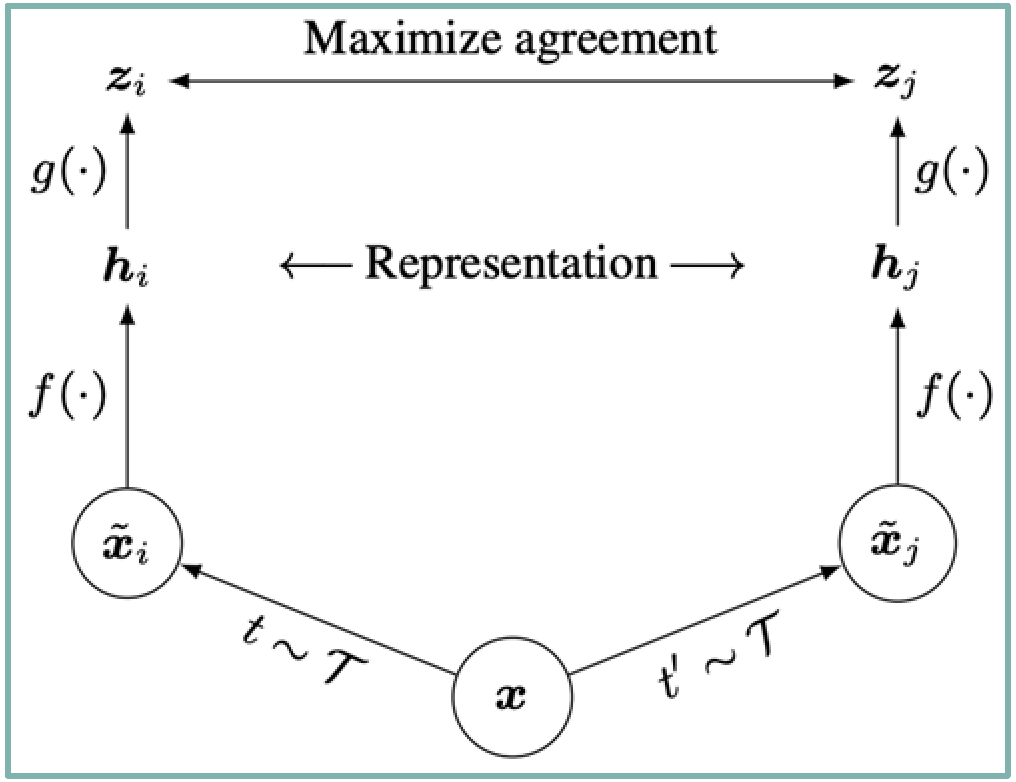
Adaptation of a pre-trained image encoder

Figures from: *Matching Networks for One Shot Learning, 2017.*

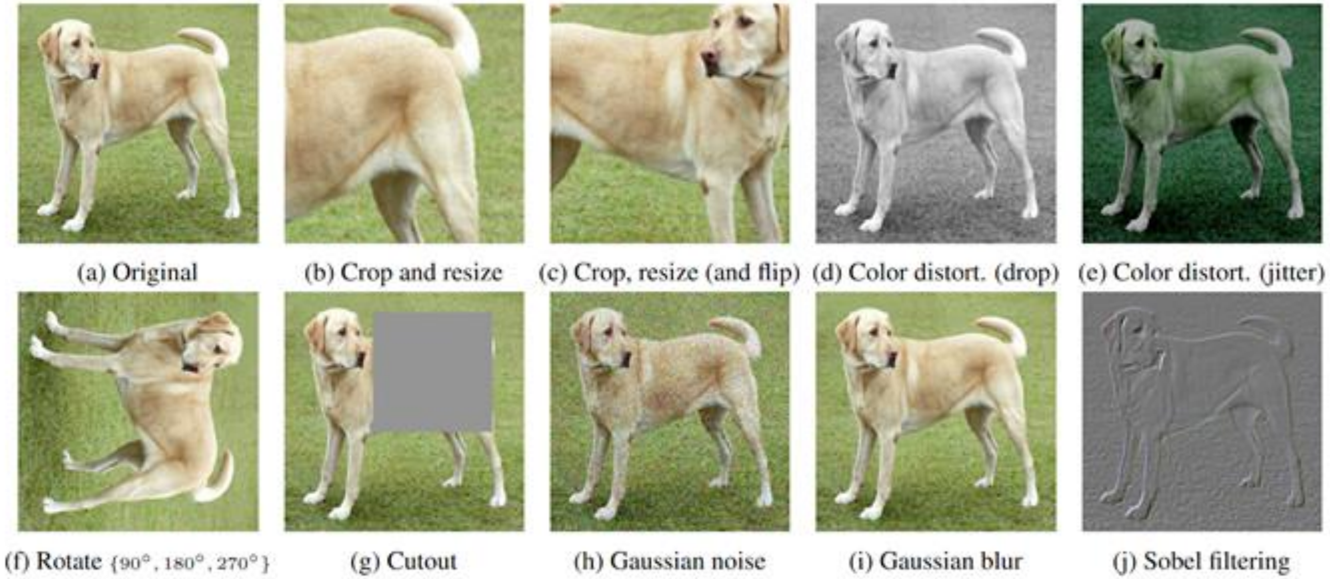
What does pre-training look like?

- Supervised learning
- Self-supervised learning:
  - Next sentence prediction (BERT)
  - Masked language prediction (BERT, RoBERTa)
  - Auto-regressive language modeling (GPT, Llama)
  - Contrastive learning (SimCLR, SimCSE, CLIP, DINO)

Contrastive Learning



$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(z_i, z_k)/\tau)}$$



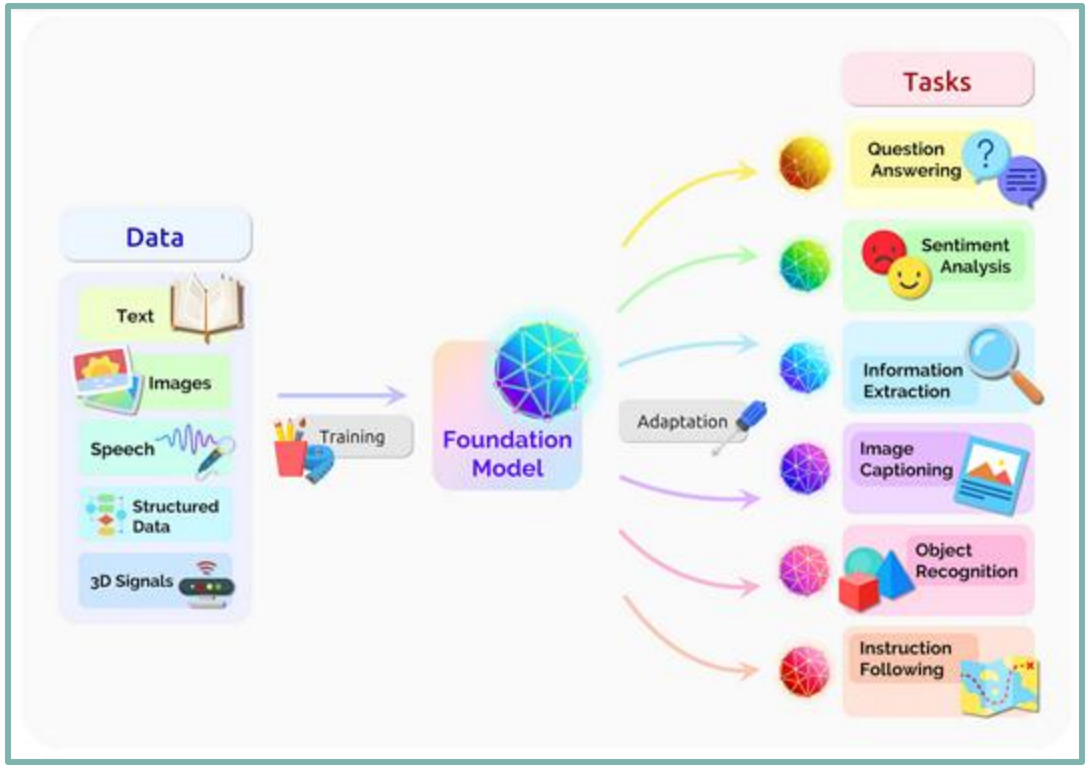
SimCLR - (Image, Image)

No need labels

Figures from: *A Simple Framework for Contrastive Learning of Visual Representations, 2020*

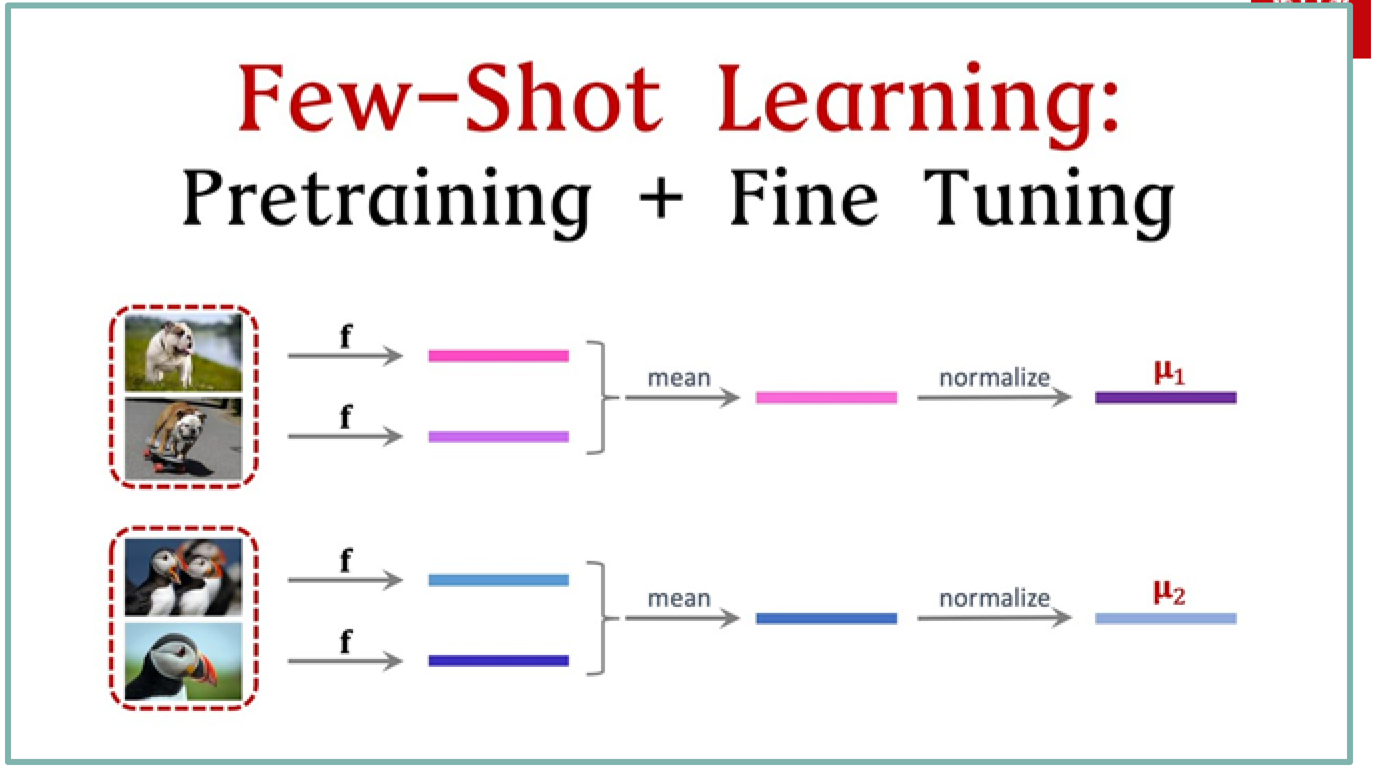
Image Data Augmentation

Figures from: *A Simple Framework for Contrastive Learning of Visual Representations, 2020*



Universality

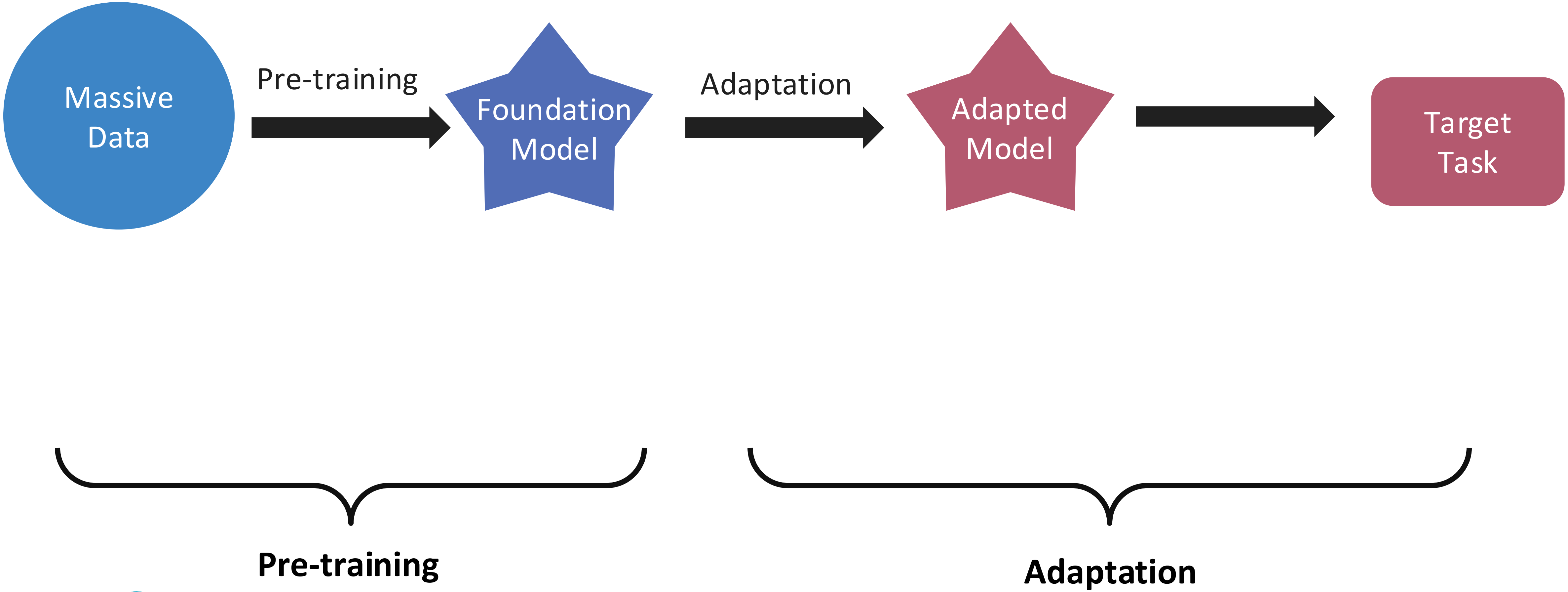
Figures from: *On the opportunities and risks of foundation models, 2021.*



Label Efficiency

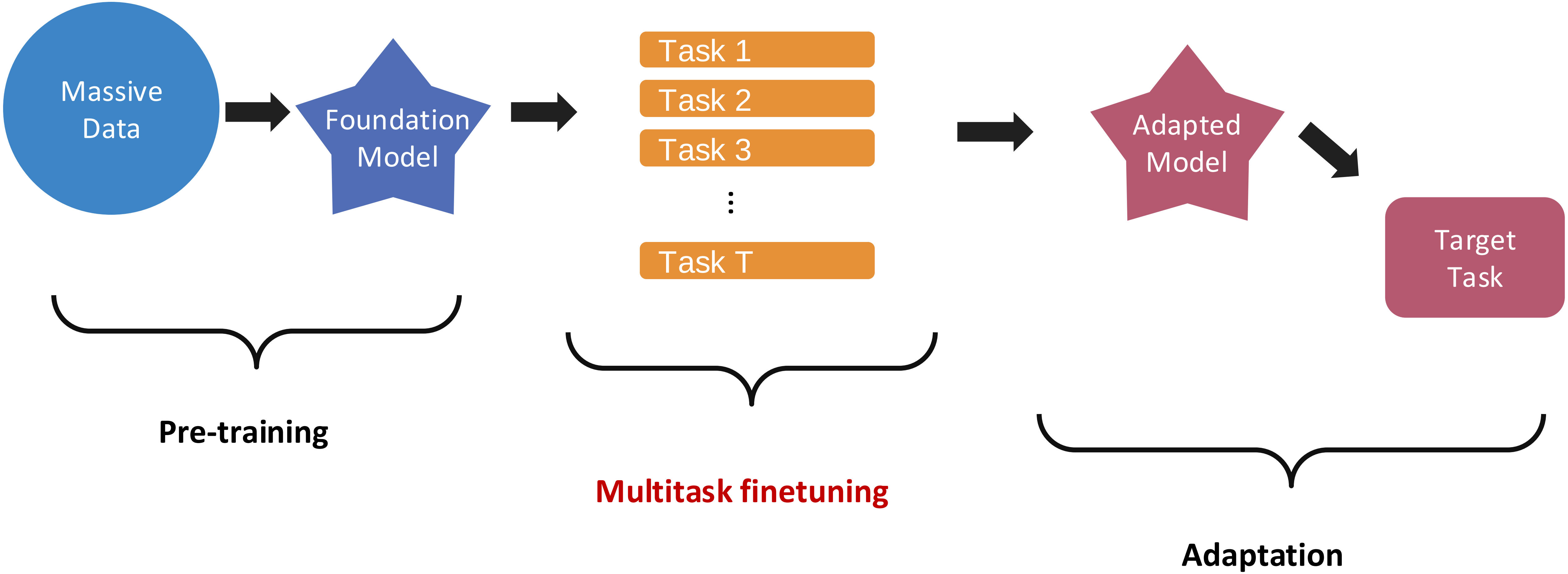
Figures from: *https://www.youtube.com/watch?v=U6uFOIURcD0&ab_channel=ShusenWang, 2020*

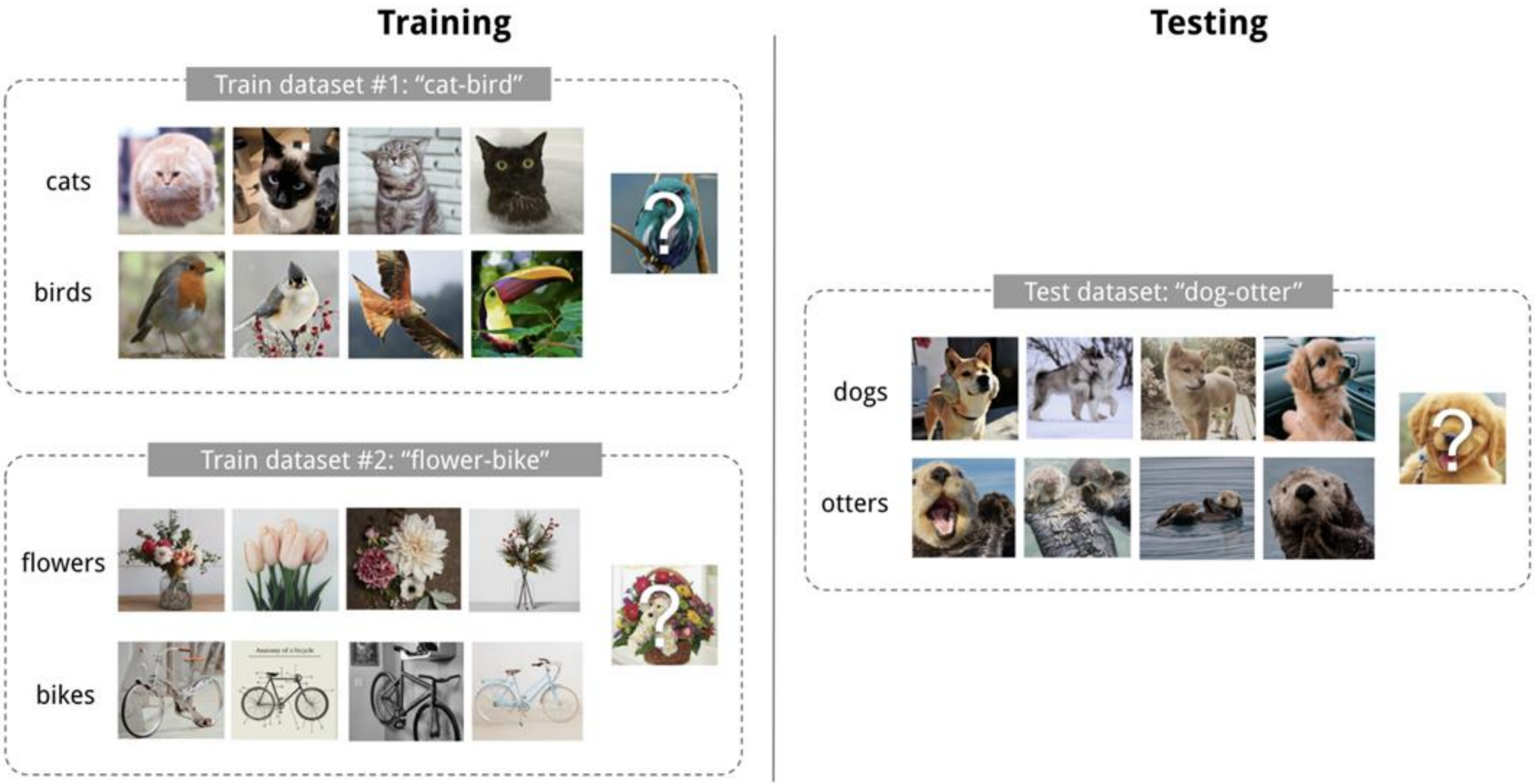
Paradigm: Pre-training + Adaptation



Q: Can we improve this?

Pre-training + Finetuning + Adaptation





An example of 4-shot 2-class image classification

Figures from: [Meta-Learning: Learning to Learn Fast](), *2018.*

Problem Setup - Hidden representation data model

- Class $y \in \mathcal{C}$ over distribution $y \sim \eta$
- Task $\mathcal{T} = (y_1, \ldots, y_K) \subseteq \mathcal{C}$, sample $x \sim \mathcal{D}(y)$
- $\phi \in \Phi$ hypothesis class of representation functions, e.g. ResNet, ViT
- $g(x) = W\phi(x)$ as prediction logits of latent class

$$y \longrightarrow x \longrightarrow \phi(x) \longrightarrow g(x) \longrightarrow \text{loss}$$

Dog

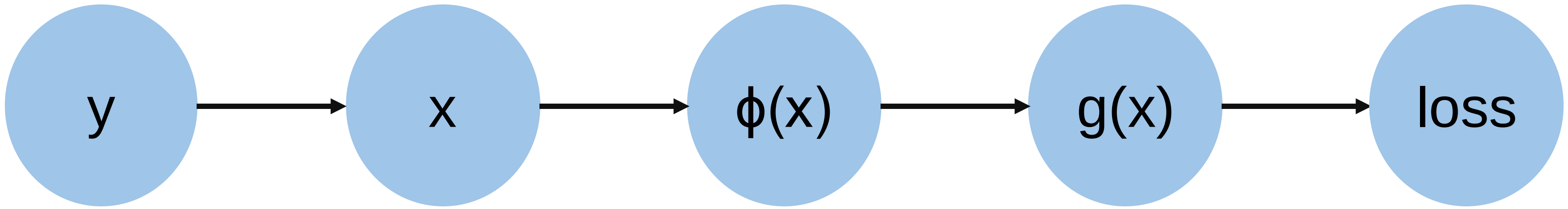
$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_d \end{bmatrix} \qquad \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_K \end{bmatrix}$$

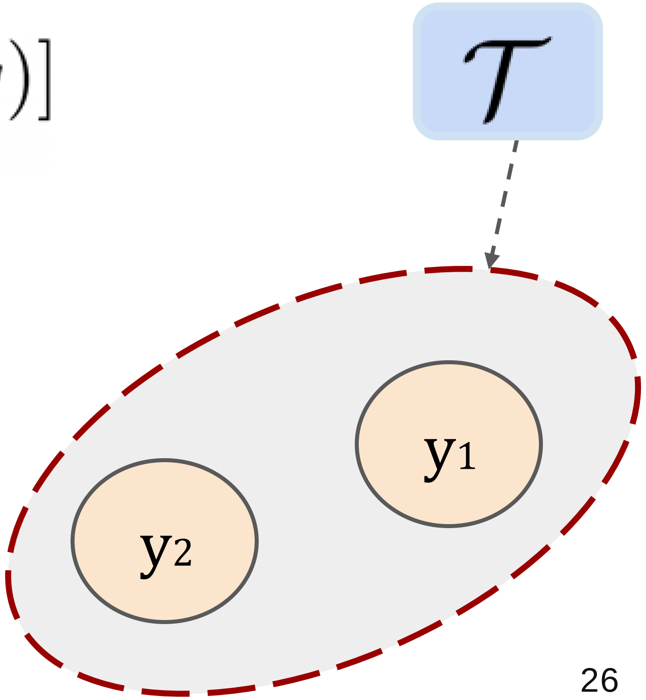
$$\ell(g(x), y) = -\log\left\{\frac{\exp\left(g(\mathbf{x})_y\right)}{\sum_{k=1}^{K} \exp\left(g(\mathbf{x})_k\right)}\right\}$$

Problem Setup - Objective for a downstream task

- Class $y \in \mathcal{C}$ over distribution $y \sim \eta$
- Task $\mathcal{T} = \{y_1, y_2\} \subseteq \mathcal{C}$ $x \sim \mathcal{D}(y)$, instance
- $g(x) = W\phi(x)$ as prediction logits of latent class
- Supervised loss w.r.t a task:

$$\mathcal{L}_{\text{sup}}(\mathcal{T}, \phi) := \min_{W} \mathop{\mathbb{E}}_{y \sim \mathcal{T}} \mathop{\mathbb{E}}_{x \sim \mathcal{D}(y)} [\ell(W\phi(x), y)]$$





Pretraining - Contrastive learning

- $(y, y^-) \sim \eta^2 \quad x, x^+ \sim \mathcal{D}(y) \;\; x^- \sim \mathcal{D}(y^-) \;\; \tau := \Pr_{(y,y^-)\sim\eta^2}\{y = y^-\}$
- Contrastive loss: $$\mathbb{E}\left[-\log\left(\frac{e^{\phi(x)^\top \phi(x^+)}}{e^{\phi(x)^\top \phi(x^+)} + e^{\phi(x)^\top \phi(x^-)}}\right)\right]$$

Data Model

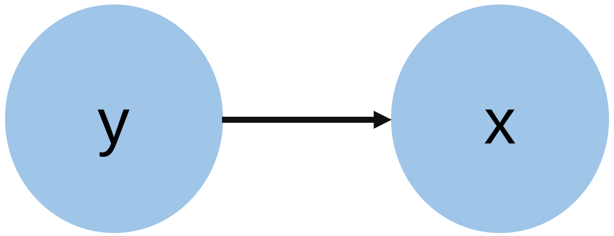
Figures from: *Expanding Small-Scale Datasets with Guided Imagination, 2023*

Pretraining - Supervised learning

- $y \sim \eta \quad x \sim \mathcal{D}(y)$
- supervised loss: $\ell(g(x), y) = \ell_u\left((g(x))_y - (g(x))_{y' \neq y, y' \in \mathcal{C}}\right)$

$$\mathcal{L}_{sup-pre}(\phi) = \min_W \mathbb{E}_{x,y}[\ell(W\phi(x), y)]$$

- In particular: $\ell_u(v) = \log(1 + \exp(-v))$ will recover the logistic loss

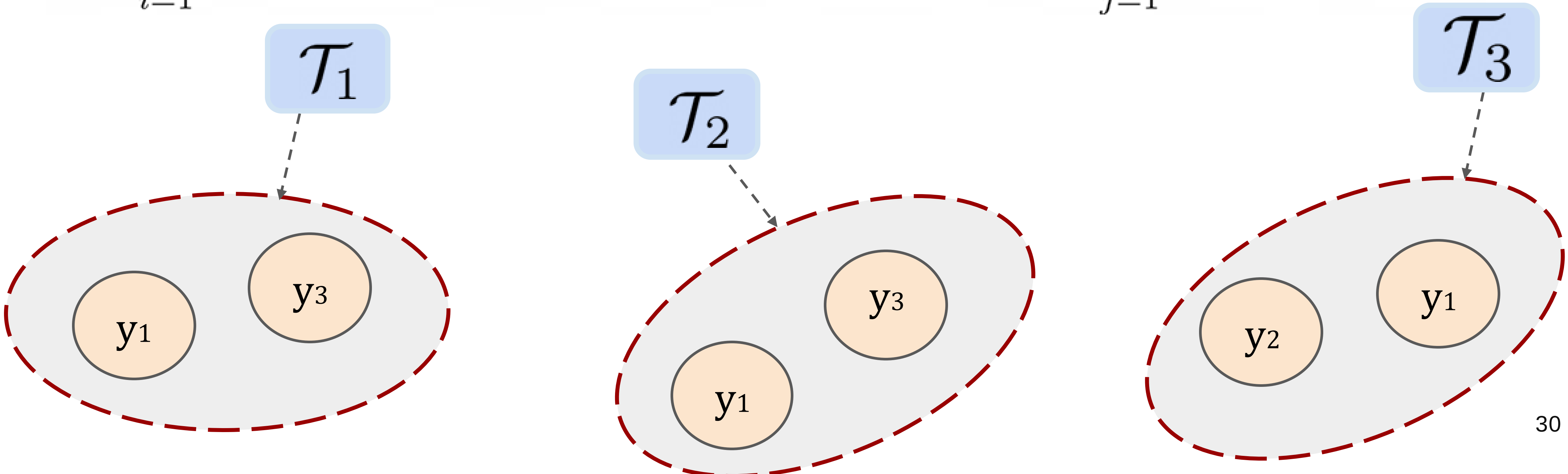


To simplify notation, we will use $\mathcal{L}_{pre}(\phi)$, we denote pretrained model as $\hat{\phi}$

Problem Setup - Multitask Finetuning

- Suppose we construct M tasks $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_M\}$
- Suppose each task with m sample $\mathcal{S}_i := \{(x_j^i, y_j^i) : j \in [m]\}$
- Given pretrained $\hat{\phi}$. We further multitask finetune it by objective:

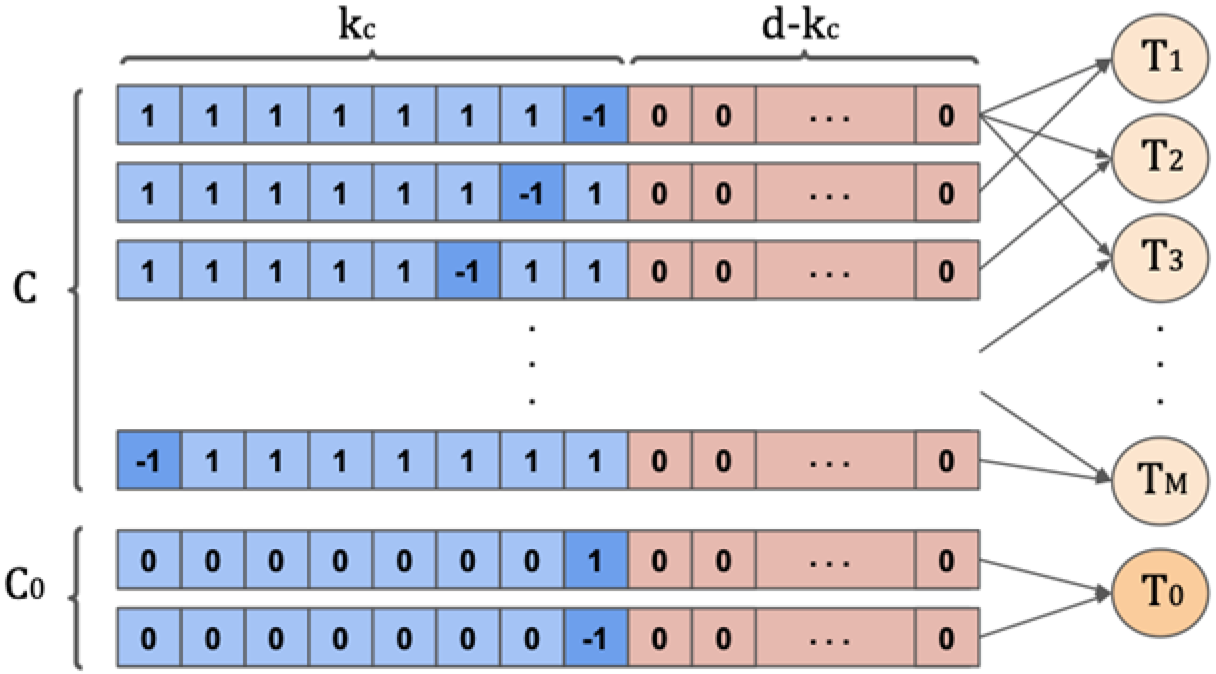
$$\min_{\phi \in \Phi} \frac{1}{M} \sum_{i=1}^{M} \widehat{\mathcal{L}}_{\text{sup}}\left(\mathcal{T}_i, \phi\right), \quad \text{where } \widehat{\mathcal{L}}_{\text{sup}}\left(\mathcal{T}_i, \phi\right) := \min_{W_i \in \mathbb{R}^d} \frac{1}{m} \sum_{j=1}^{m} \ell\left(W_i^{\top} \phi\left(x_j^i\right), y_j^i\right)$$



Diversity and Consistency

Definition 1 (Diversity and Consistency (Informal))
Consider the latent feature space of target task data and finetuning task data. **Diversity** refer to the coverage of the finetuning tasks on the target task in the latent feature space. **Consistency** refer to similarity in the feature space.

- Suppose target task is $\mathcal{T}_0$

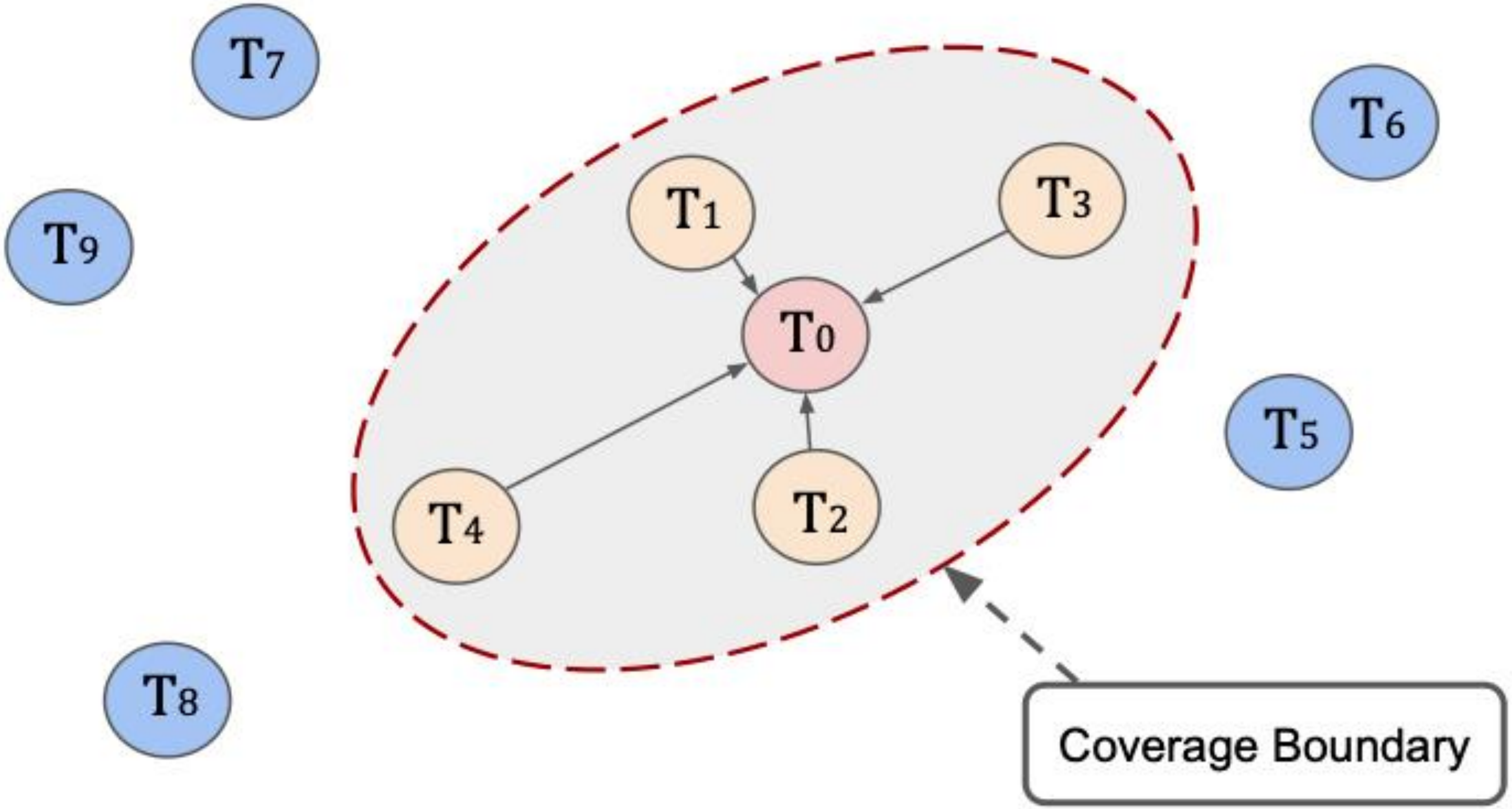


Main Result

- Suppose target task is $\mathcal{T}_0$
- Let $\phi^* \in \Phi$ denote the model with the lowest target task loss $\mathcal{L}_{sup}(\mathcal{T}_0, \phi^*)$
- We want to bound $\mathcal{E}(\phi) = \mathcal{L}_{\text{sup}}(\mathcal{T}_0, \phi) - \mathcal{L}_{\text{sup}}(\mathcal{T}_0, \phi^*)$
- Pretraining loss as $\hat{\mathcal{L}}_{\text{pre}}(\hat{\phi})$

Theorem (Multitask finetuning loss (Informal))
Suppose in pretraining we have empirical pretraining loss $\hat{\mathcal{L}}_{\text{pre}}(\hat{\phi}) \leq \epsilon_0$
The error will be $\mathcal{E}(\hat{\phi}) \leq \mathcal{O}(\epsilon_0)$. After sufficient multitask finetuning and get ϕ', the error will be $\mathcal{E}(\phi') \leq \mathcal{O}(\alpha\epsilon_0)$ with high probability. The finetuning sample complexity will be $\Omega\left(\frac{1}{\alpha\epsilon_0}\right)$.

Practical solution: Task selection



Practical solution: Task selection

Algorithm 1 Consistency-Diversity Task Selection

Input: Target task $\mathcal{T}_0$, candidate finetuning tasks: $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_M\}$, model ϕ, threshold p.

1: Compute $\phi(\mathcal{T}_i)$ and $\mu_{\mathcal{T}_i}$ for $i = 0, 1, \ldots, M$.

2: Sort $\mathcal{T}_i$'s in descending order of similarity $(\mathcal{T}_0, \mathcal{T}_i)$. Denote the sorted list as $\{\mathcal{T}_1', \mathcal{T}_2', \ldots, \mathcal{T}_M'\}$.

3: $L \leftarrow \{\mathcal{T}_1'\}$

4: **for** $i = 2, \ldots, M$ **do**

5: If coverage$(L \cup \mathcal{T}_i'; \mathcal{T}_0) \geq (1 + p) \cdot$ coverage$(L; \mathcal{T}_0)$, then $L \leftarrow L \cup \mathcal{T}_i'$; otherwise, break.

6: **end for**

Output: selected data L for multitask finetuning.

Experiments: Few-shot Vision tasks

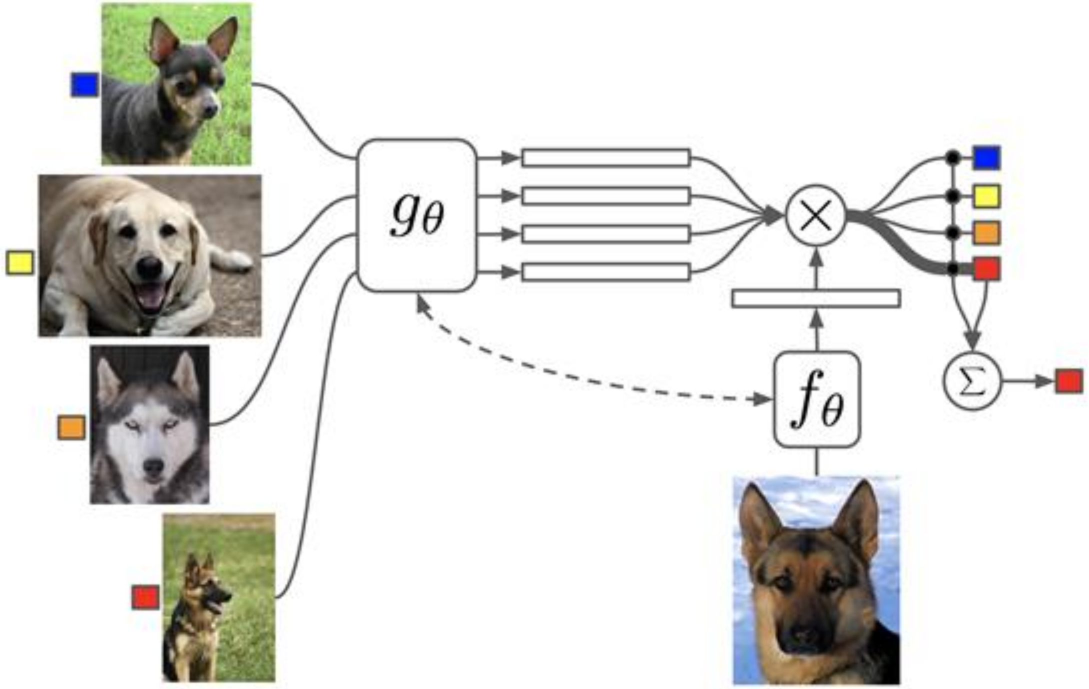
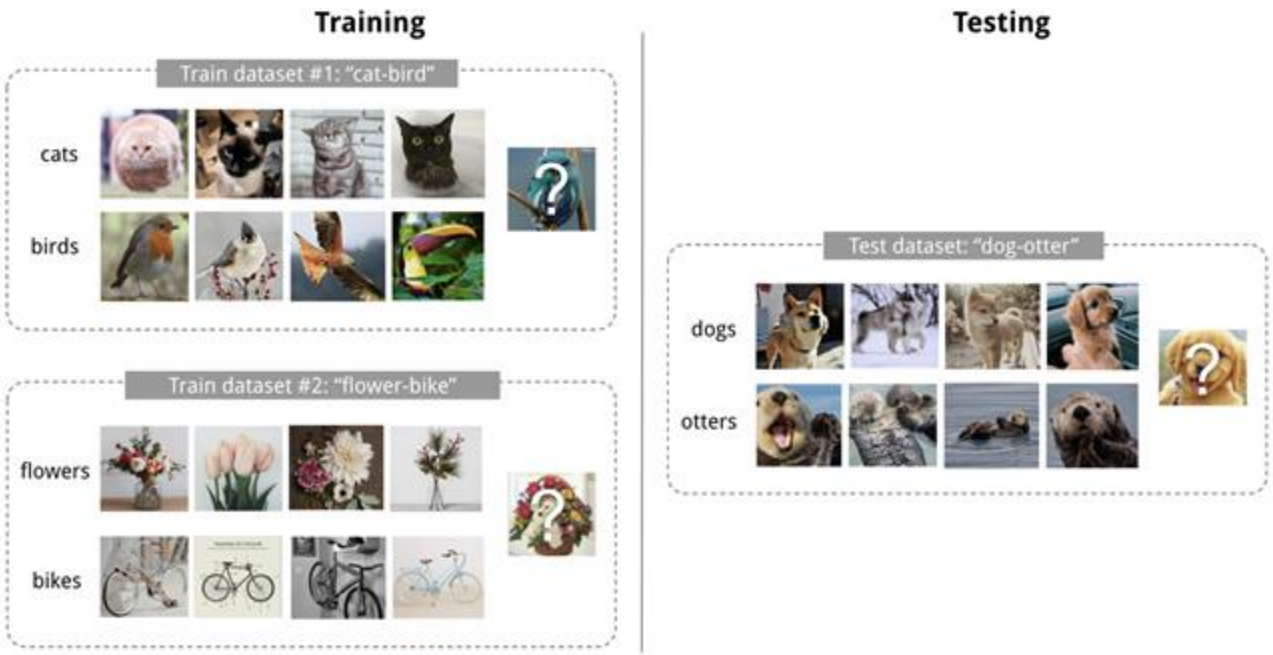
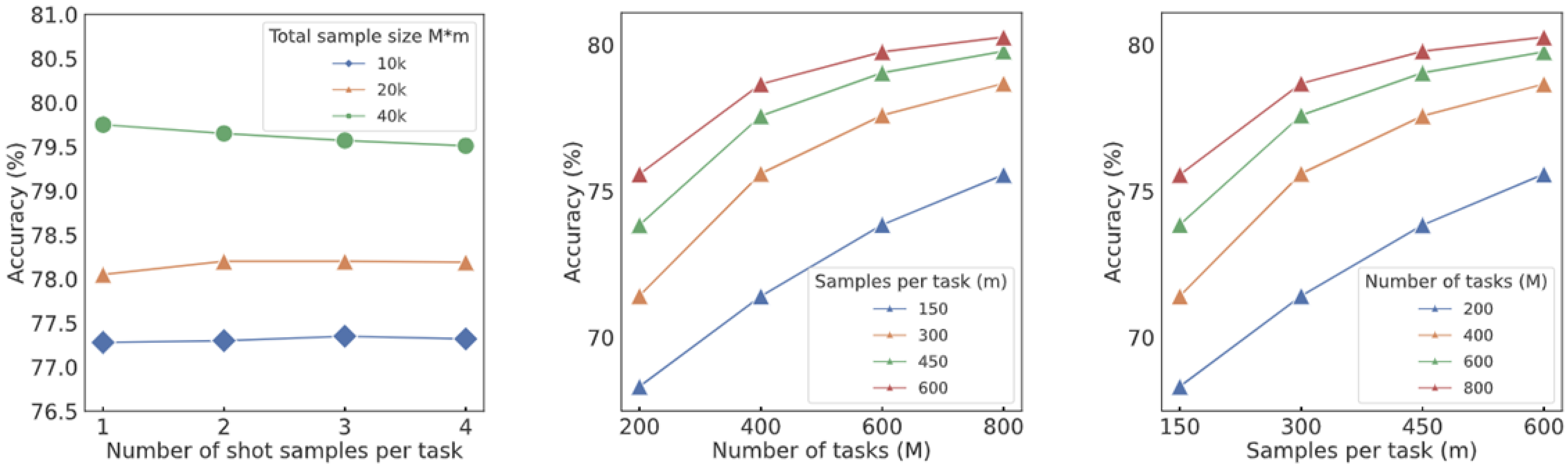


Figure 1: Matching Networks architecture



Experiments: Verification of Theoretical Analysis



(a) # shots during finetuning. (b) # tasks during finetuning. (c) # samples during finetuning.

Figure 3: Results on ViT-B backbone pretrained by MoCo v3. (a) Accuracy v.s. number of shots per finetuning task. Different curves correspond to different total numbers of samples Mm. (b) Accuracy v.s. the number of tasks M. Different curves correspond to different numbers of samples per task m. (c) Accuracy v.s. number of samples per task m. Different curves correspond to different numbers of tasks M.

Experiments: Task selection algorithm

Pretrained	Selection	INet	Omglot	Acraft	CUB	QDraw	Fungi	Flower	Sign	COCO
CLIP	Random	56.29	65.45	31.31	59.22	36.74	31.03	75.17	33.21	30.16
	No Con.	60.89	72.18	31.50	66.73	40.68	35.17	81.03	37.67	34.28
	No Div.	56.85	73.02	32.53	65.33	40.99	33.10	80.54	34.76	31.24
	Selected	**60.89**	**74.33**	**33.12**	**69.07**	**41.44**	**36.71**	**80.28**	**38.08**	**34.52**
DINOv2	Random	83.05	62.05	36.75	93.75	39.40	52.68	98.57	31.54	47.35
	No Con.	83.21	76.05	36.32	93.96	50.76	53.01	98.58	34.22	47.11
	No Div.	82.82	79.23	36.33	93.96	55.18	52.98	98.59	35.67	44.89
	Selected	**83.21**	**81.74**	**37.01**	**94.10**	**55.39**	**53.37**	**98.65**	**36.46**	**48.08**
MoCo v3	Random	59.66	60.72	18.57	39.80	40.39	32.79	58.42	33.38	32.98
	No Con.	59.80	60.79	18.75	40.41	40.98	32.80	59.55	34.01	33.41
	No Div.	59.57	63.00	18.65	40.36	41.04	32.80	58.67	34.03	33.67
	Selected	**59.80**	**63.17**	**18.80**	**40.74**	**41.49**	**33.02**	**59.64**	**34.31**	**33.86**

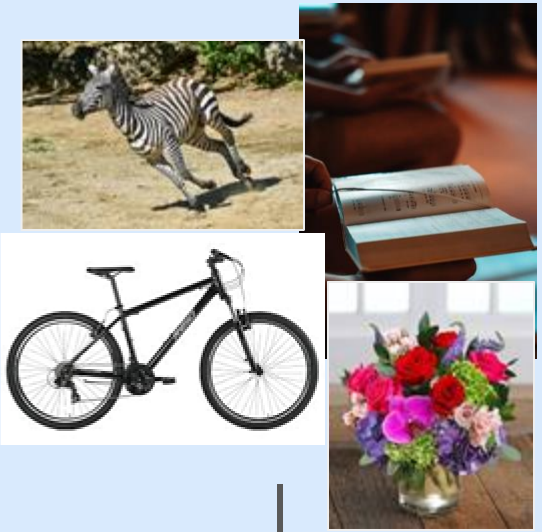
Table 1: Results evaluating our task selection algorithm on Meta-dataset using ViT-B backbone. No Con.: Ignore consistency. No Div.: Ignore diversity. Random: Ignore both consistency and diversity.

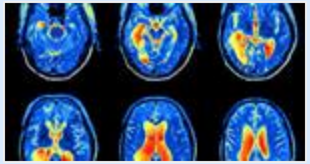
Multitask Finetuning for Adaptation

Developed a targeted adaptation framework that:

1. Identifies and selects relevant data matching target task characteristics
2. Designs specialized multitask finetuning pipeline
3. Achieves strong performance with limited target data

Data Distribution Shift





Compositional Ability

* bird

BIRD

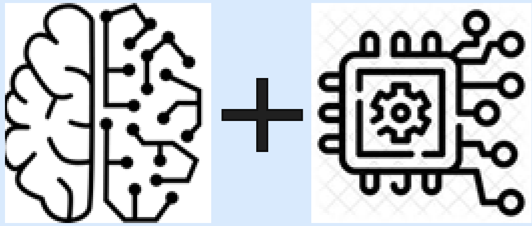
(house hat)

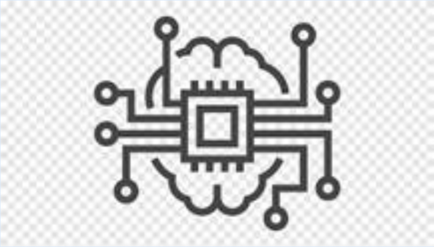
hat house

(* pie * sports)

?

Efficient Inference

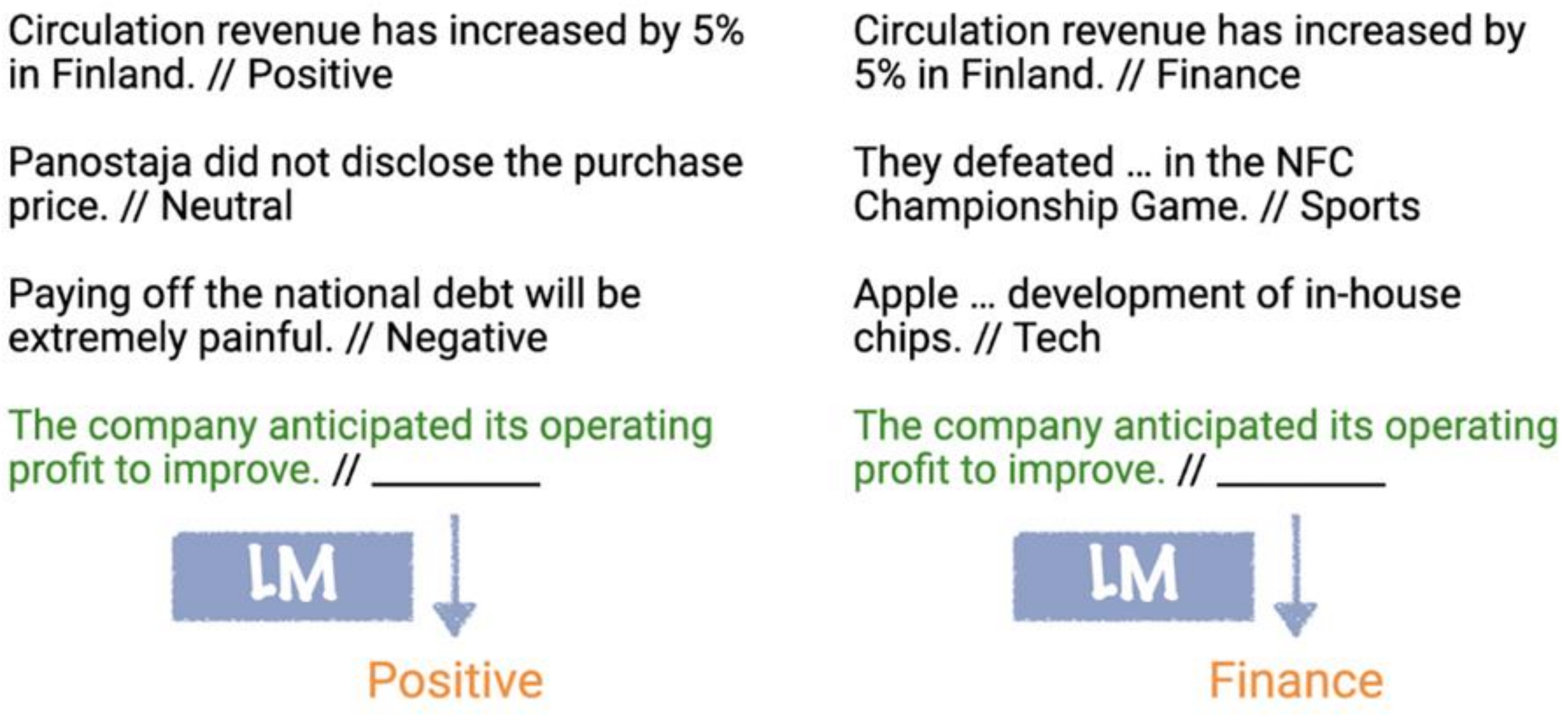




Reasoning Abilities

In-context Learning

In-Context Learning (ICL)



Figures from: *How does in-context learning work? A framework for understanding the differences from traditional supervised learning, 2022.*

Motivation

Simple tasks

Just give me output.
input: * apple
output: APPLE
input: * bird

output: BIRD

Just give me output.
input: (ball book)
output: book ball
input: (house hat)

output: hat house

Composite task

Just give me output.

input: * toe

output: TOE

input: (farm frog)

output: frog farm

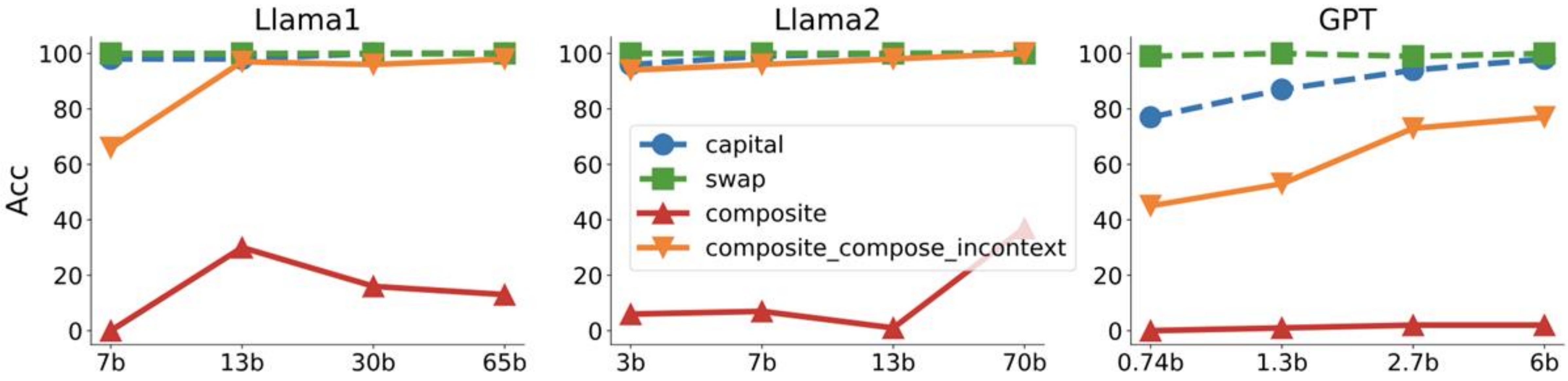
input: (* pie * sports)

output: sports * pie *

A Failure Case for Composition

	Composite	Composite in-context
Prompt	*input: * apple* *output: APPLE* *input: (farm frog)* *output: frog farm* *input: (* bell * ford)*	*input: (* good * zebra)* *output: ZEBRA GOOD* *input: (* bicycle * add)*
Truth	*output: FORD BELL*	*output: ADD BICYCLE*

Failure Case for LLM



Design Experiments to investigate

1. How do LLMs perform in various tasks?

2. Does scaling up the model help in general?

3. Is the variability in performance relevant to the nature of tasks?

Simple Logical Tasks

Tasks	Task	Input	Output
Words	(A) Capitalization	apple	APPLE
	(B) Swap	bell ford	ford bell
	(C) Two Sum	twenty @ eleven	thirty-one
	(D) Past Tense	pay	paid
	(E) Opposite	Above	Below
Numerical	(F) Plus One	435	436
	(G) Modular	15 @ 6	3
	(H) Two Sum Plus One	12 # 5	18

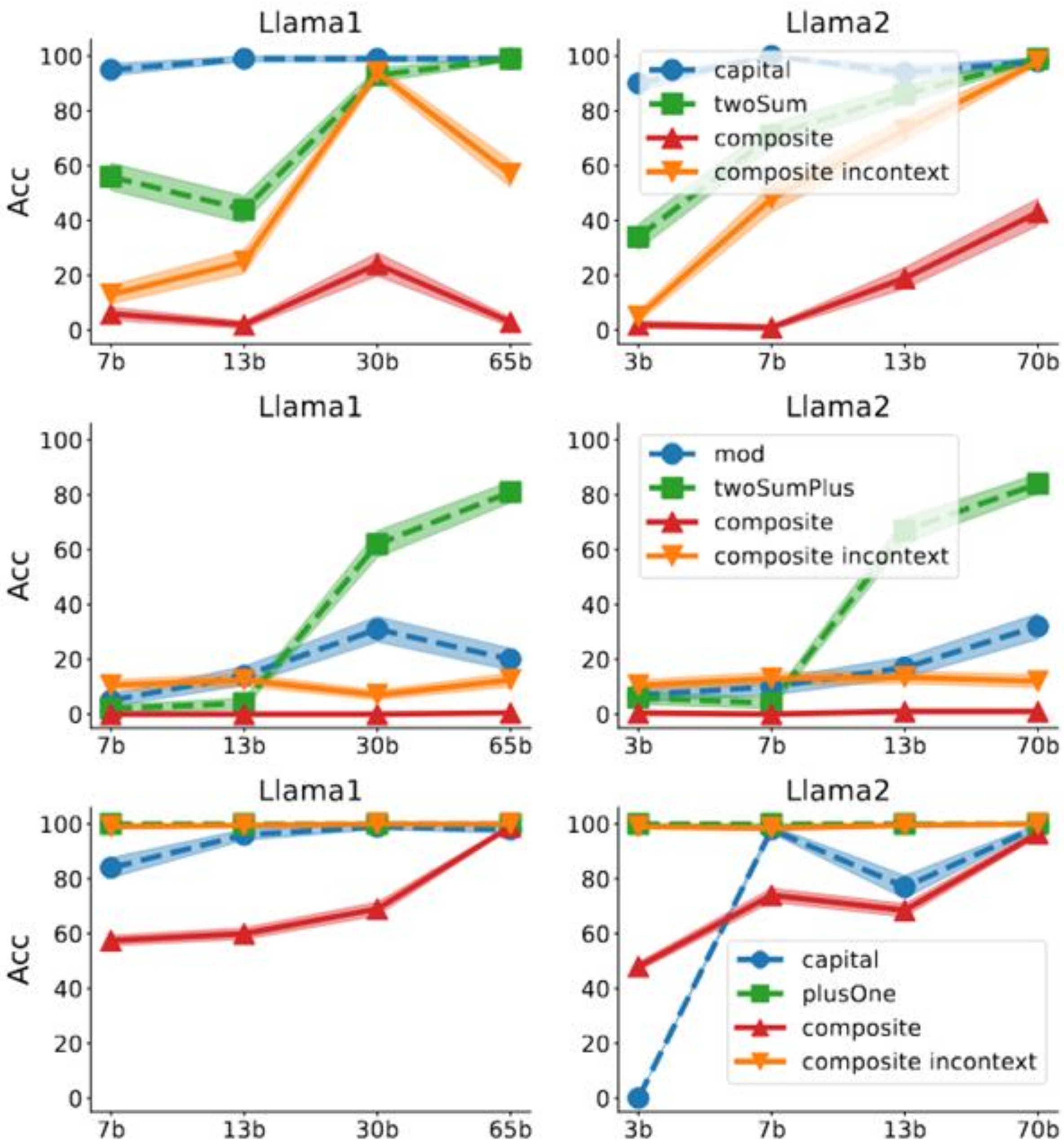
Compositional Logical Tasks

Tasks	Simple Task	Simple Task	Composite
(A) + (B)	input: * apple output: APPLE	input: (farm frog) output: frog farm	input: (* bell * ford) output: FORD BELL
(A) + (C)	input: * (*five*) output: FIVE	input: *twenty* @ *eleven* output: thirty-one	input: * (*thirty-seven* @ *sixteen*) output: FIFTY-THREE
(G) + (H)	input: 15 @ 6 output: 3	input: 12 # 5 output: 18	input: 8 # 9 @ 7 Ouput: 4
(A) + (F)	input: 435 output: 436	input: cow output: COW	input: 684 cat output: 685 CAT

(A) + (C)

(G) + (H)

(A) + (F)


Llama1
Llama2
Acc
7b
13b
30b
65b
3b
7b
13b
70b
capital
twoSum
composite
composite incontext
Llama1
Llama2
Acc
mod
twoSumPlus
composite
composite incontext
Llama1
Llama2
Acc
capital
plusOne
composite
composite incontext

Compositional Ability

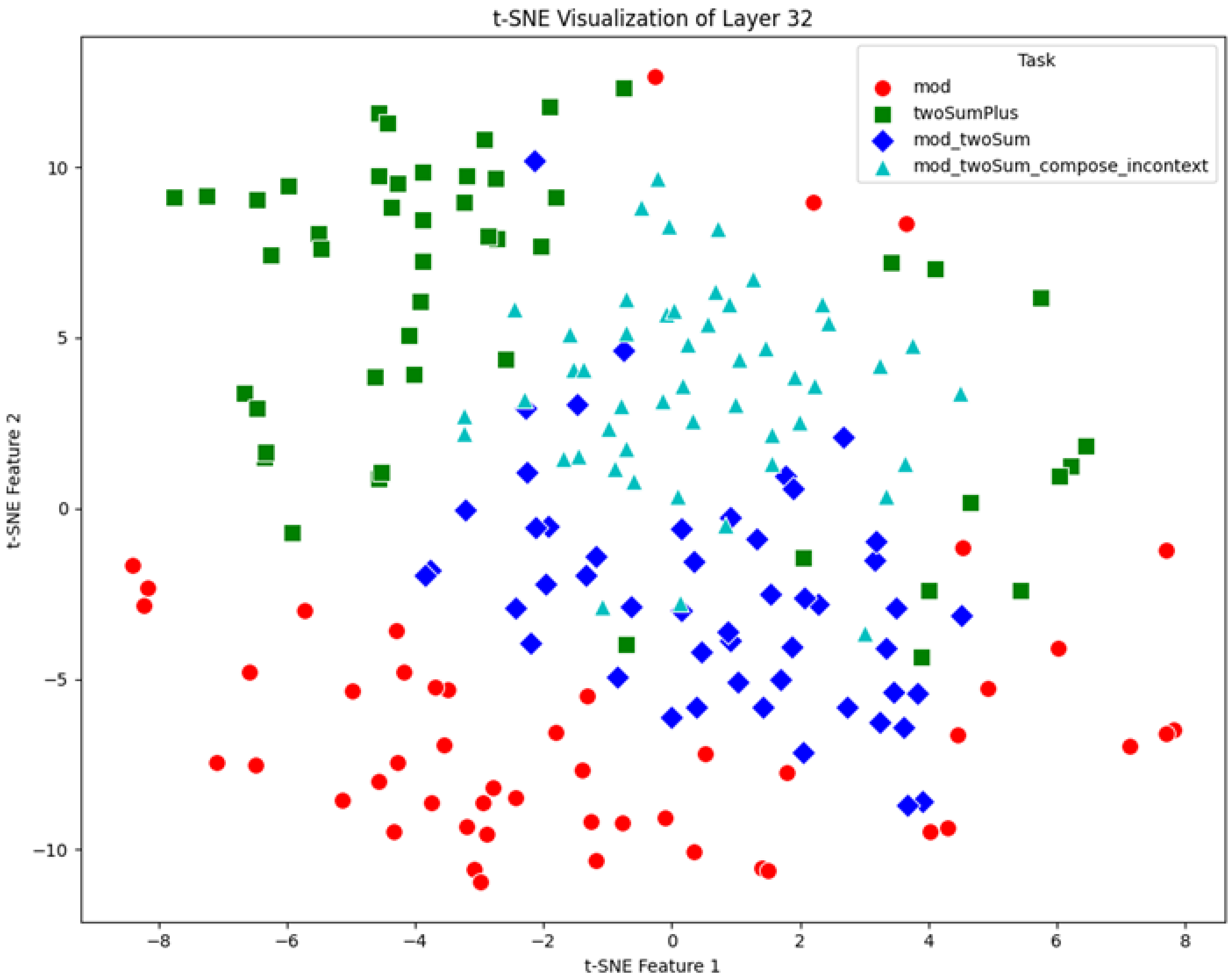
Definition1 (Compositional Ability)
Consider a composite tasks combines two simple tasks (A) and (B).
Consider each simple tasks contains samples .

Given a composite test prompt, we say model has compositional ability on composite task (A) + (B) if model has higher accuracy using in-context examples from both (A) and (B) than from either single one.

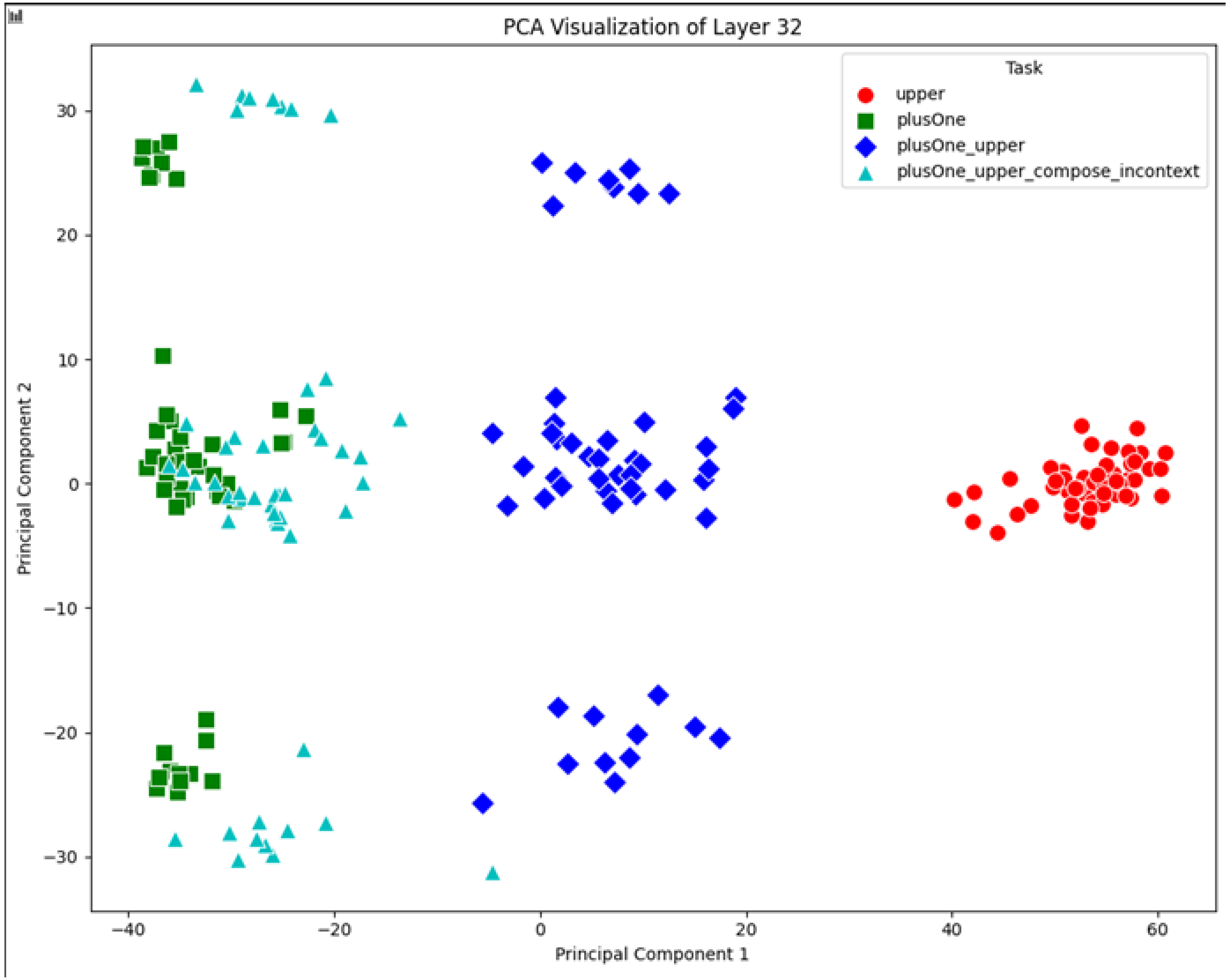
Compositional ability under confined support (Informal)

Theorem (Compositional ability under confined support (Informal))

Consider input embedding $x \in \mathbb{R}^d$ of each simple tasks. Consider each simple has a disjoint subset of indices from $1, 2, \ldots, d$. Each simple task only has large values within its corresponding subsets of dimensions of input embeddings. Then with high probability, the model has the compositional ability.



(G) + (H)	input: 15 @ 6 output: 3	input: 12 # 5 output: 18	input: 8 # 9 @ 7 Ouput: 4



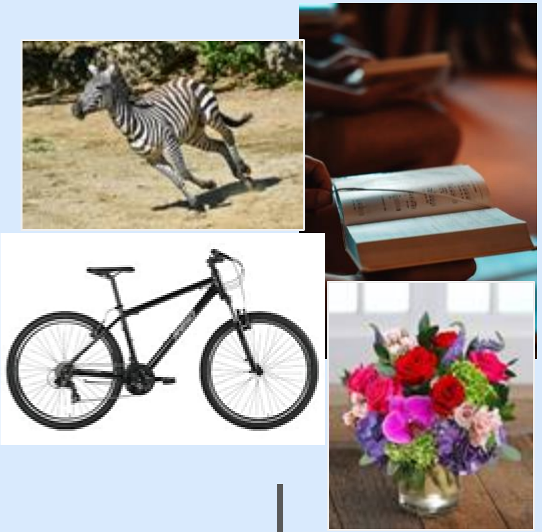
(A) + (F)	input: 435 output: 436	input: cow output: COW	input: 684 cat output: 685 CAT

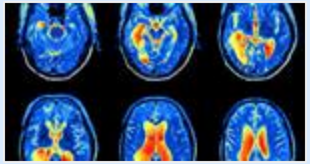
ICL for compositional

Our findings on compositional ability in LLMs reveal:

1. Simple composition (distinct mappings on different inputs): Models perform well and benefit from scaling
2. Complex composition (multi-step reasoning): Models struggle, with limited gains from scaling

Data Distribution Shift





Compositional Ability

* bird

BIRD

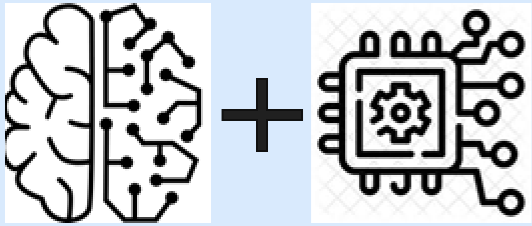
(house hat)

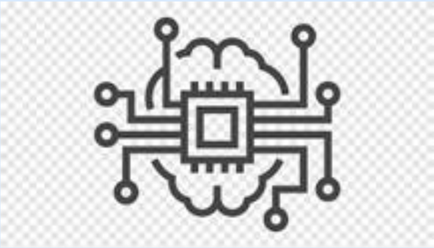
hat house

(* pie * sports)

?

Efficient Inference





Adaptive Inference in Multimodal LLMs (Preliminary work)

Current MLLMs lack adaptability to meet varying latency constraints in resource-limited environments.

Prior approaches for MLLM efficiency provide static efficiency improvement:

- Compress models to fixed smaller size
- Use predetermined token selection strategies

Adaptive Inference in Multimodal LLMs (Preliminary work)

Input image

What is the image showing?

LLaVA

The image shows a snowman holding a colorful egg. — 8 TFLOPs

AdaLLaVA

50% latency budget

The image is showing a painting or drawing of a snowy winter scene. — **4 TFLOPs**

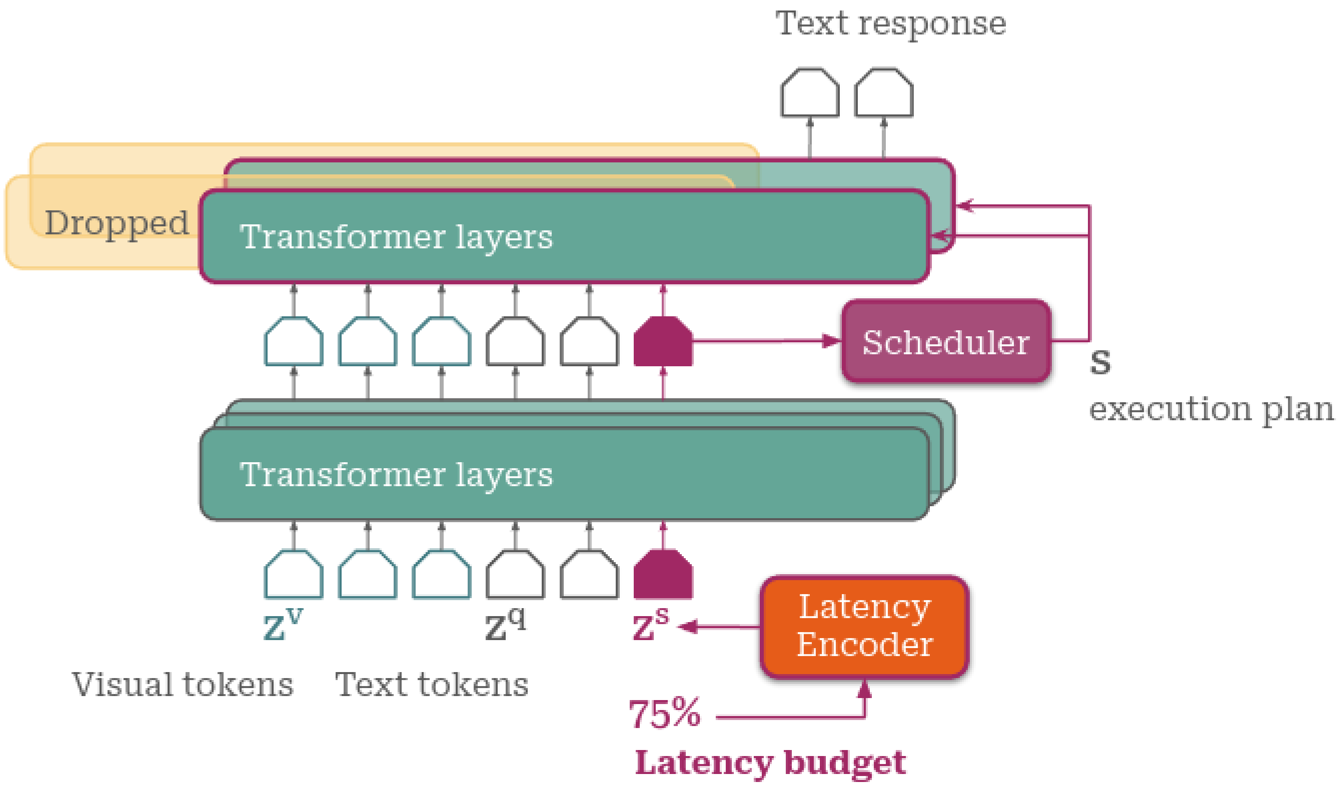
75% latency budget

The image shows a snowman holding a colorful ball. — **6 TFLOPs**

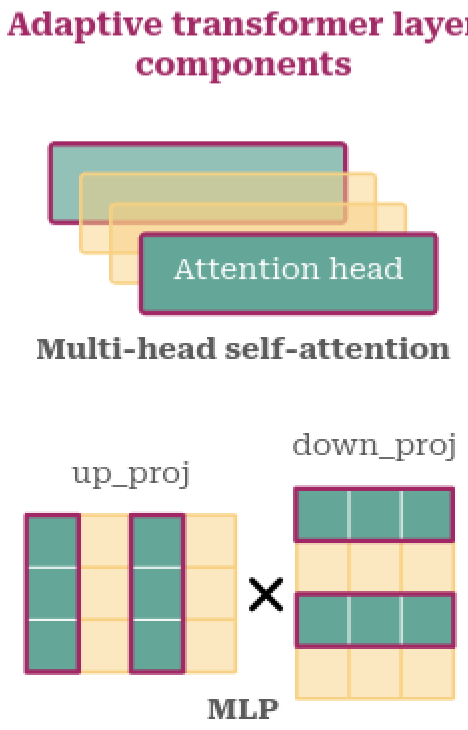
100% latency budget

The image shows a snowman holding a colorful egg. — **8 TFLOPs**

Adaptive Inference in Multimodal LLMs (Preliminary work)

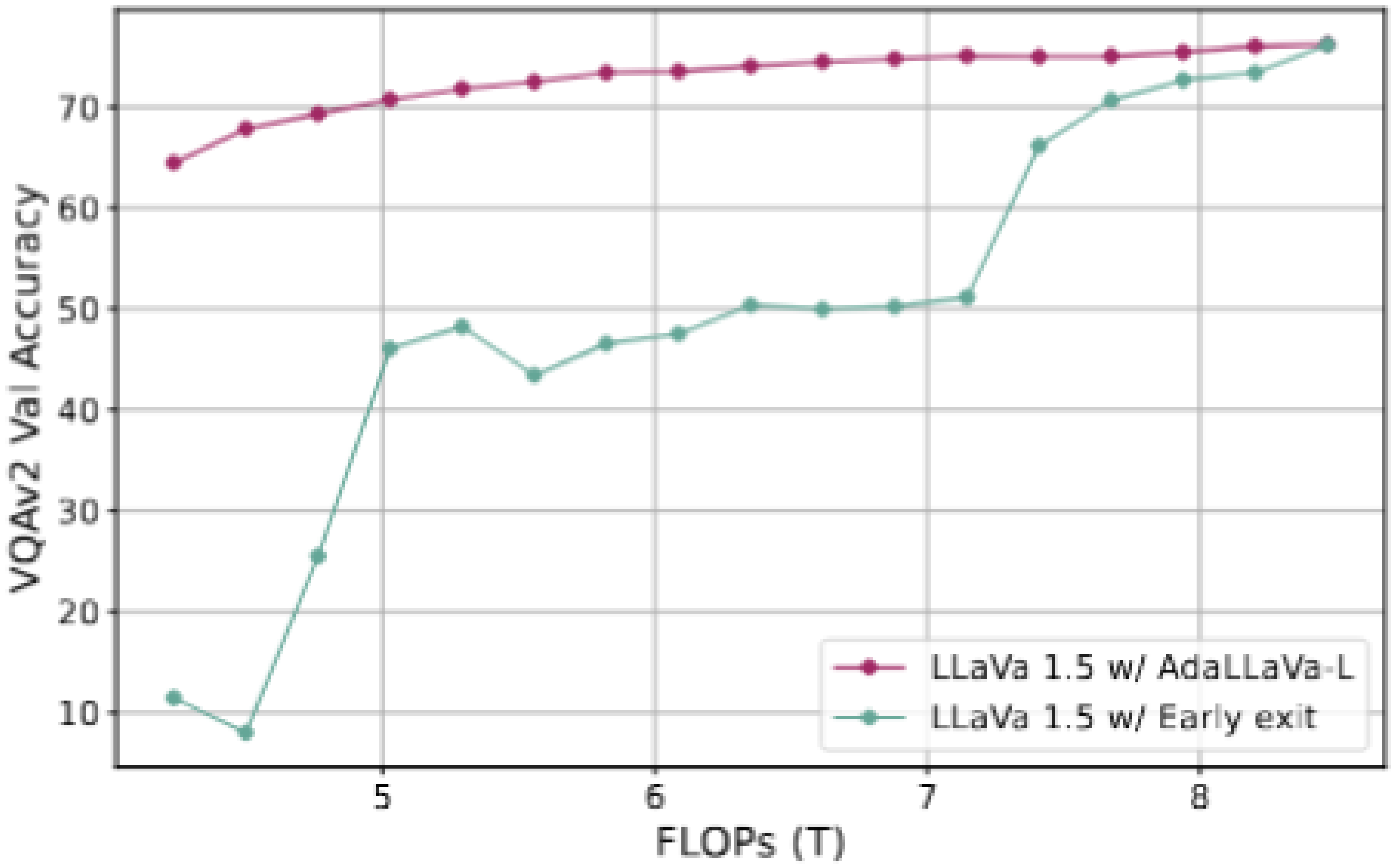


(a)



(b)

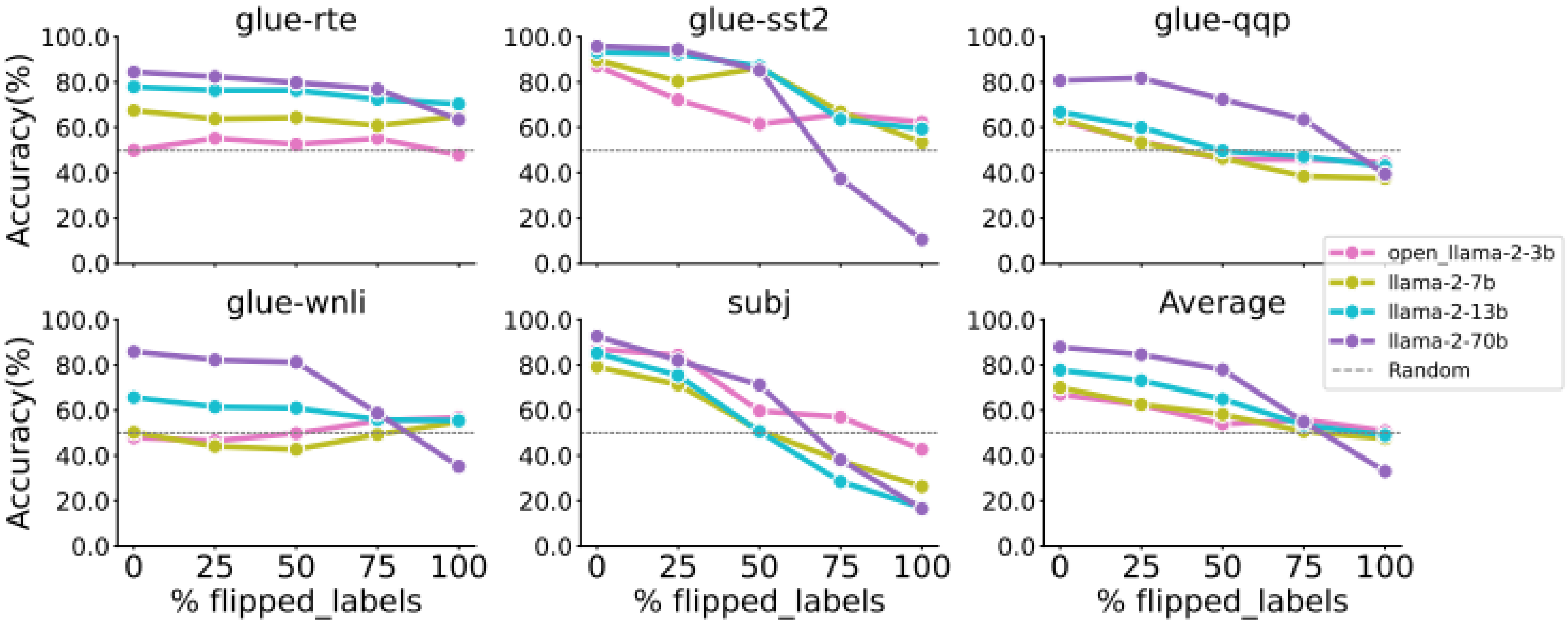
Adaptive Inference in Multimodal LLMs (Preliminary work)



Discussions

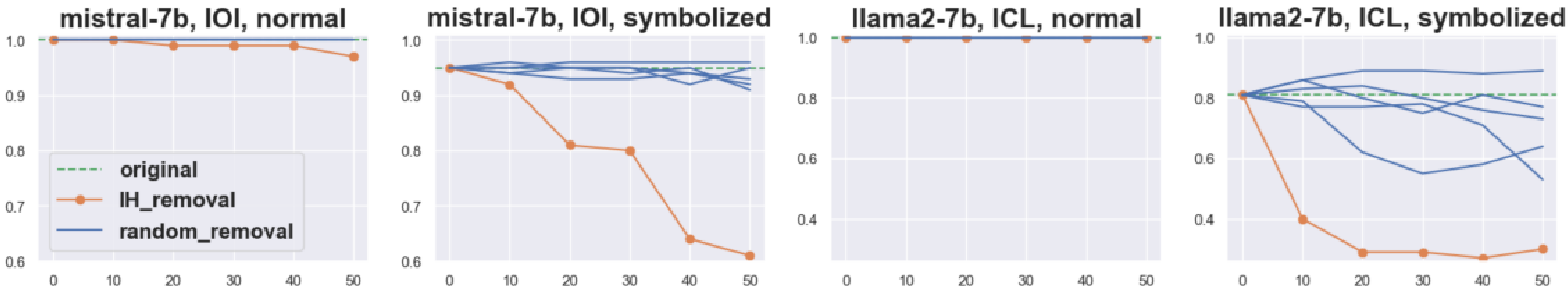
Additional Works

Scale Effects in In-Context Learning



Zhenmei Shi, Junyi Wei, Zhuoyan Xu, and Yingyu Liang. Why larger language models do in-context learning differently? ICML, 2024.

OOD Generalization Through Induction Heads



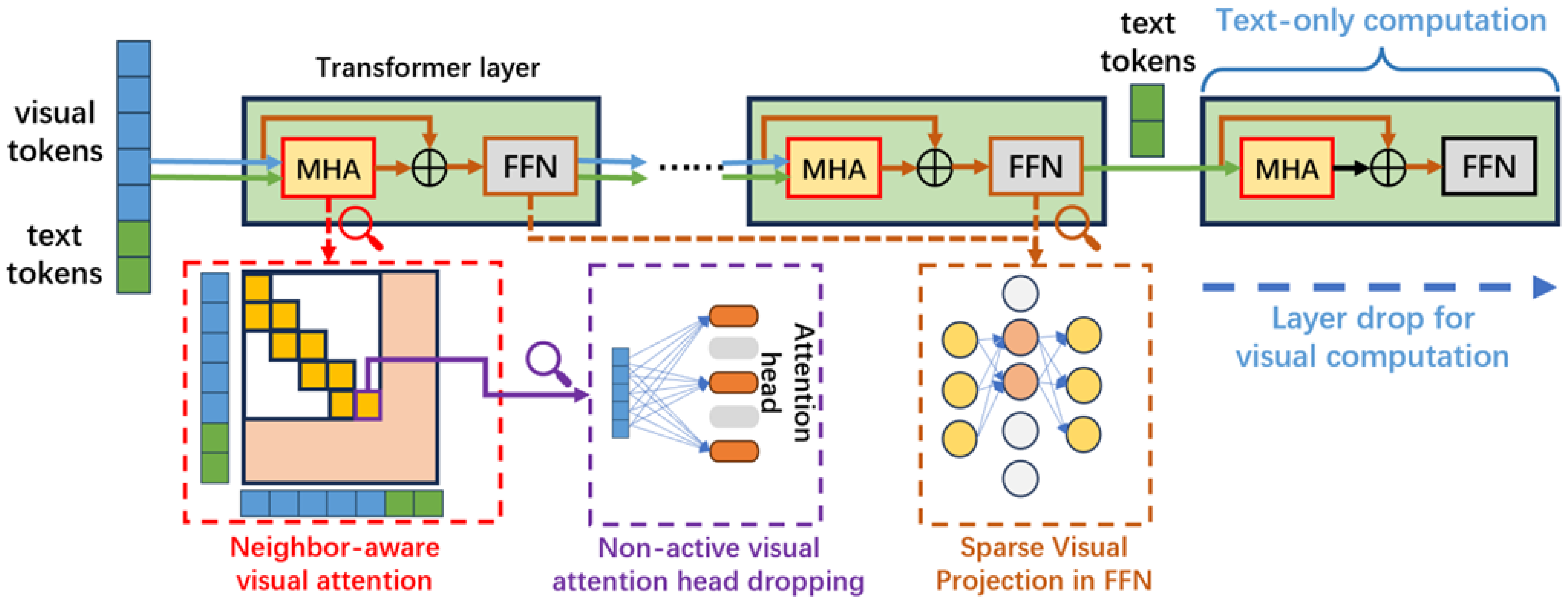
Jiajun Song, Zhuoyan Xu, and Yiqiao Zhong. Out-of-distribution generalization via composition: a lens through induction heads in transformers.

Proposed Works

Efficient Architecture-Guided LLM inference

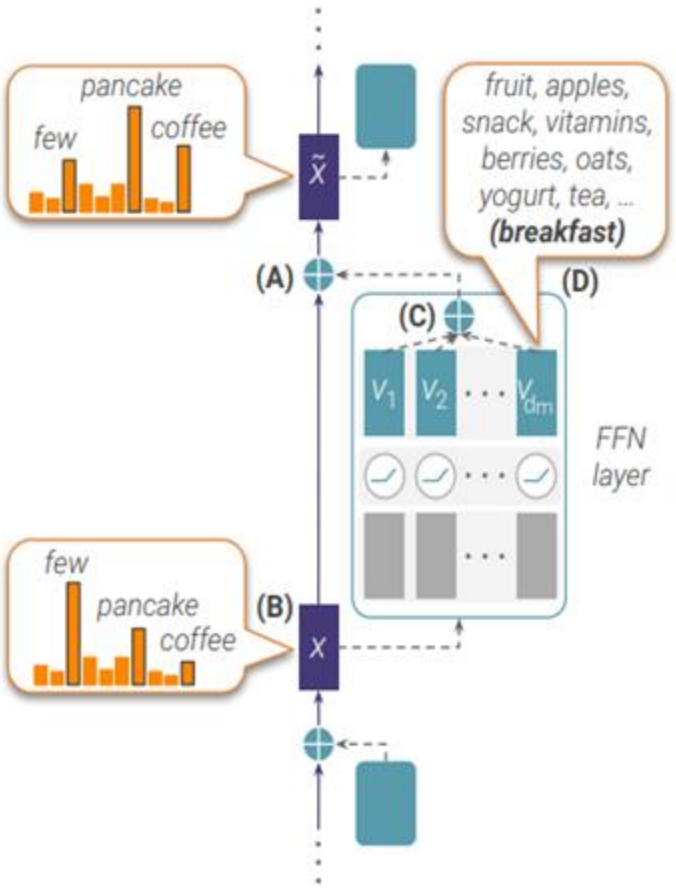
Efficient LLM Inference

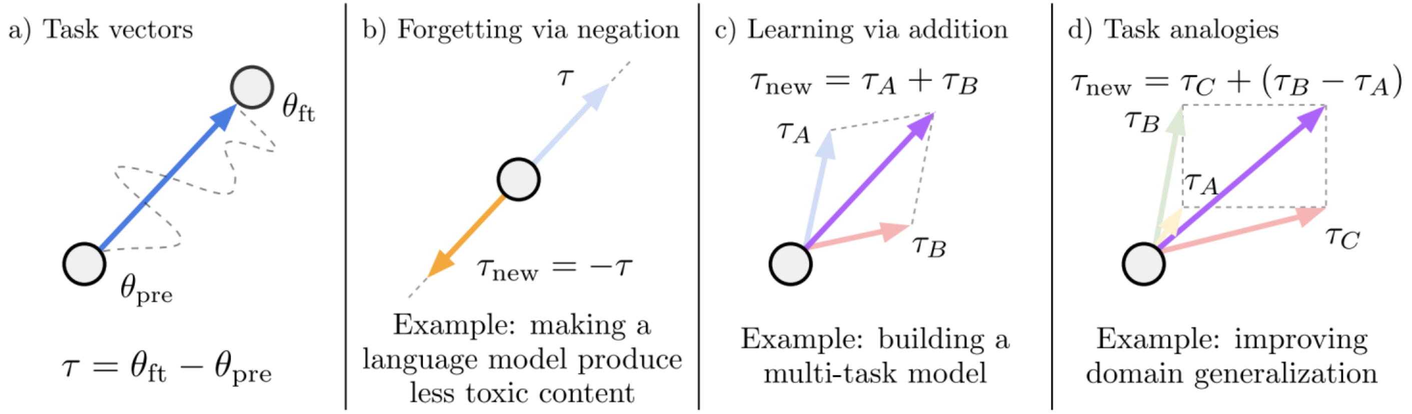
Computation-level optimization:



Zeliang Zhang et al. Treat Visual Tokens as Text? But Your MLLM Only Needs Fewer Efforts to See

Investigations on Architecture





Gabriel Ilharco. Editing Models with Task Arithmetic. ICLR 2023

Mor Geva. Transformer Feed-Forward Layers Build Predictions by Promoting Concepts in the Vocabulary Space. EMNLP 2022

Motivations

1. Our prior work showed promise in adaptive inference of MLLMs
2. Recent mechanistic interpretability findings reveal key architectural components for different tasks

Can we combine them and develop training-free algorithms for component selection, specialized for different tasks?

Efficient LLM Inference

1. Token-Level Optimization

    a. Develop dynamic token selection strategies based on input

2. Mechanistically-Guided Component Selection

    a. Identify crucial model components (attention heads) and patterns using interpretability insights

3. Task-specific Patch

    a. Develop training-free algorithms for component selection

Expected Outcome

1. New algorithms for interpretability-guided model inference

2. Improved understanding of architectural components in reasoning

3. Significant efficiency gains for specific tasks

Timeline

Adainf	Efficient LLM inference	Wrap up + Thesis writing	Defence
Jan 2025	May 2025	Fall 2025	Dec 2025

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON