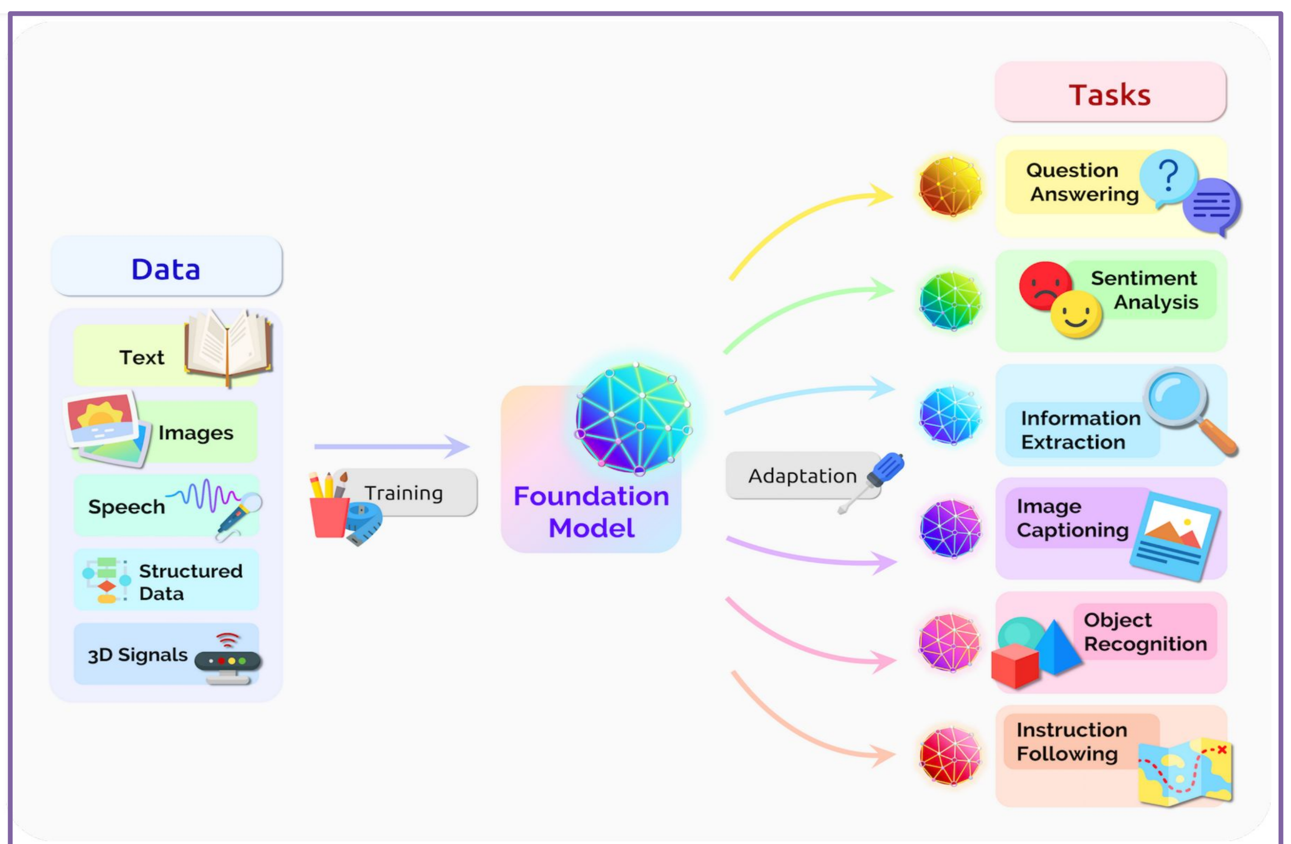
Improving Foundation Models for Few-Shot Learning via Multitask Finetuning

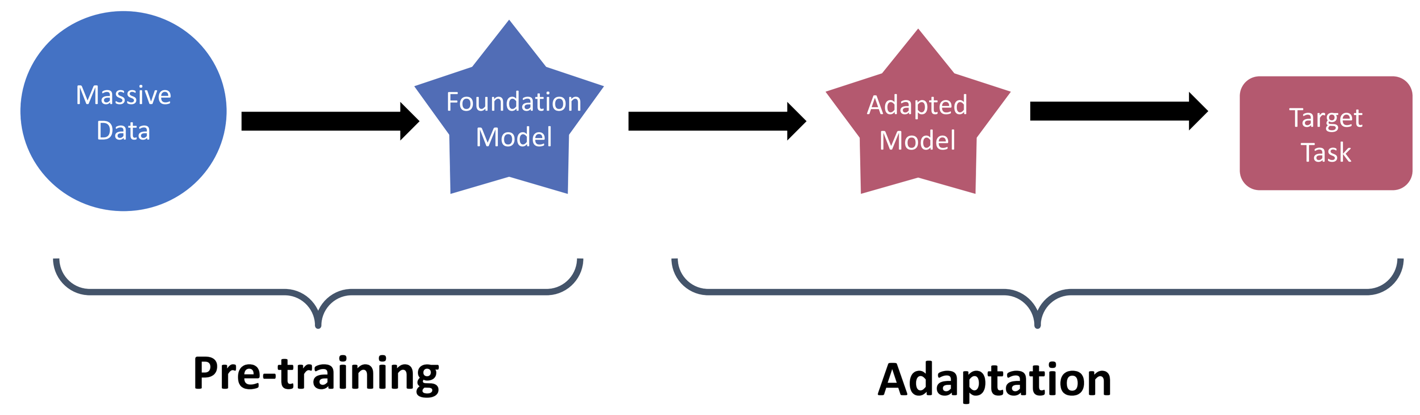
Zhuoyan Xu, Zhenmei Shi, Junyi Wei, Yin Li, Yingyu Liang

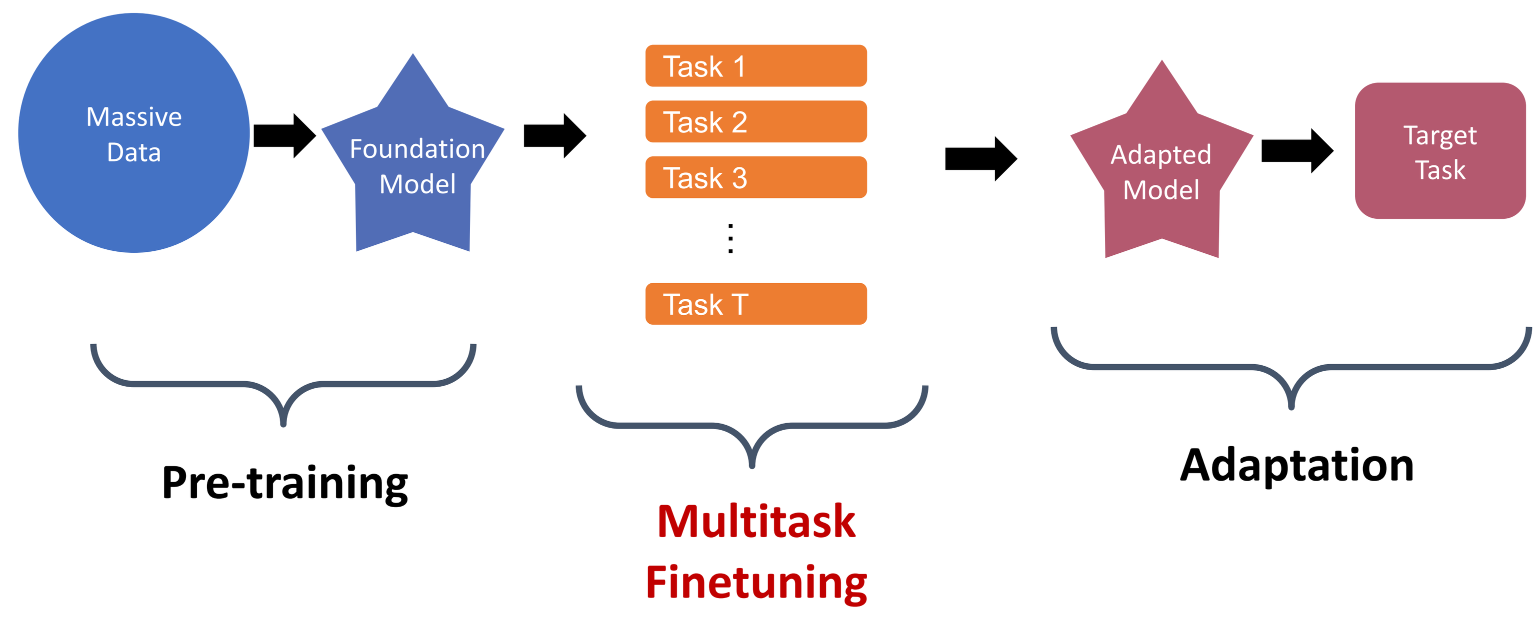
Motivation



Foundation Model

Source "On the opportunities and risks of foundation models." (2021)



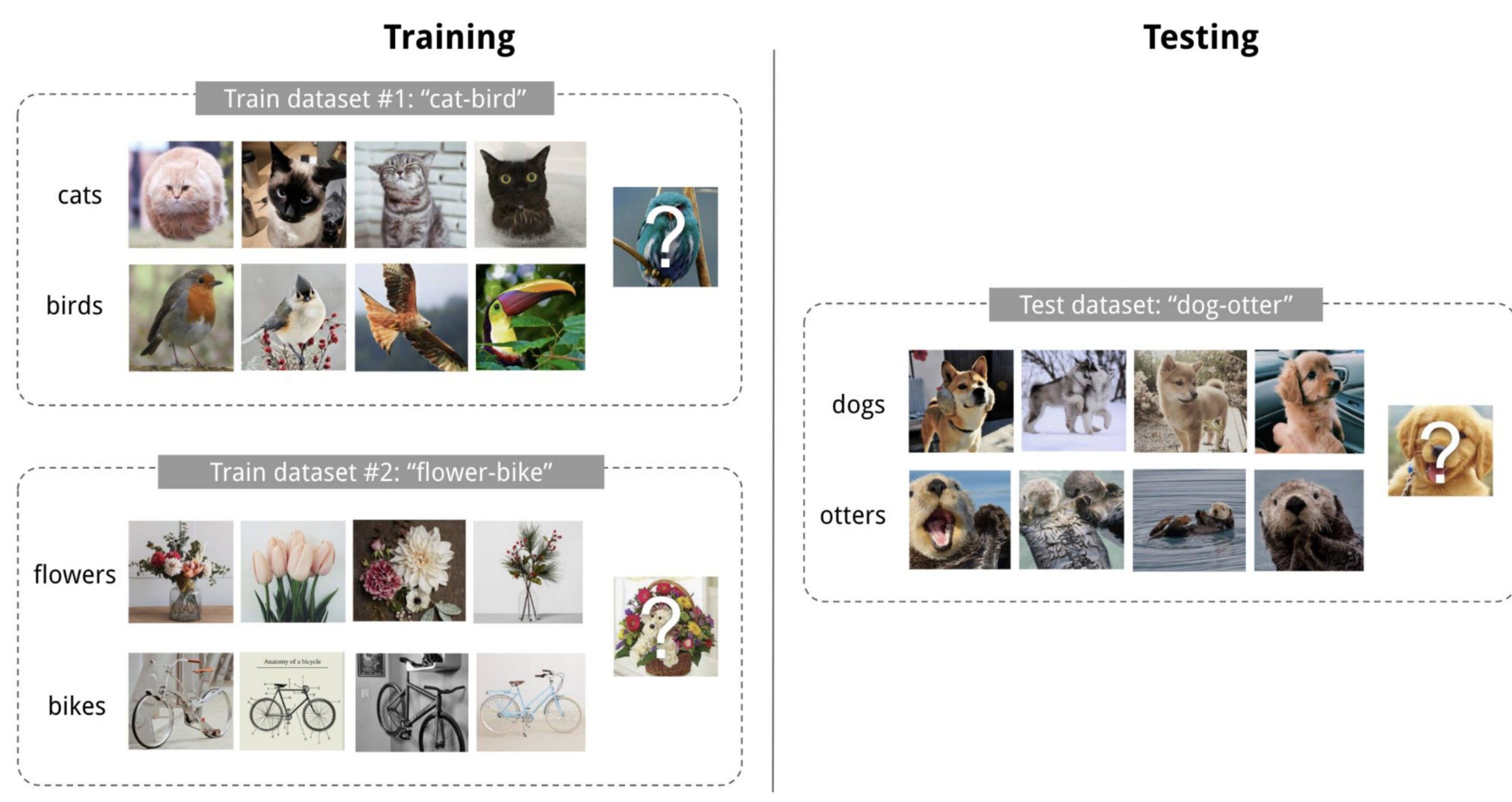


Take-Home Message

We use a paradigm that first finetunes a foundation model with multiple relevant tasks before adapting it to a target task.

Key Intuition

- Pre-training uses unlabeled and noisy data for general purpose learning, where the model learns representation rather than task-specific knowledge. Its performance on a specific task may only be adequate.
- Although the target data is limited, we have a clear understanding of the target task and its associated data.
  - We select additional data from a relevant source that covers its characteristic data.
  - We construct specific tasks for multitask finetuning to allow the model to handle the particular types of target tasks.



An example of 4-shot 2-class image classification

Source: "Meta-Learning: Learning to Learn Fast", 2018.

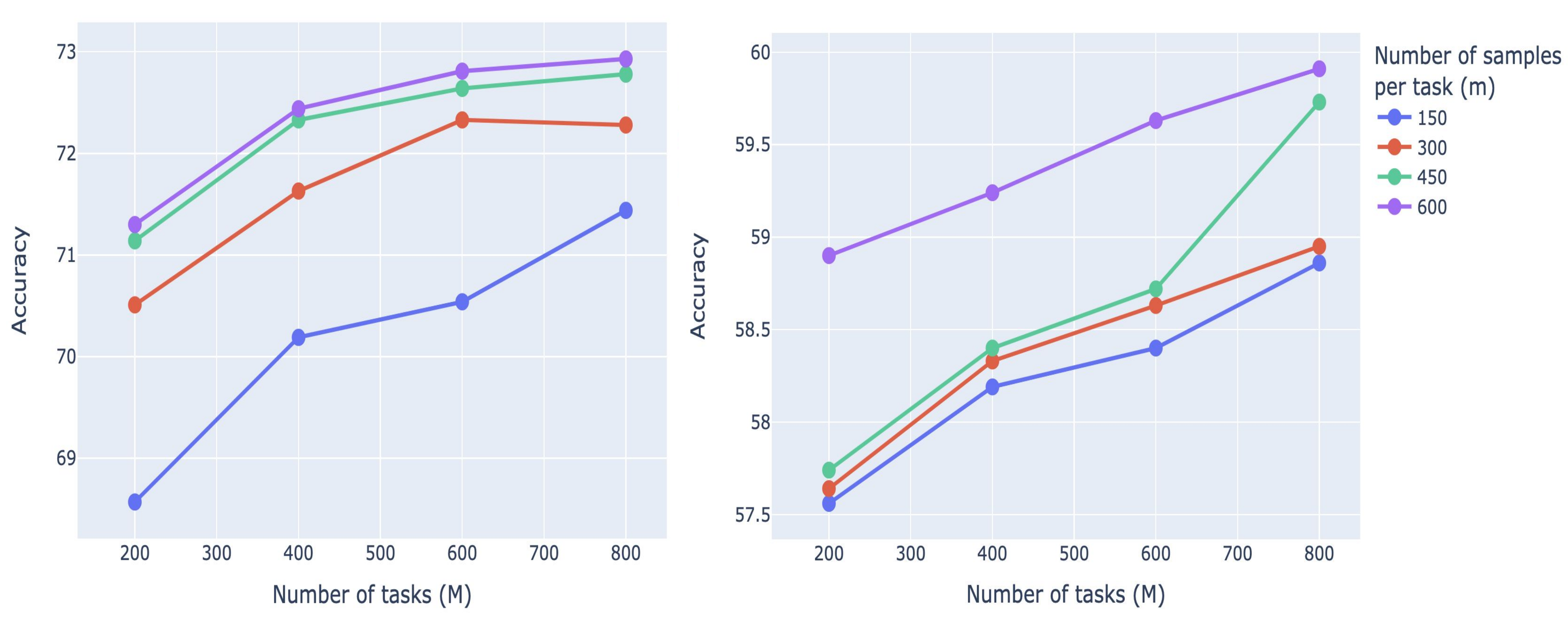
Experiments

Few-shot Vision tasks

15-way accuracy (%) on *tiered-ImageNet*, 1 image per class in target task

Backbone	Direct Adaptation	Finetuning
ViT-B32	59.55 ± 0.21	**68.57** ± 0.37
ResNet50	51.76 ± 0.36	**57.56** ± 0.36

200 finetuning tasks, 150 images per task



ViT-B32 ResNet50

Accuracy with varying number of tasks and samples

Few-shot Language tasks

Text classification for different text dataset, with prompt-base finetuning

	SST-2 (acc)	SST-5 (acc)	MR (acc)	CR (acc)	MPQA (acc)	Subj (acc)	TREC (acc)	CoLA (Matt.)
Prompt-based zero-shot	83.6	35.0	80.8	79.5	67.6	51.4	32.0	2.0
Multitask FT zero-shot	**92.9**	37.2	86.5	88.8	73.9	55.3	36.8	-0.065
Prompt-based FT†	92.7 (0.9)	47.4 (2.5)	87.0 (1.2)	90.3 (1.0)	84.7 (2.2)	**91.2** (1.1)	84.8 (5.1)	**9.3** (7.3)
Multitask Prompt-based FT	92.0 (1.2)	**48.5** (1.2)	86.9 (2.2)	90.5 (1.3)	**86.0** (1.6)	89.9 (2.9)	83.6 (4.4)	5.1 (3.8)
+ task selection	92.6 (0.5)	47.1 (2.3)	**87.2** (1.6)	**91.6** (0.9)	85.2 (1.0)	90.7 (1.6)	**87.6** (3.5)	3.8 (3.2)

	MNLI (acc)	MNLI-mm (acc)	SNLI (acc)	QNLI (acc)	RTE (acc)	MRPC (F1)	QQP (F1)
Prompt-based zero-shot	50.8	51.7	49.5	50.8	51.3	61.9	49.7
Multitask FT zero-shot	63.2	65.7	61.8	65.8	74.0	81.6	63.4
Prompt-based FT†	68.3 (2.3)	70.5 (1.9)	77.2 (3.7)	64.5 (4.2)	69.1 (3.6)	74.5 (5.3)	65.5 (5.3)
Multitask Prompt-based FT	70.9 (1.5)	73.4 (1.4)	**78.7** (2.0)	71.7 (2.2)	**74.0** (2.5)	**79.5** (4.8)	67.9 (1.6)
+ task selection	**73.5** (1.6)	**75.8** (1.5)	77.4 (1.6)	**72.0** (1.6)	70.0 (1.6)	76.0 (6.8)	**69.8** (1.7)

Our main results using RoBERTa-large. †: Result in (GFC20);

[GFC20] "Gao, Fisch, and Chen. Making pre-trained language models better few-shot learners." ACL'2020.

Zero-shot Vision-Language tasks

160(all)-way zero-shot accuracy (%) on *tiered-ImageNet* test split

Backbone	Zero-shot	Multitask finetune
ViT-B32	69.9	71.4

Effects of multitask finetuning

Theoretical Analysis

Contrastive Learning

Objective function: $\mathcal{L}_{un}(\phi) := \mathbb{E}\left[-\log\left(\frac{e^{\phi(x)^\top \phi(x^+)}}{e^{\phi(x)^\top \phi(x^+)} + e^{\phi(x)^\top \phi(x^-)}}\right)\right]$

Supervised loss respect to a task $\mathcal{T}$, W is a linear classifier:

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

Multitask finetuning

Suppose we construct M tasks, each with m sample

$$\min_{W_i \in \mathbb{R}^d, \phi \in \Phi} \frac{1}{M}\sum_{i=1}^{M}\frac{1}{m}\sum_{j=1}^{m}\ell\left(W_i \cdot \phi\left(x_j^i\right), z_j^i\right), \quad \text{s.t.} \quad \widehat{\mathcal{L}}_{un}(\phi) \le \epsilon_0$$

Hidden Representation Data Model

- First sampling the latent class, and then sampling input.
- In contrastive pre-training, positive pair sampled from the same latent class.
- A task $\mathcal{T}$ contains a subset of latent classes.

Proposition of target task error (Informal)

Suppose in pre-training we have target task error bounded by ε with high probability, our multitask fineutning reduce error on target task to $\alpha\varepsilon$, where finetuning sample complexity is $\theta(1/\alpha\varepsilon)$.