# Adaptive Transfer Learning from Pre-trained Models

Mingsheng Long

School of Software, Tsinghua University

August 24, 2022

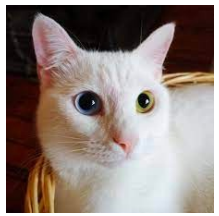


# Supervised Training

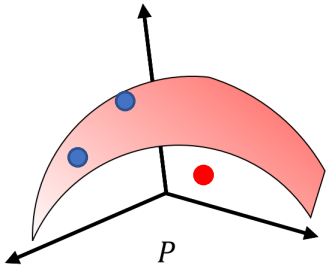
Training images, scarce


Dog


Cat


Cat

Hard to generalize well




Deep Model

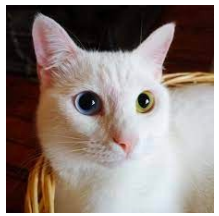
Heavy parameters for large capacity

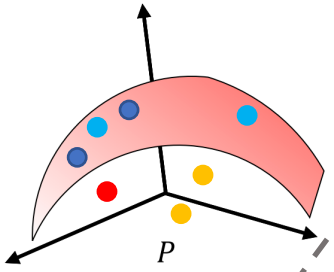
# Pre-training

Training images, scarce


Dog


Cat


Cat




Supervised pre-training

Pre-trained Model (PTM)

Unsupervised pre-training

Obtains transferable knowledge


Fox

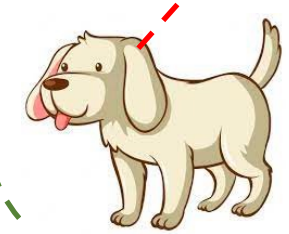

Wolf


Lynx


Owl

Related images, labeled

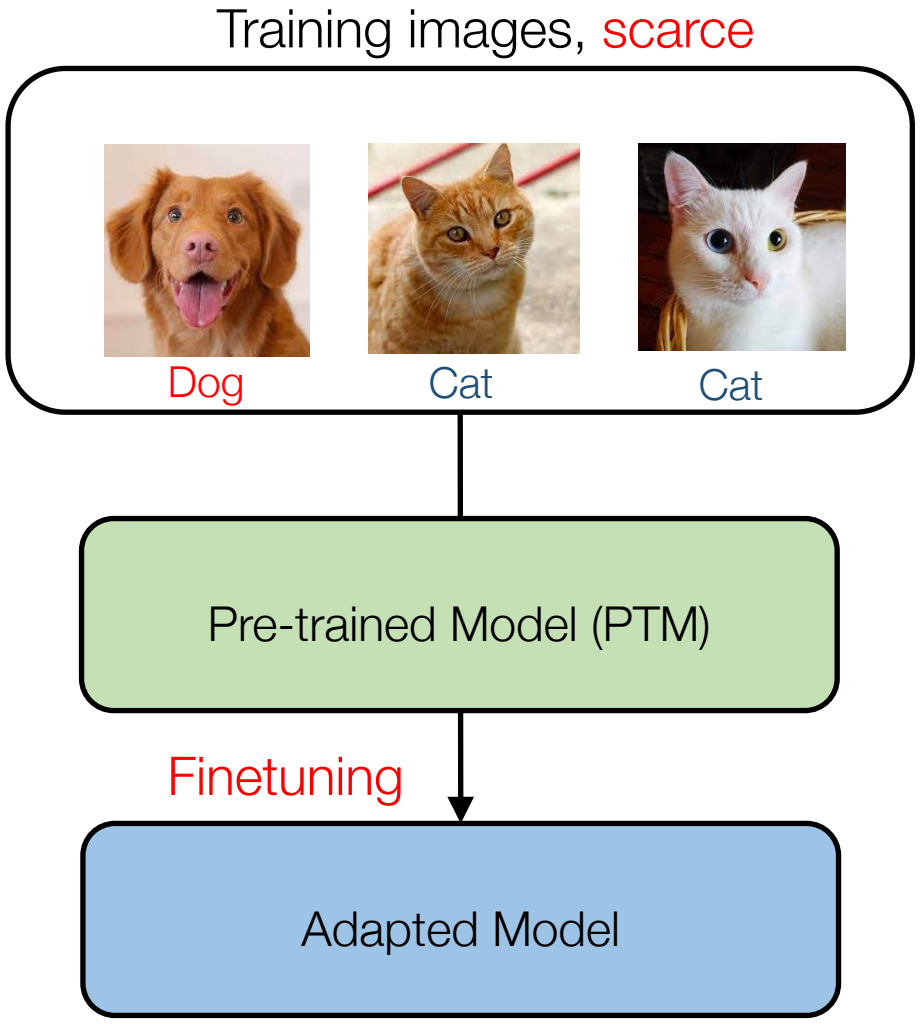
Contrastive Learning



About Cat
Description
…

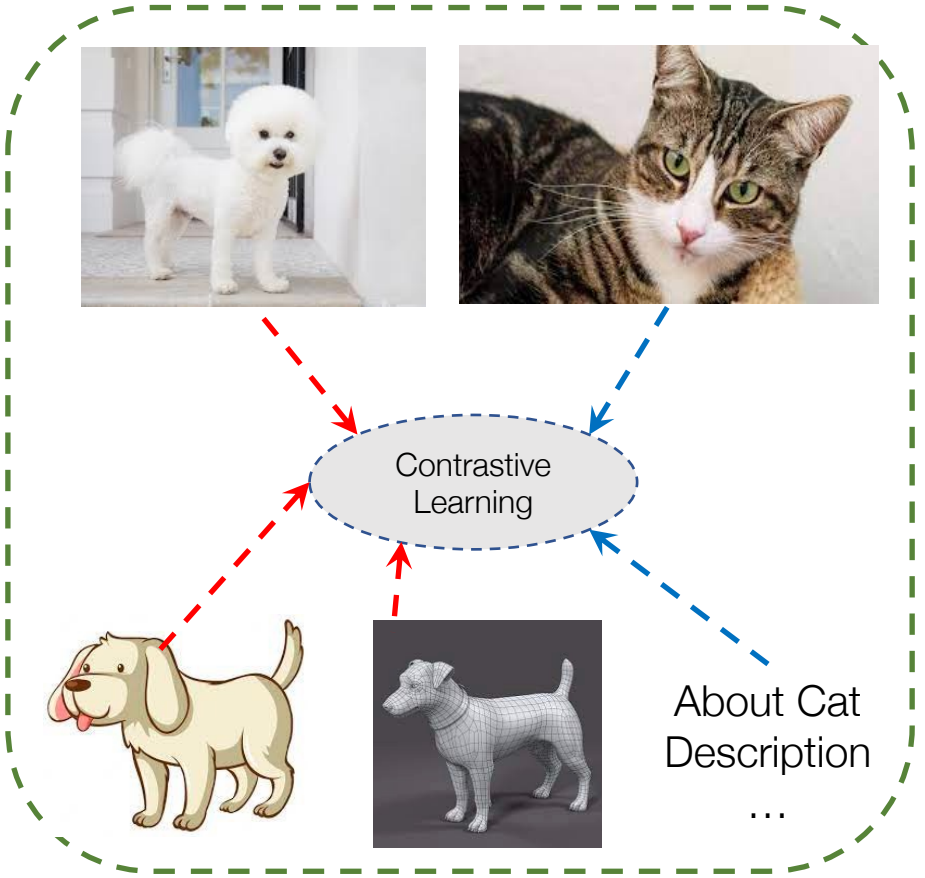
Related data, unlabeled

# Adaptation



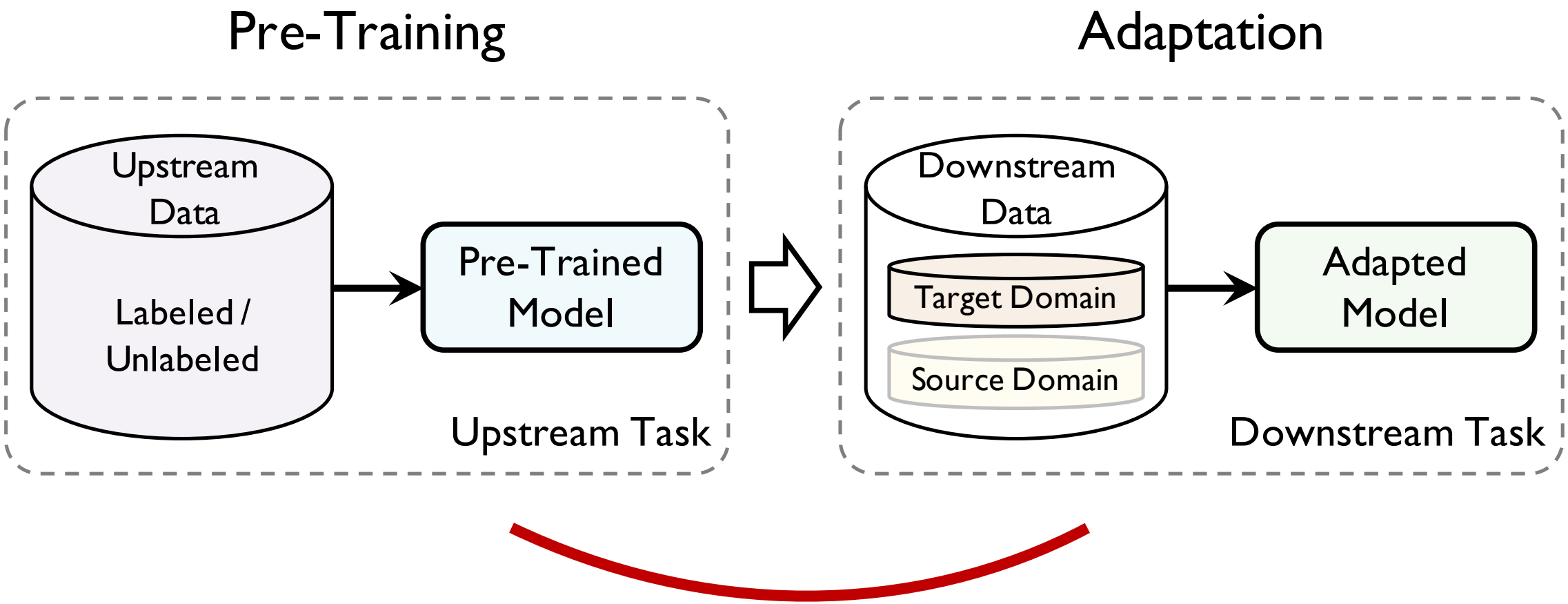


Related images, labeled




Related data, unlabeled

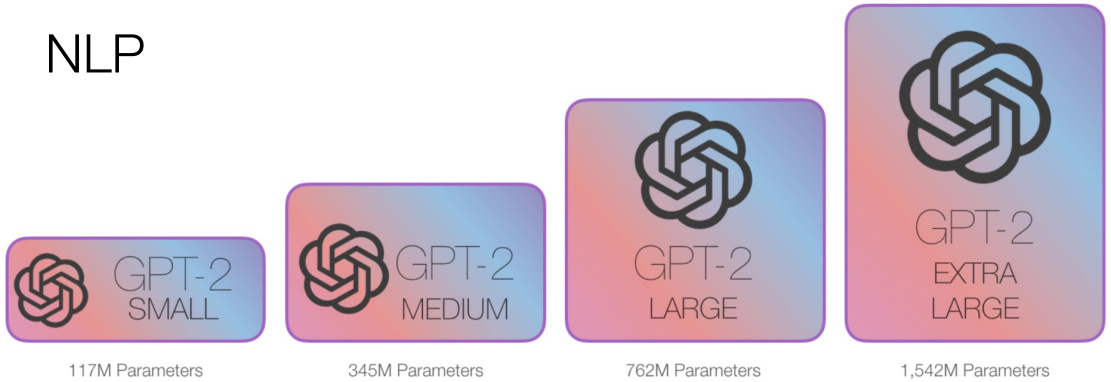
# Pre-training and Adaption




A Paradigm for Deep Learning Application

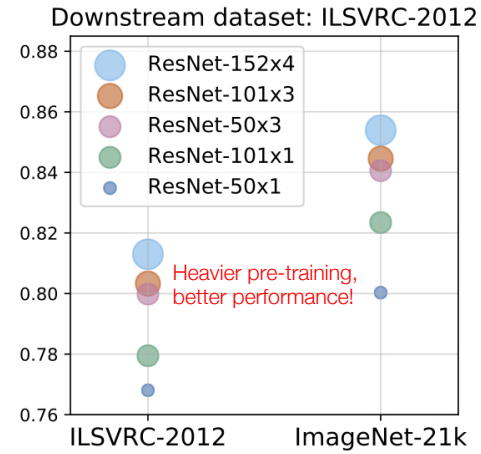
# Pre-trained Models

NLP




GPT: Large-scale Corpus Pre-training

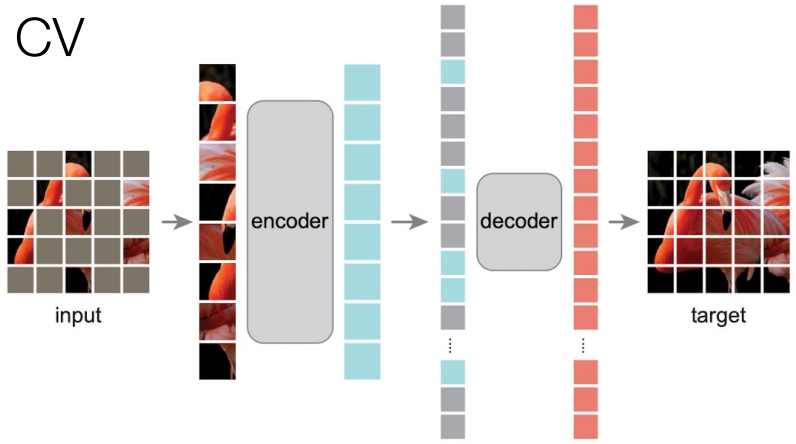
CV




BiT: General Visual Representation Learning

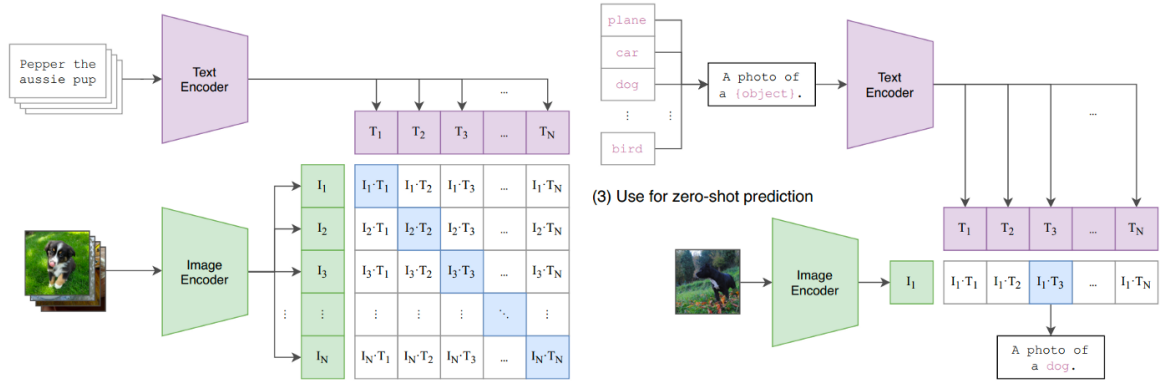
GPT 2018 → GPT-2, BERT 2019 → GPT-3, BiT, MoCo 2020 → MAE, CLIP 2021

CV



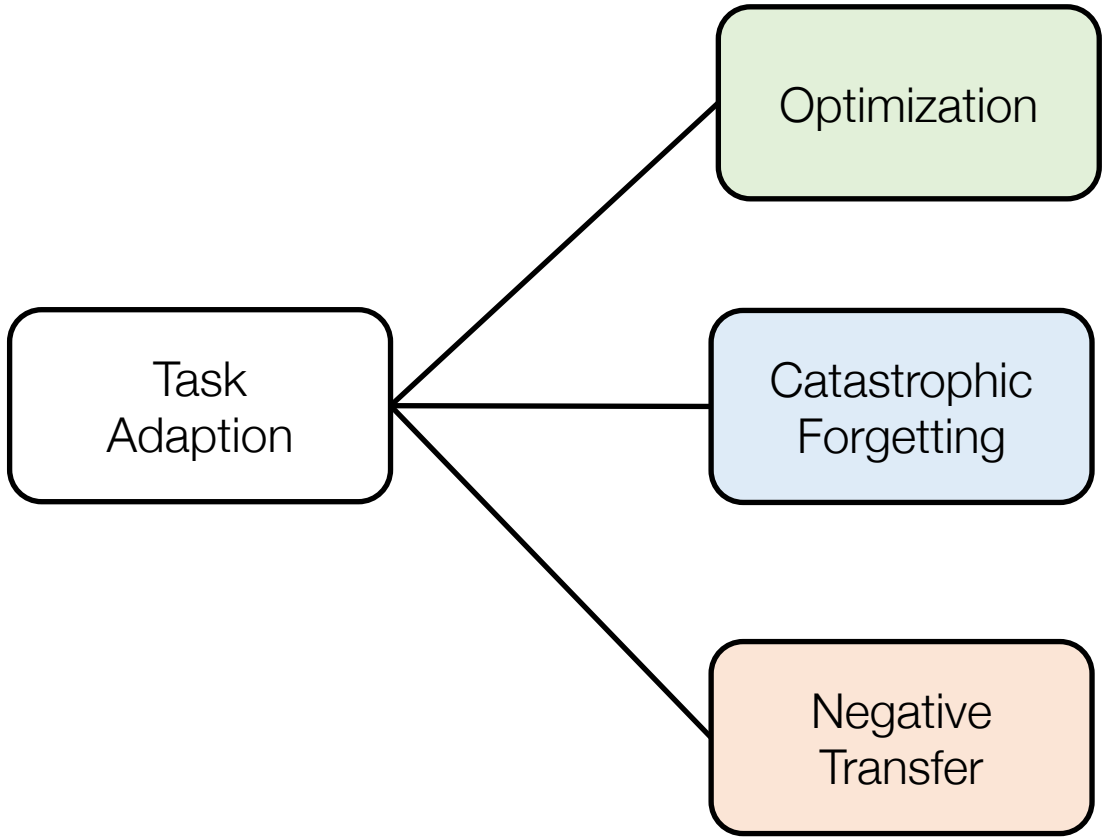

MAE: Masked AutoEncoder as Self-supervised Learner

All In One



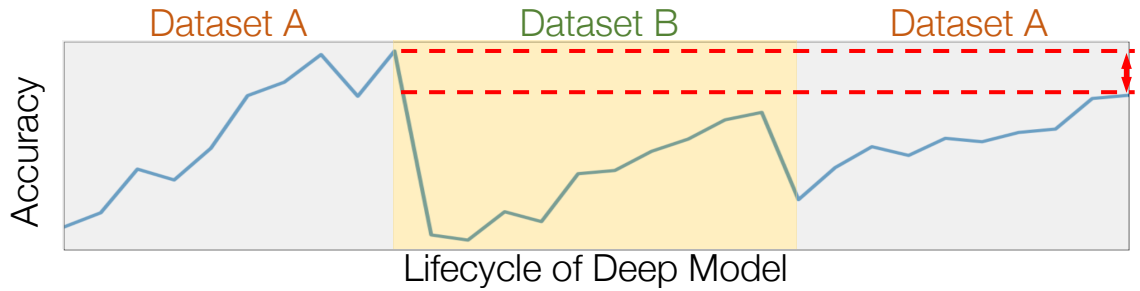

CLIP: Contrastive Language-Image Pre-Training

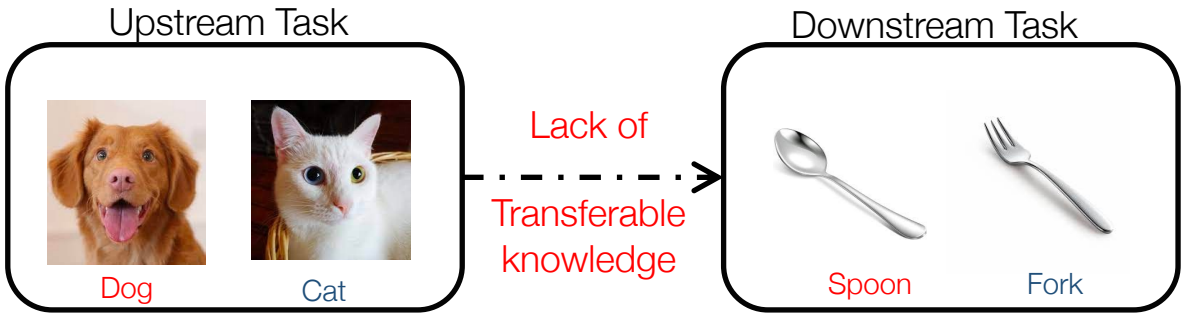
# Adaptation: Foundation Problems

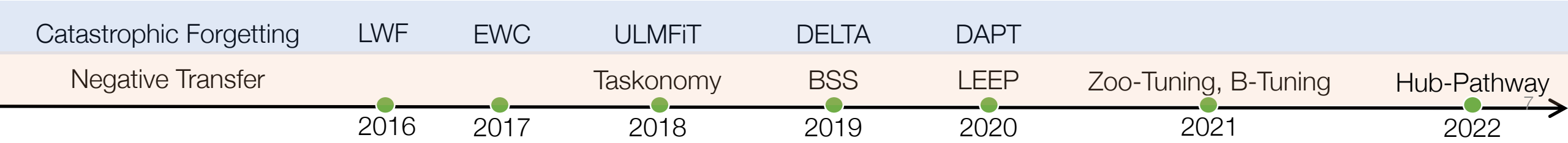



Training Strategies

- smaller lr of task-specific head. [Yosinski et al, 2014]
- lr decay helps transfer. [You et al, 2019]



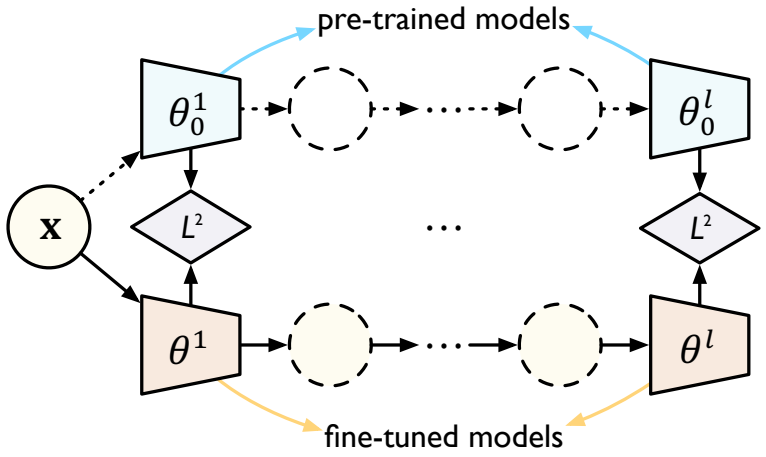



# Overcome Catastrophic Forgetting

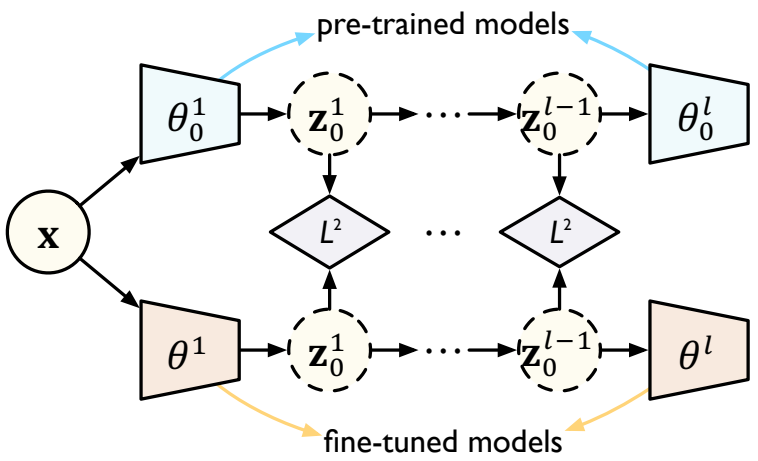
Regularization Tuning

Loss Function: $\min_\theta \sum_{i=1}^{m} L(h_\theta(x_i), y_i) + \lambda \cdot \Omega(\theta)$

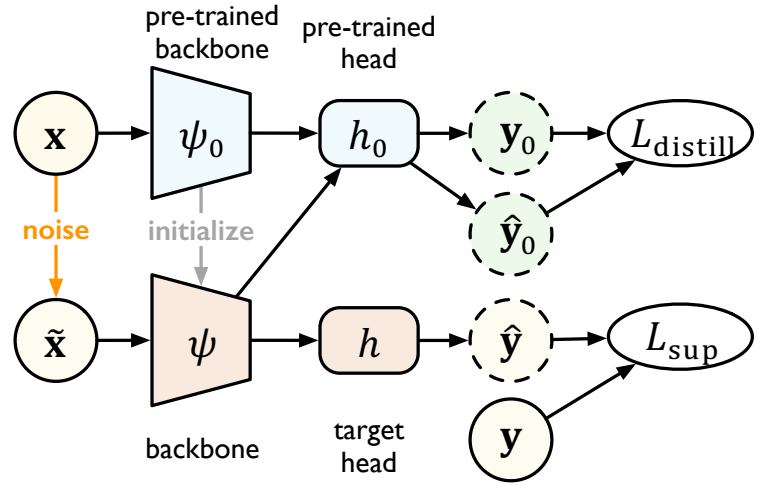
Regularization term




(a) EWC




(b) DELTA




(c) LWF

Catastrophic Forgetting LWF EWC ULMFiT DELTA DAPT

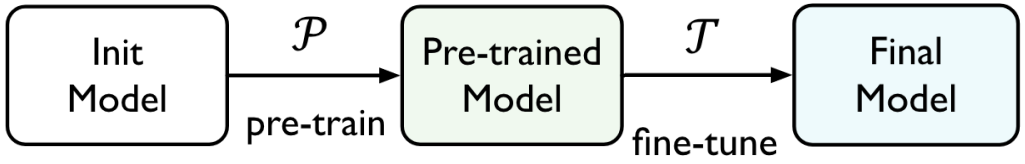
Negative Transfer Taskonomy BSS LEEP Zoo-Tuning, B-Tuning Hub-Pathway

2016 2017 2018 2019 2020 2021 2022

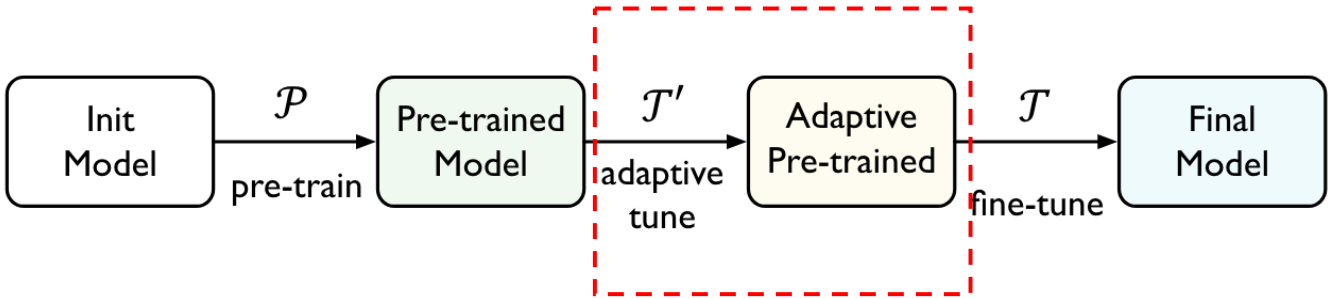
# Overcome Catastrophic Forgetting

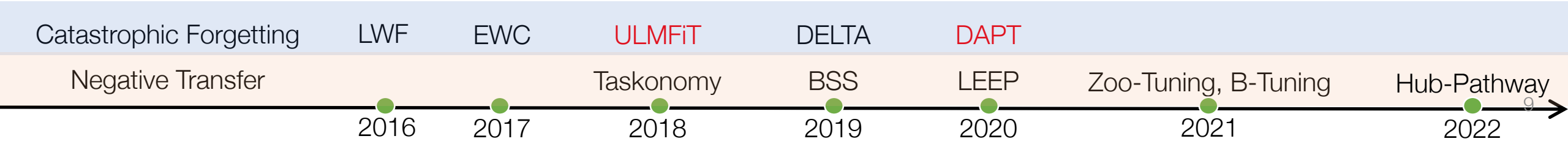
Domain Adaptive Tuning

- ULMFiT
- DAPT
- SiATL

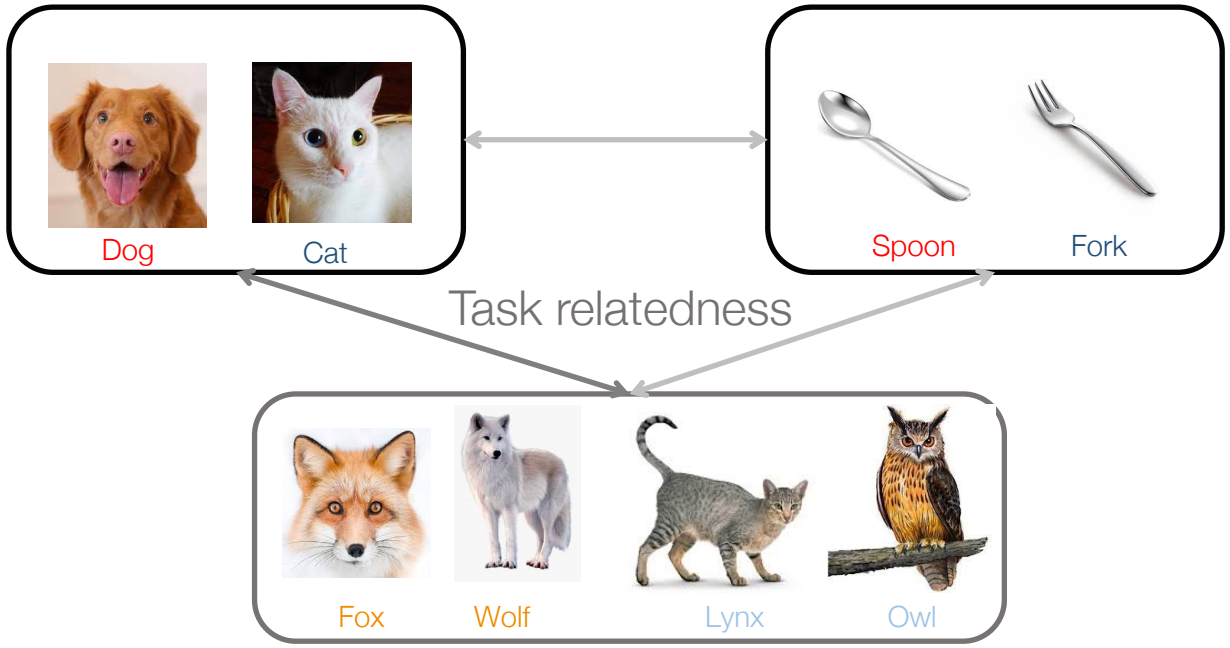


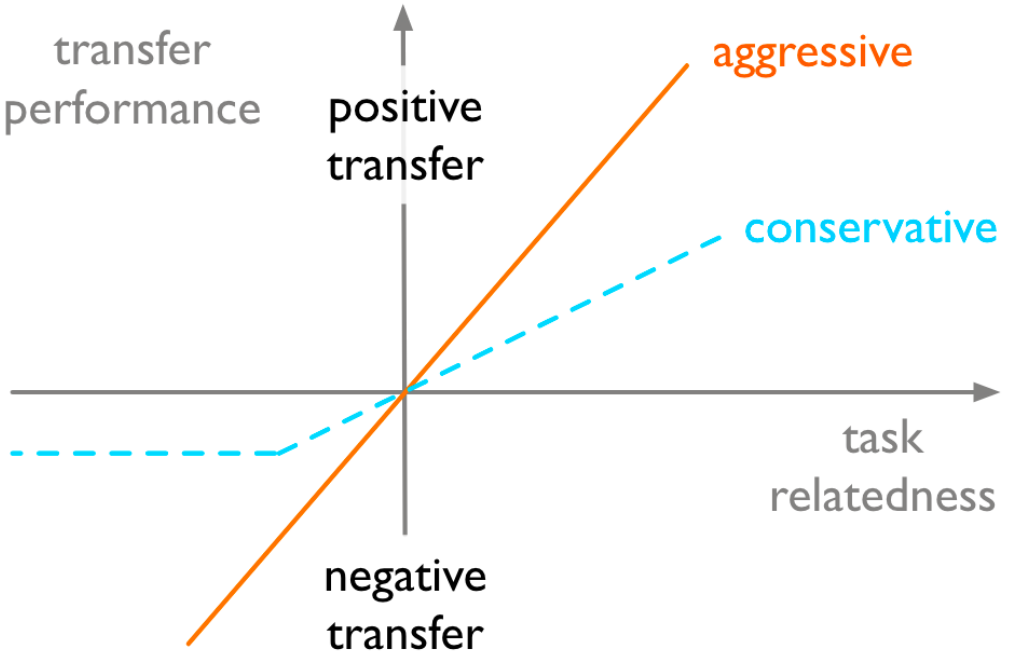

To bridge dataset shift



# Overcome Negative Transfer




Catastrophic Forgetting — LWF — EWC — ULMFiT — DELTA — DAPT

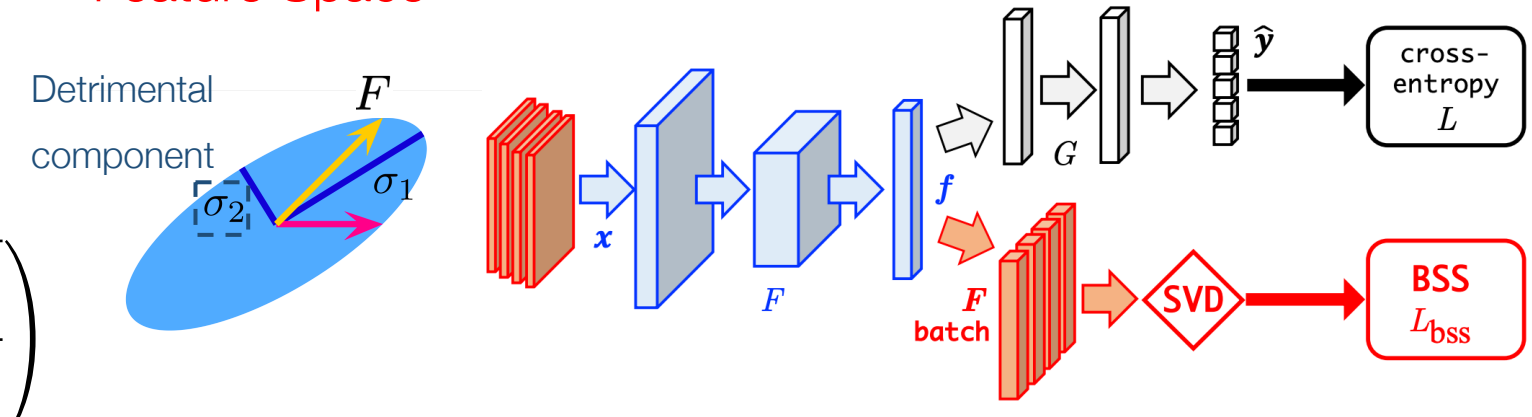
Negative Transfer — Taskonomy — BSS — LEEP — Zoo-Tuning, B-Tuning — Hub-Pathway

2016 — 2017 — 2018 — 2019 — 2020 — 2021 — 2022

# Overcome Negative Transfer

- Enhance Safe Transfer
  - BSS, Zoo-tuning

$$\mathrm{err}_P(g) \le \mathrm{err}^{\gamma}_{\hat{P}}(f) + O\left(\sqrt{\frac{p_g \log_2 r_g}{n}}\right)$$



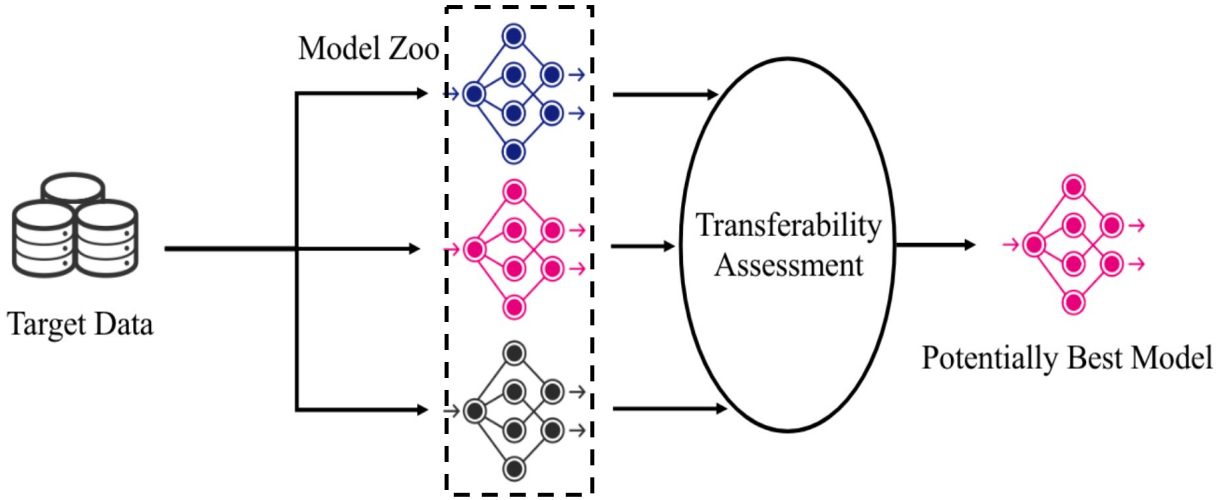

- Choose Pre-trained Models
  - LEEP, LogME

Penalize smallest singular values : $L_{\mathrm{bss}}(F) = \eta \sum_{i=1}^{k} \sigma_{-i}^2$


Xinyang Chen, Sinan Wang, Bo Fu, , Jianmin Wang, **Mingsheng Long** ✉. Catastrophic Forgetting Meets Negative Transfer: Batch Spectral Shrinkage for Safe Transfer Learning, **NeurIPS** 2019

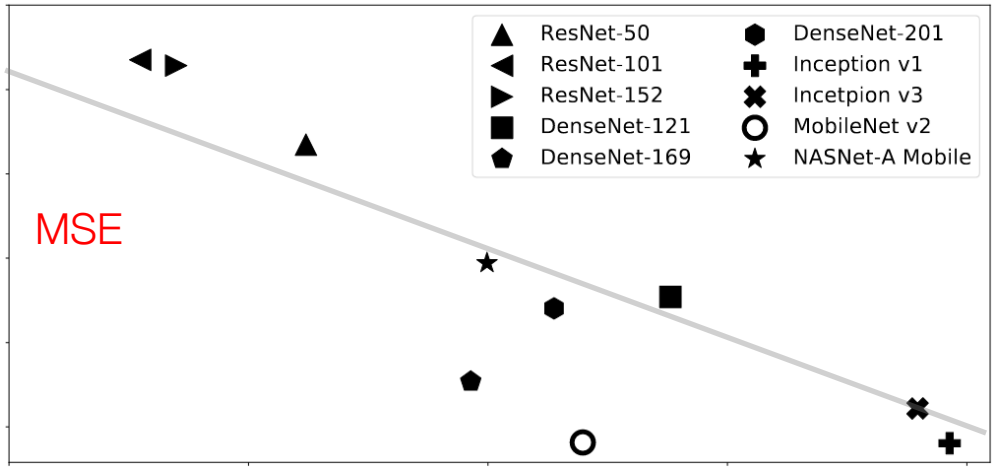
# Overcome Negative Transfer

- Enhance Safe Transfer
  - BSS, Zoo-tuning




Transferable knowledge pool

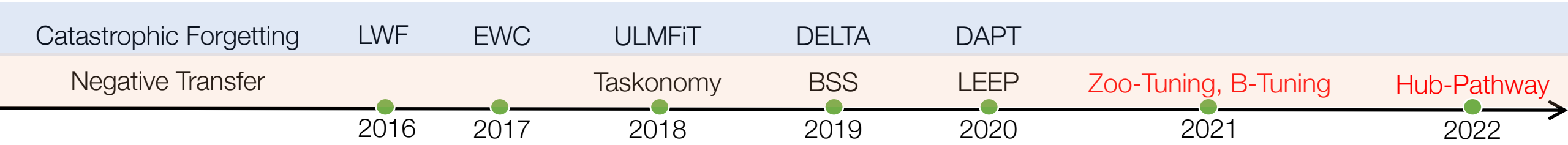
- Choose Pre-trained Models
  - LEEP, LogME




Transferability (LogME)

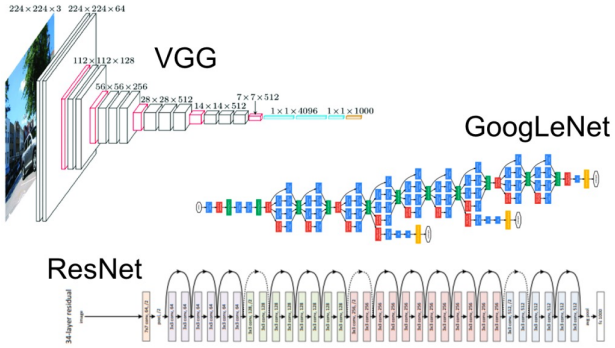
Kaichao You, Yong Liu, Jianmin Wang, **Mingsheng Long** ✉. LogME: Practical Assessment of Pre-trained Models for Transfer Learning, **ICML** 2021

# Adaptive Transfer of Model Hubs



# Pre-trained Model Hub

## Various Models and Platforms



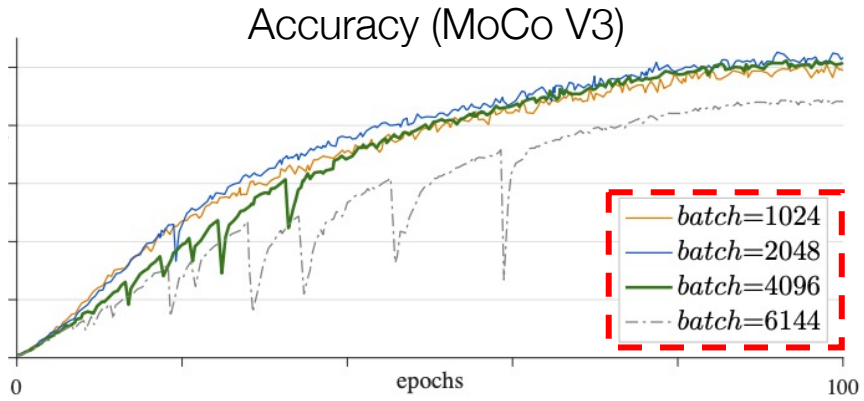

PyTorch

Hugging Face

TensorFlow Hub

Pytorch Image Models

## Avoid Heavy Pre-training




## Plenty of Transferable Knowledge

IMAGENET SUP.
MOCO PT.
MASKRCNN PT.
DEEPLAB PT.
KEYPOINT PT.

ResNet

Same architectue

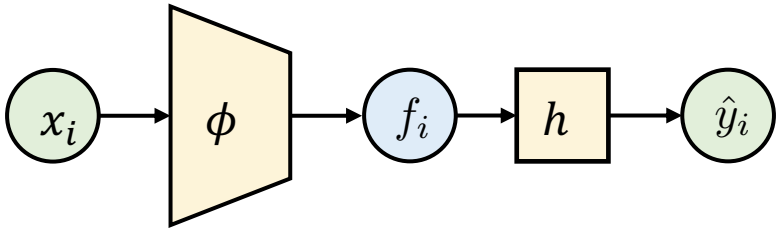
Pre-trained differently

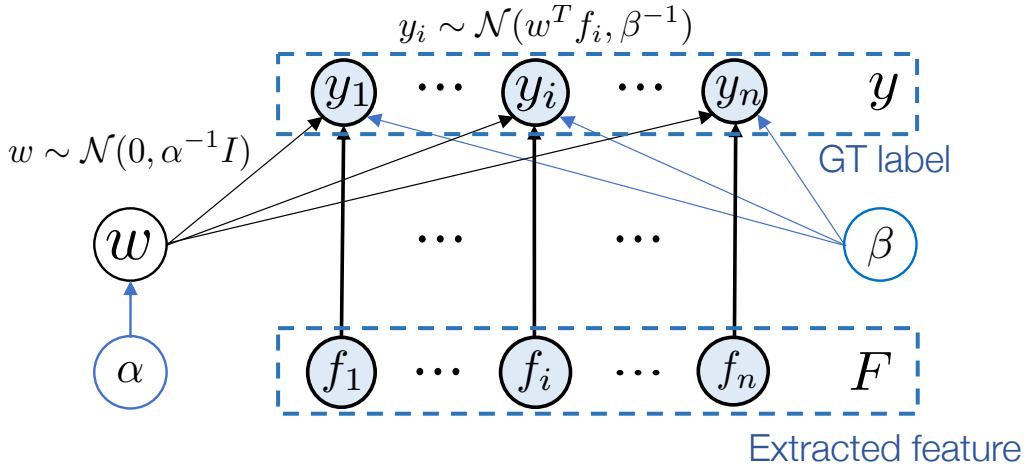
- Adapt one model
  - Which one is the best?
- Adapt multiple models
  - How to aggregate transferable knowledge?

# Transferability Assessment

Estimate adaption performance of PTM on given dataset without finetuning.

## LogME Approach



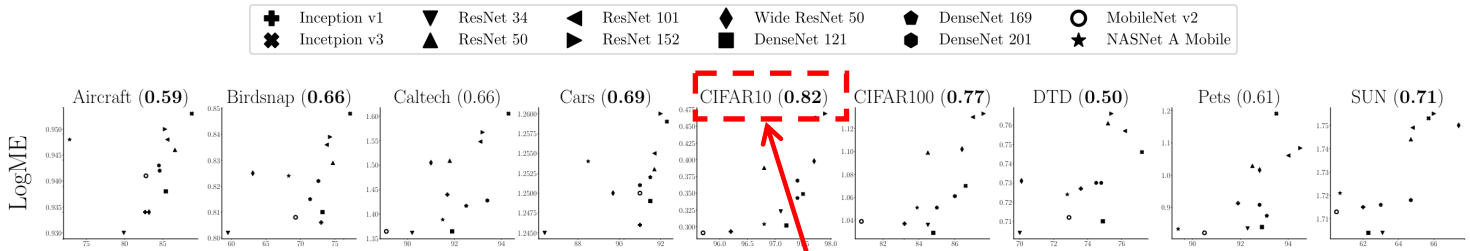




- Fixed PTM (as feature extractor).
- $P(y \mid F)$ : Graphical modeling between extracted features and GT label.
- Parameterize $P(y \mid F)$ by prior $\alpha, \beta$.
- Maximize evidence $P(y \mid F, \alpha, \beta)$.
  - MacKay algorithm with guarantee!

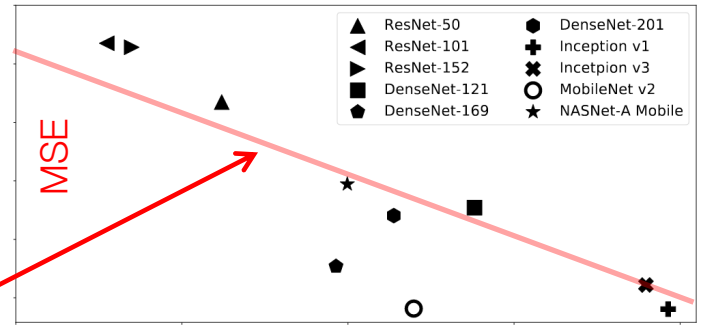
# Effectiveness of LogME

## General and Accurate

### Vision tasks

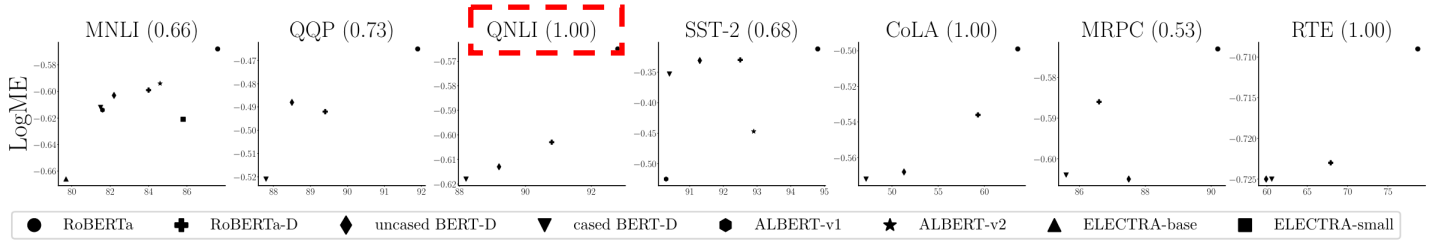



### Regression task




High correlation between LogME and finetuned performance.

### NLP tasks




### Unsupervised PTMs

| Pre-trained Network | Aircraft | | dSprites | |
|---|---|---|---|---|
| | Accuracy (%) | LogME | MSE | LogME |
| MoCo V1 | 81.68 | 0.93 | 0.069 | 1.52 |
| MoCo V2 | 84.16 | 0.94 | 0.047 | 1.64 |
| MoCo 800 | 86.99 | 0.95 | 0.050 | 1.58 |
| | $\tau_w$: 1.0 | | $\tau_w$: 1.0 | |

# Effectiveness of LogME

Computation Efficient --- MacKay algorithm with theoretical guarantee!

**Algorithm 2** Evidence Maximization by MacKay's Algorithm

**Input:** Extracted features $F \in \mathbb{R}^{n \times D}$ and corresponding labels $y \in \mathbb{R}^n$
**Output:** Logarithm of Maximum Evidence (LogME)
**Note:** $F$ has been pre-decomposed into $F = U\Sigma V^T$
Initialize $\alpha = 1, \beta = 1$
**while** $\alpha, \beta$ not converge **do**
  Compute $\gamma = \sum_{i=1}^{D} \frac{\beta\sigma_i^2}{\alpha+\beta\sigma_i^2}, \Lambda = \mathrm{diag}\{(\alpha + \beta\sigma^2)\}$
  **Naïve**: $A = \alpha I + \beta F^T F, m = \beta A^{-1} F^T y$
  Update $\alpha \leftarrow \frac{\gamma}{m^T m}, \beta \leftarrow \frac{n-\gamma}{||Fm-y||_2^2}$
**end while**
Compute and return $\mathcal{L} = \frac{1}{n}\mathcal{L}(\alpha, \beta)$ using Equation 2

$$\mathcal{O}(nCD^2 + CD^3)$$

Biquadrate complexity

**Algorithm 2** Evidence Maximization by MacKay's Algorithm

**Input:** Extracted features $F \in \mathbb{R}^{n \times D}$ and corresponding labels $y \in \mathbb{R}^n$
**Output:** Logarithm of Maximum Evidence (LogME)
**Note:** $F$ has been pre-decomposed into $F = U\Sigma V^T$
Initialize $\alpha = 1, \beta = 1$
**while** $\alpha, \beta$ not converge **do**
  Compute $\gamma = \sum_{i=1}^{D} \frac{\beta\sigma_i^2}{\alpha+\beta\sigma_i^2}, \Lambda = \mathrm{diag}\{(\alpha + \beta\sigma^2)\}$
  **Optimized** by You et al. (2021): $m = \beta(V(\Lambda^{-1}(V^T(F^T y))))$
  Update $\alpha \leftarrow \frac{\gamma}{m^T m}, \beta \leftarrow \frac{n-\gamma}{||Fm-y||_2^2}$
**end while**
Compute and return $\mathcal{L} = \frac{1}{n}\mathcal{L}(\alpha, \beta)$ using Equation 2

$$\mathcal{O}(nD^2 + nCD + CD^2 + D^3)$$

Cubic complexity

**Algorithm 3** Evidence Maximization by Optimized Fixed Point Iteration

**Input:** Extracted features $F \in \mathbb{R}^{n \times D}$ and corresponding labels $y \in \mathbb{R}^n$
**Output:** Logarithm of Maximum Evidence (LogME)
**Require:** Truncated SVD of $F$: $F = U_r \Sigma_r V_r^T$, with $U_r \in \mathbb{R}^{n \times r}, \Sigma_r \in \mathbb{R}^{r \times r}, V_r \in \mathbb{R}^{D \times r}$.
**Compute** the first $r$ entries of $z = U_r^T y$
**Compute** the sum of remaining entries $\Delta = \sum_{i=r+1}^{n} z_i^2 = \sum_{i=1}^{n} y_i^2 - \sum_{i=1}^{r} z_i^2$
Initialize $\alpha = 1, \beta = 1, t = \frac{\alpha}{\beta} = 1$
**while** $t$ not converge **do**
  Compute $m^T m = \sum_{i=1}^{r} \frac{\sigma_i^2 z_i^2}{(t+\sigma_i^2)^2}$, $\gamma = \sum_{i=1}^{r} \frac{\sigma_i^2}{t+\sigma_i^2}$, $||Fm-y||_2^2 = \sum_{i=1}^{r} \frac{z_i^2}{(1+\sigma_i^2/t)^2} + \Delta$
  Update $\alpha \leftarrow \frac{\gamma}{m^T m}, \beta \leftarrow \frac{n-\gamma}{||Fm-y||_2^2}, t = \frac{\alpha}{\beta}$
**end while**
Compute $m = V_r \Sigma' z$, where $\Sigma'_{ii} = \frac{\sigma_i}{t+\sigma_i^2} (1 \le i \le r)$.
Compute and return $\mathcal{L} = \frac{1}{n}\mathcal{L}(\alpha, \beta)$ using Equation 2

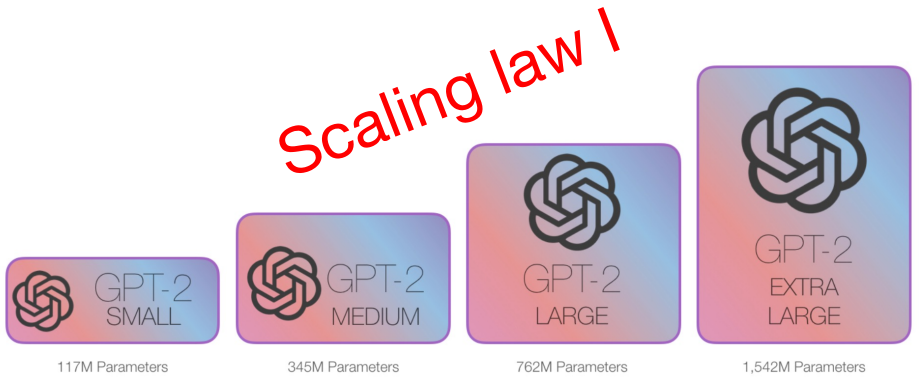
$$\mathcal{O}(nD^2 + nCD)$$

Cubic complexity with fewer terms

| | wall-clock time | | memory footprint | |
|---|---|---|---|---|
| Computer Vision | fine-tune (upper bound) | 161000s | fine-tune (upper bound) | 6.3 GB |
| | extract feature (lower bound) | 37s | extract feature (lower bound) | 43 MB |
| | LogME | 43s | LogME | 53 MB |
| | benefit | 3700 ↑ | benefit | 120 ↑ |
| Natural Language Processing | fine-tune (upper bound) | 100200s | fine-tune (upper bound) | 88 GB |
| | extract feature (lower bound) | 1130s | extract feature (lower bound) | 1.2 GB |
| | LogME | 1136s | LogME | 1.2 GB |
| | benefit | 88 ↑ | benefit | 73 ↑ |

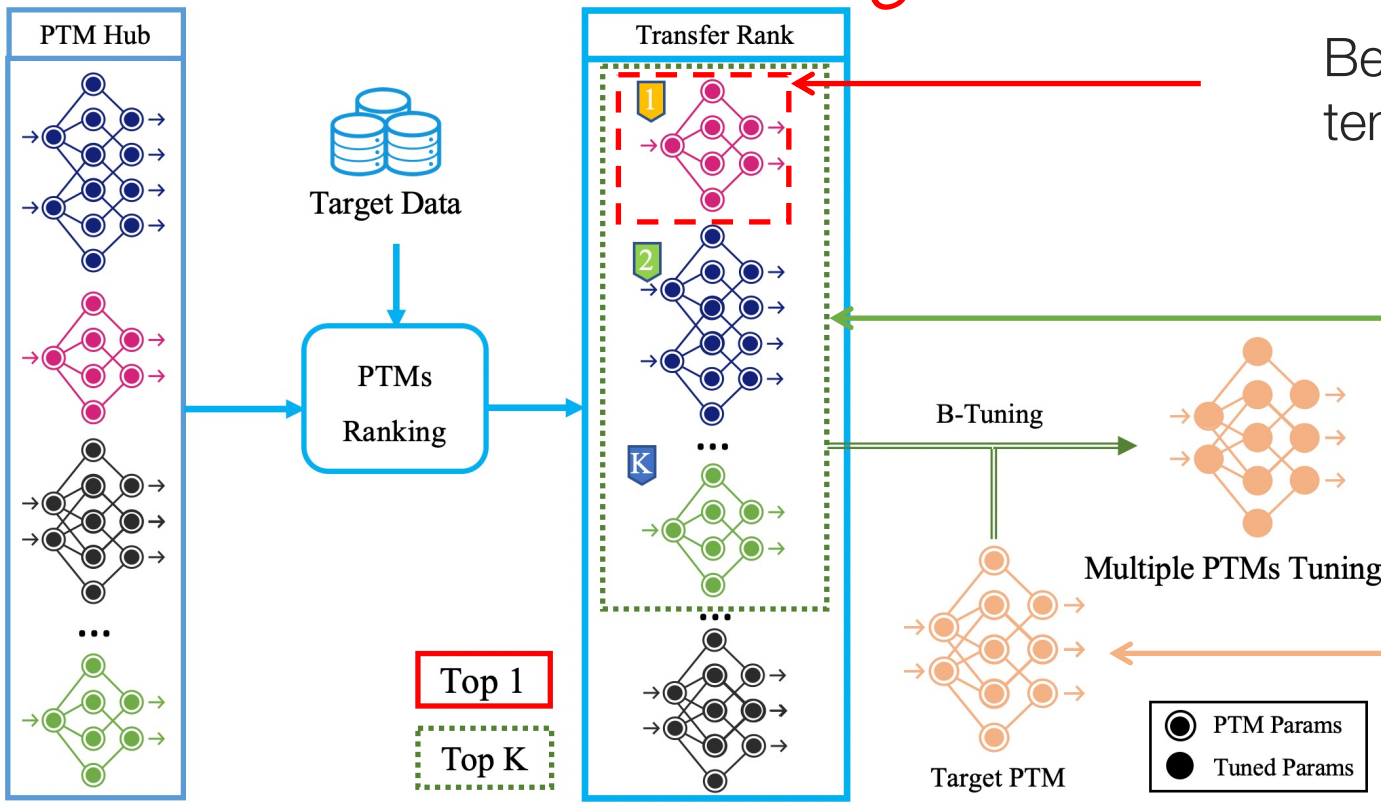
# Tuning Pre-trained Models

Why adapt multiple models?




Scaling law II

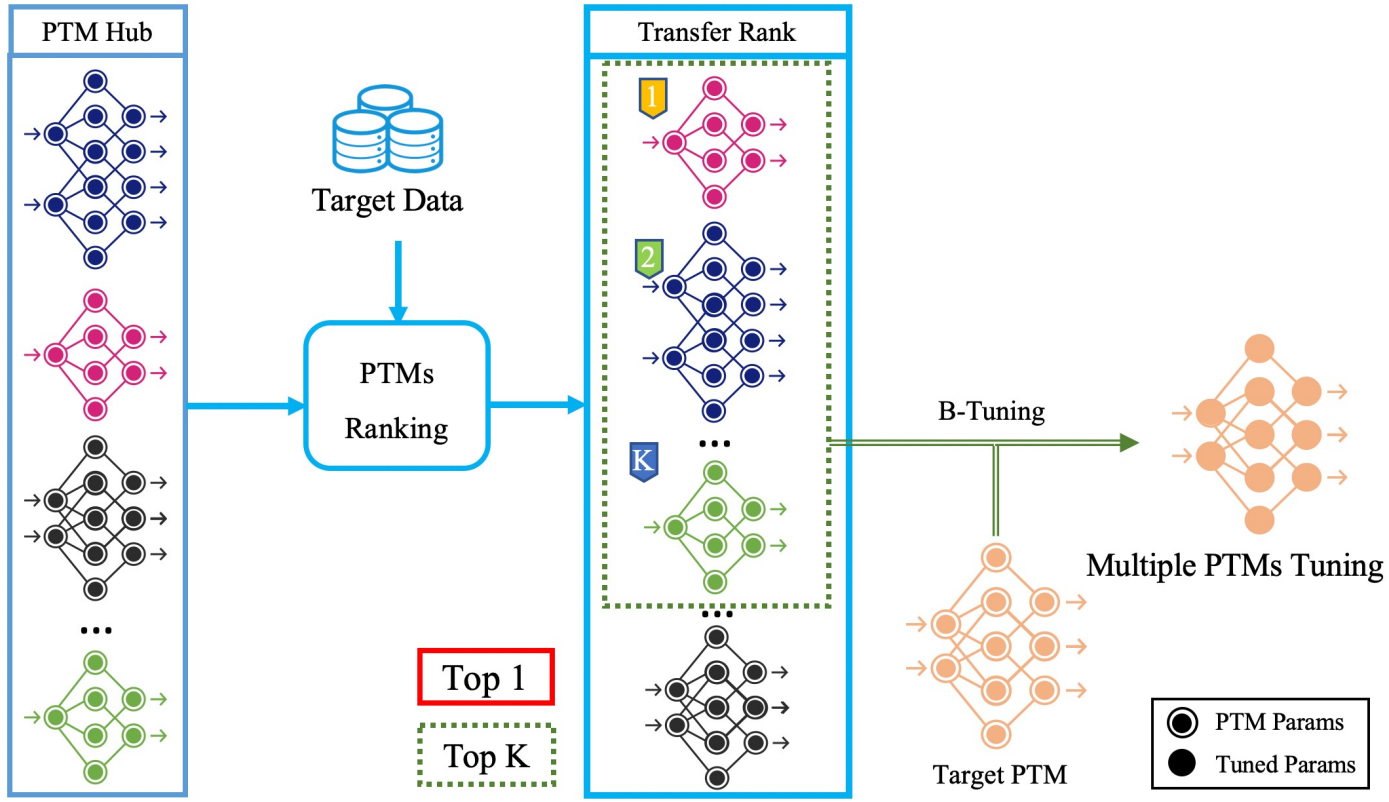
Best ranked models (e.g. DenseNet-201) tend to be large and expensive to deploy.




Transfer from multiple pre-trained models can be more beneficial.

Given a desired target architecture in industrial requirement.

# Ranking and Tuning Paradigm


- Ranking
  - LEEP, NCE, LogME…
- Selection
  - Top-K: Heuristic but Effective
- Tuning
  - Architecture heterogeneity
  - Dimensionality of features
  - Always challenging part…

# B-Tuning

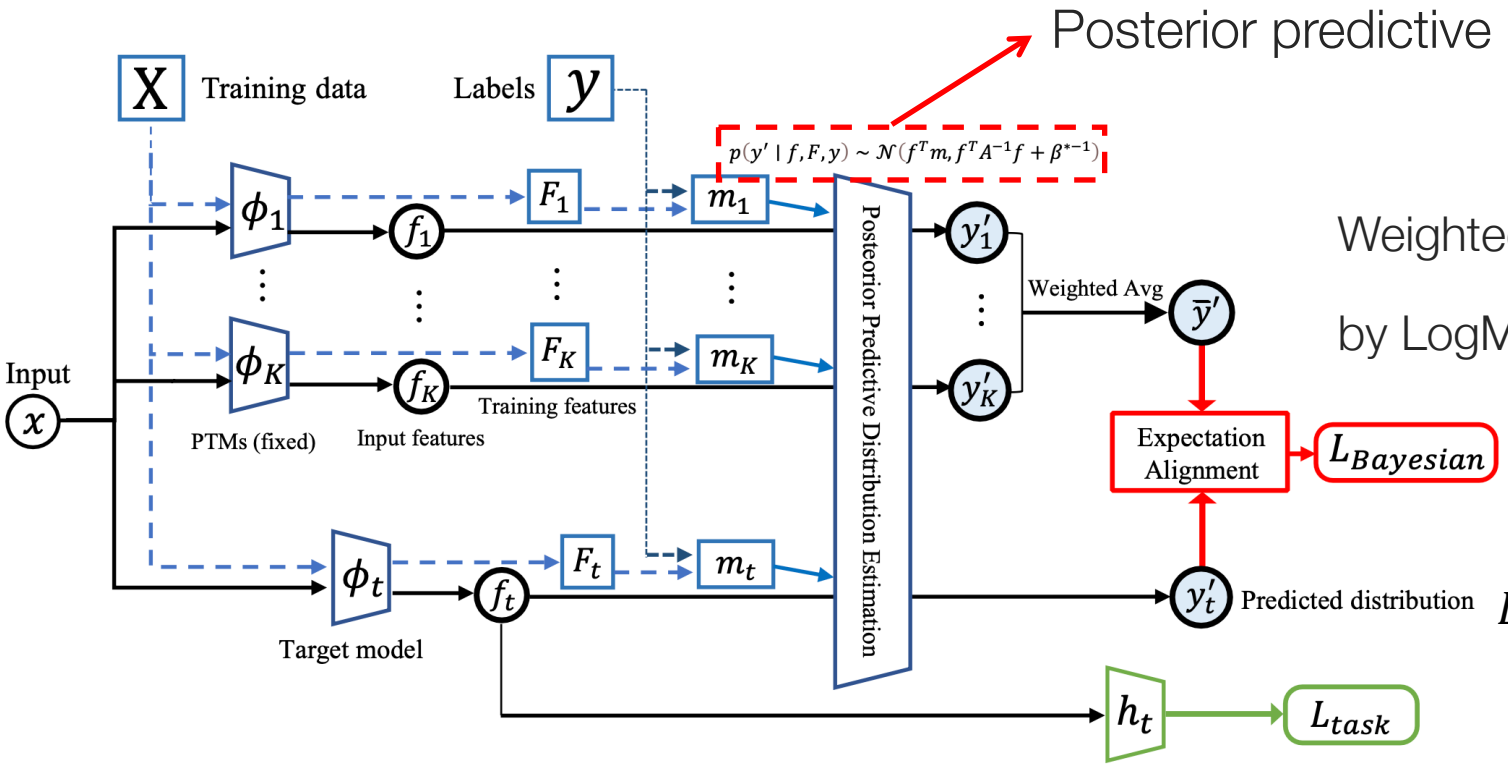
Consider simple Knowledge Distillation (KD):

$$L_{\mathrm{KD}} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{K}\sum_{k=1}^{K}|\phi_k(x_i) - W_k\phi_t(x_i)|_2^2$$

- Needs additional learnable projection $W_k$ for each teacher model.
- Treats all teacher models as equal:
  - No adaptive mechanism to transfer only useful knowledge.

Kaichao You, Yong Liu, Jianmin Wang, Michael I Jordan, **Mingsheng Long** ✉. Ranking and Tuning Pre-trained Models: A New Paradigm of Exploiting Model Hubs, **JMLR** 2022

# B-Tuning




Weighted average: $\bar{y}' = \sum_{k=1}^{K} \pi_k y'_k$

by LogME score: $\pi_k = \frac{\exp(\mathcal{L}_k / t)}{\sum_{j=1}^{K} \exp(\mathcal{L}_j / t)}$

$$L_{Bayesian} = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{C} \sum_{c=1}^{C} \left| \sum_{k=1}^{K} \pi_k f_k^T m_{k,c} - f_t^T m_{t,c} \right|_2^2$$

- Project teacher features into a common output space by LogME.
- Transfer them to target model with weighting from their LogME score.

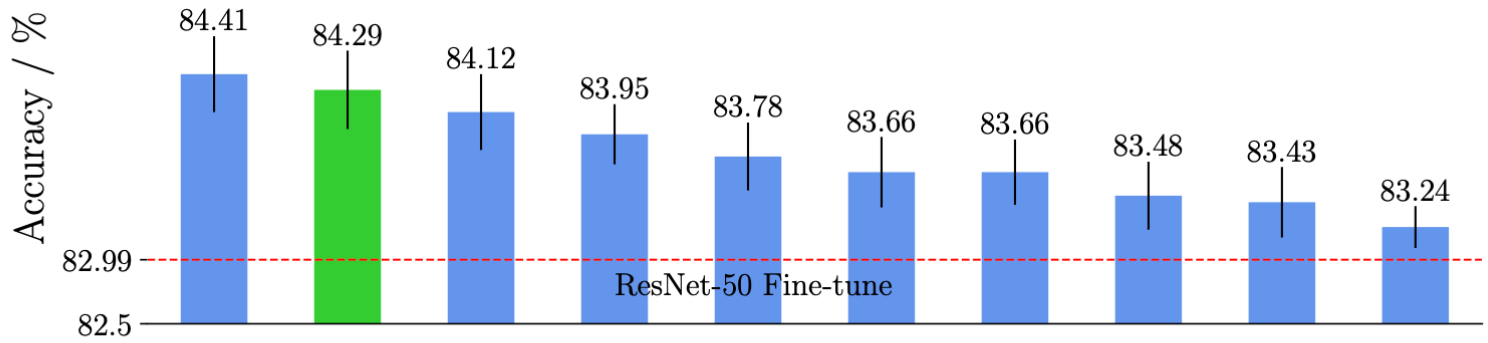
**Intuition:** encourage the target model to behave like the best top-K teachers.

Kaichao You, Yong Liu, Jianmin Wang, Michael I Jordan, **Mingsheng Long** ✉. Ranking and Tuning Pre-trained Models: A New Paradigm of Exploiting Model Hubs, JMLR 2022

# Experiments

Reduced burden of Selection and Adaptation.

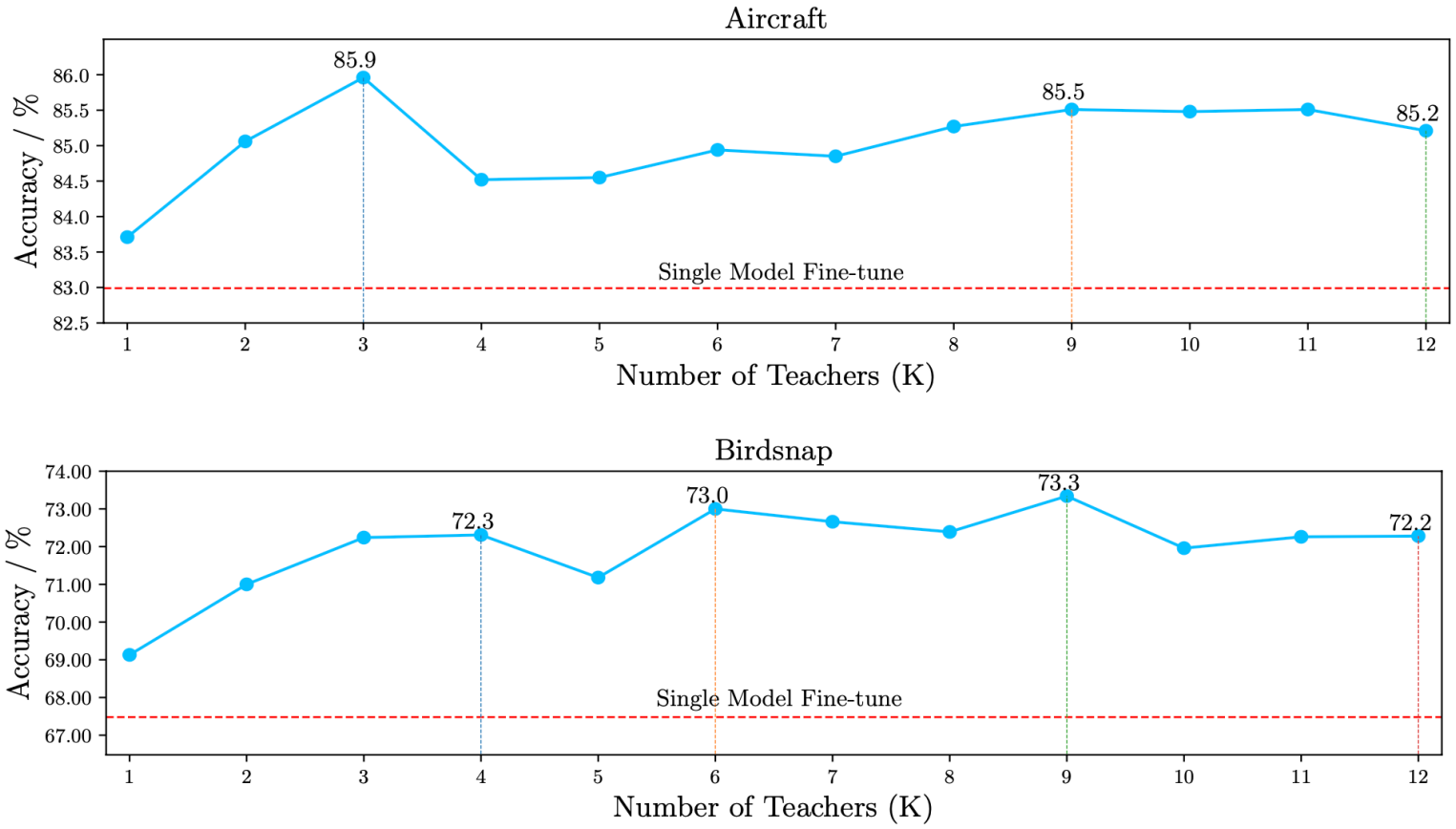
- Exhaustively fine-tune 10 times: 84.41% accuracy.
- Rank by LogME and fine-tune once: 84.29% accuracy.




| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| ImageNet Sup. | 0.947 | ✓ | ✓ | ✓ | | ✓ | ✓ | | | | ✓ |
| MaskRCNN PT. | 0.936 | ✓ | ✓ | | ✓ | | ✓ | | ✓ | ✓ | |
| MoCo PT. | 0.934 | | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | | |
| KeyPoint PT. | 0.914 | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| DeepLab PT. | 0.913 | ✓ | | ✓ | ✓ | | | ✓ | | ✓ | ✓ |
| PTM name | LogME | | | | | | | | | | |

# Experiments

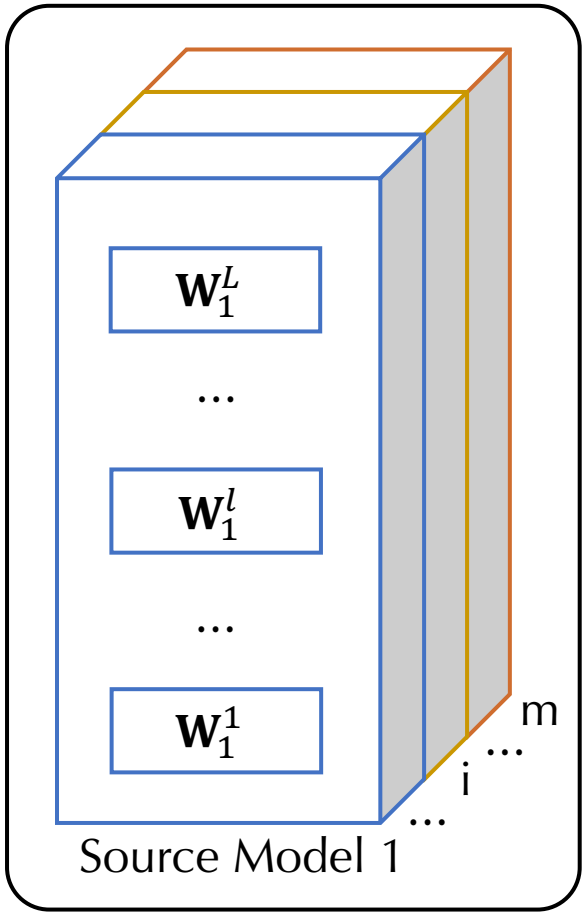
Fully utilization of transferable knowledge in Model Hub.

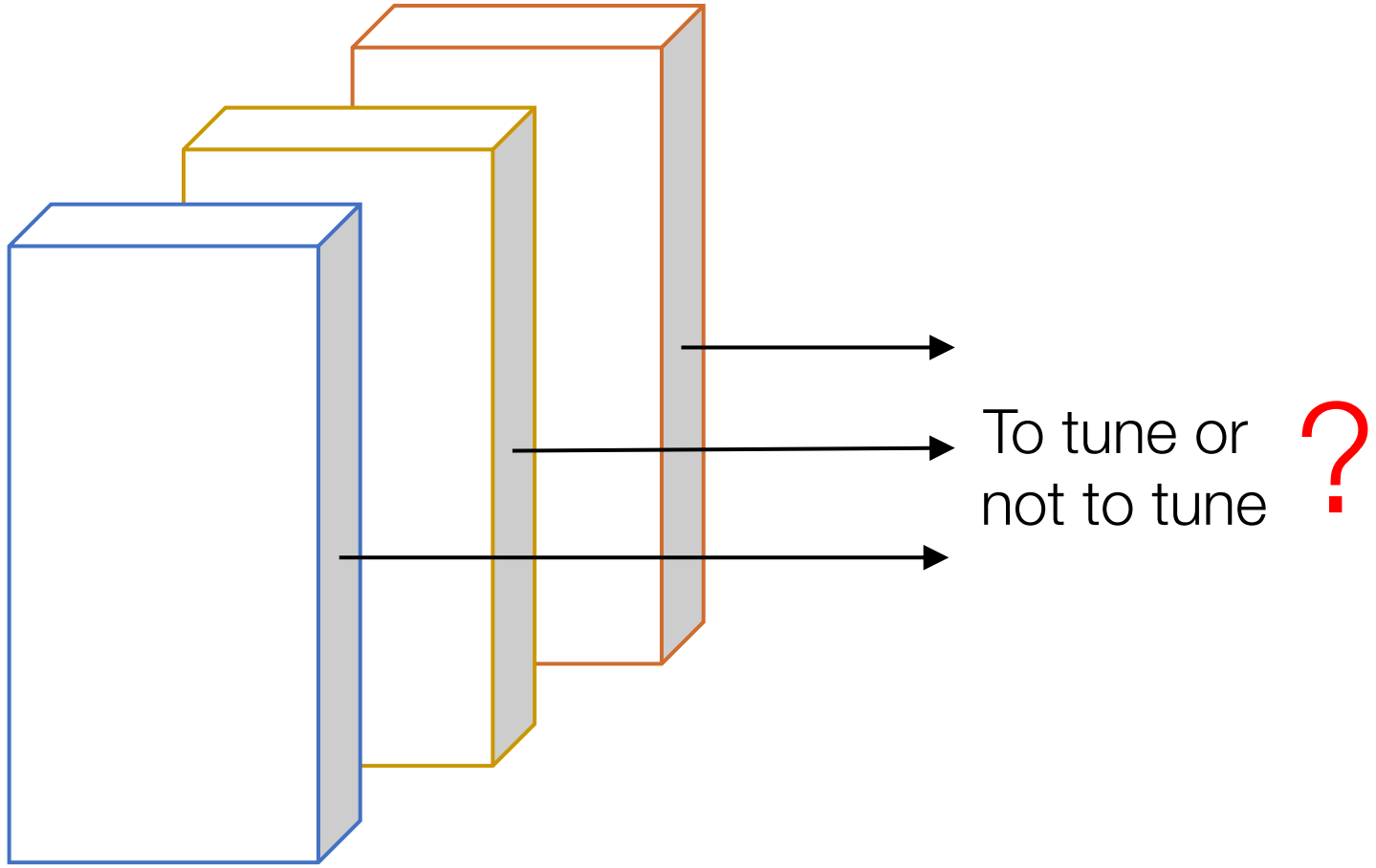



- Just fine-tune the most popular model is sub-optimal.
- The ranking and B-Tuning paradigm brings 3%~5% accuracy gain.

# Adaption from Model Zoo

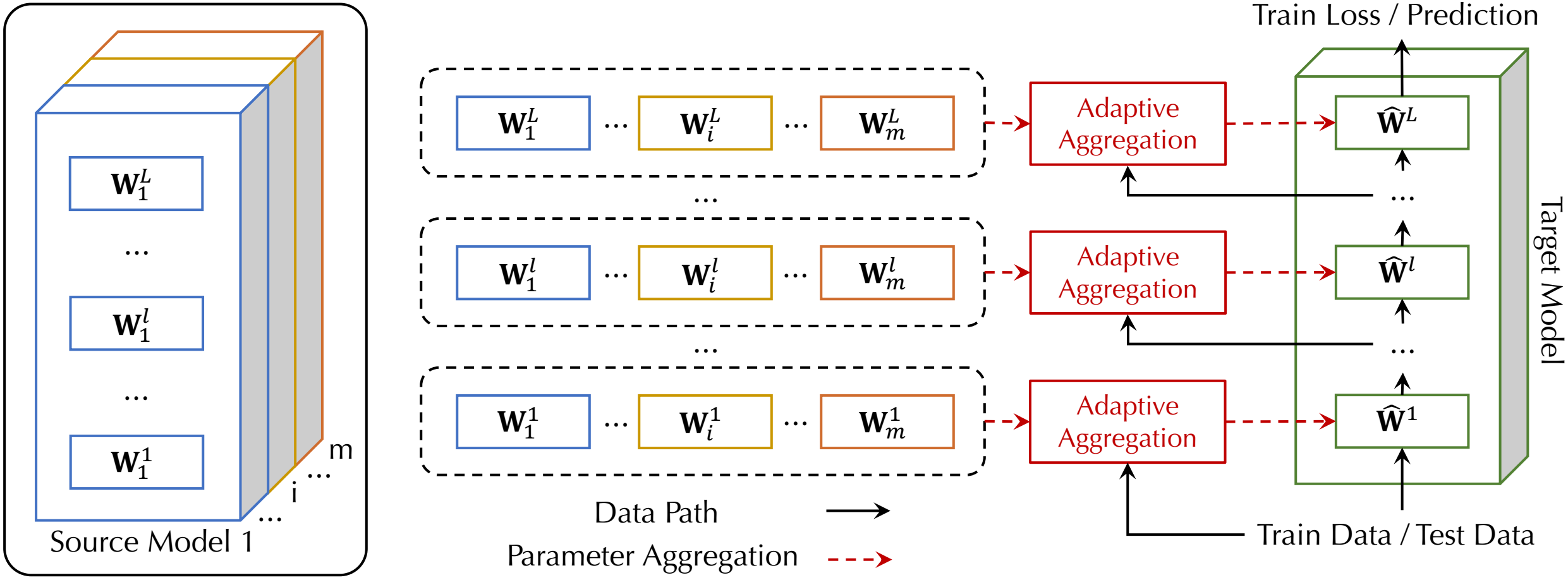
Considering models with same architecture but different knowledge.



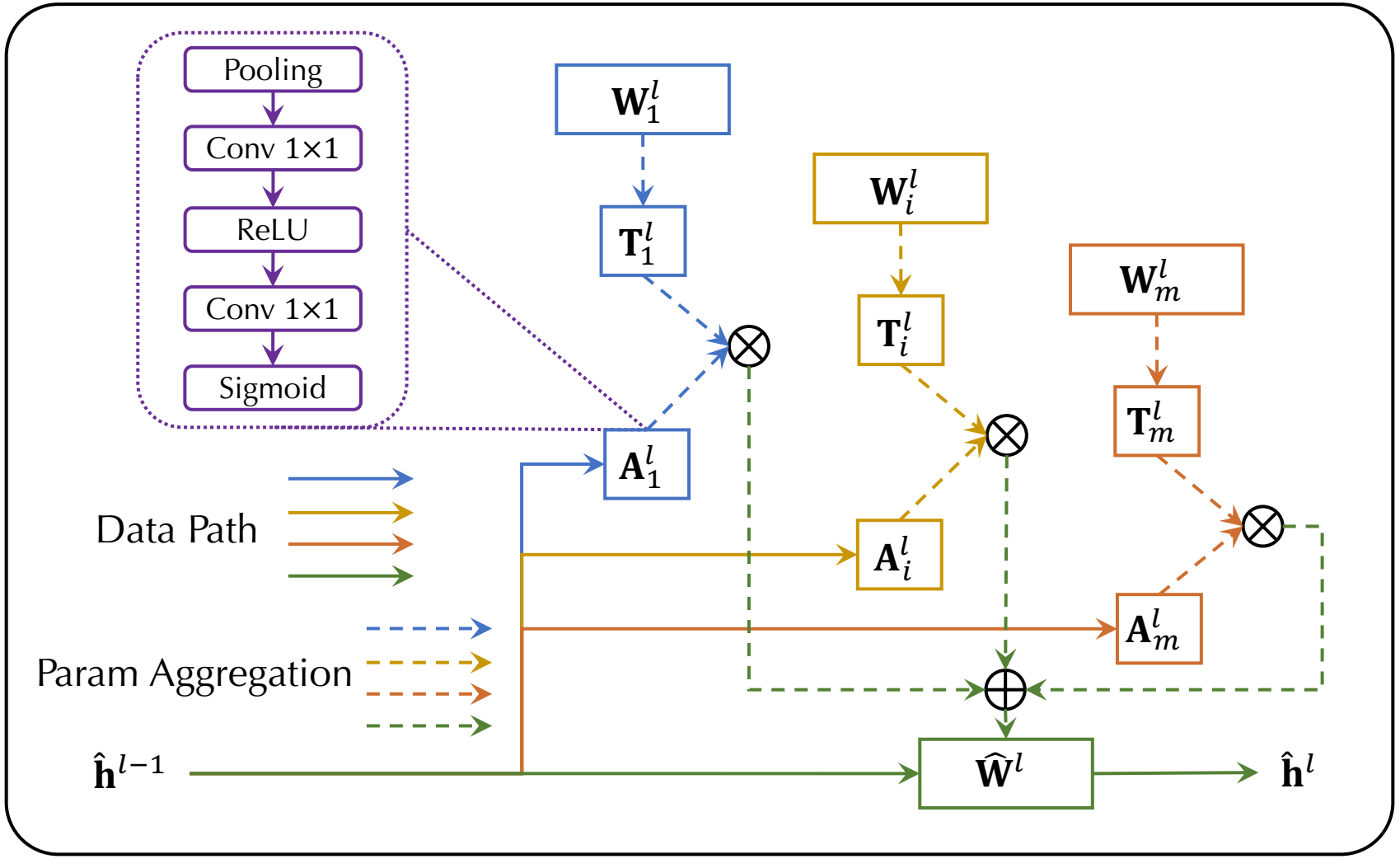



# Zoo-Tuning

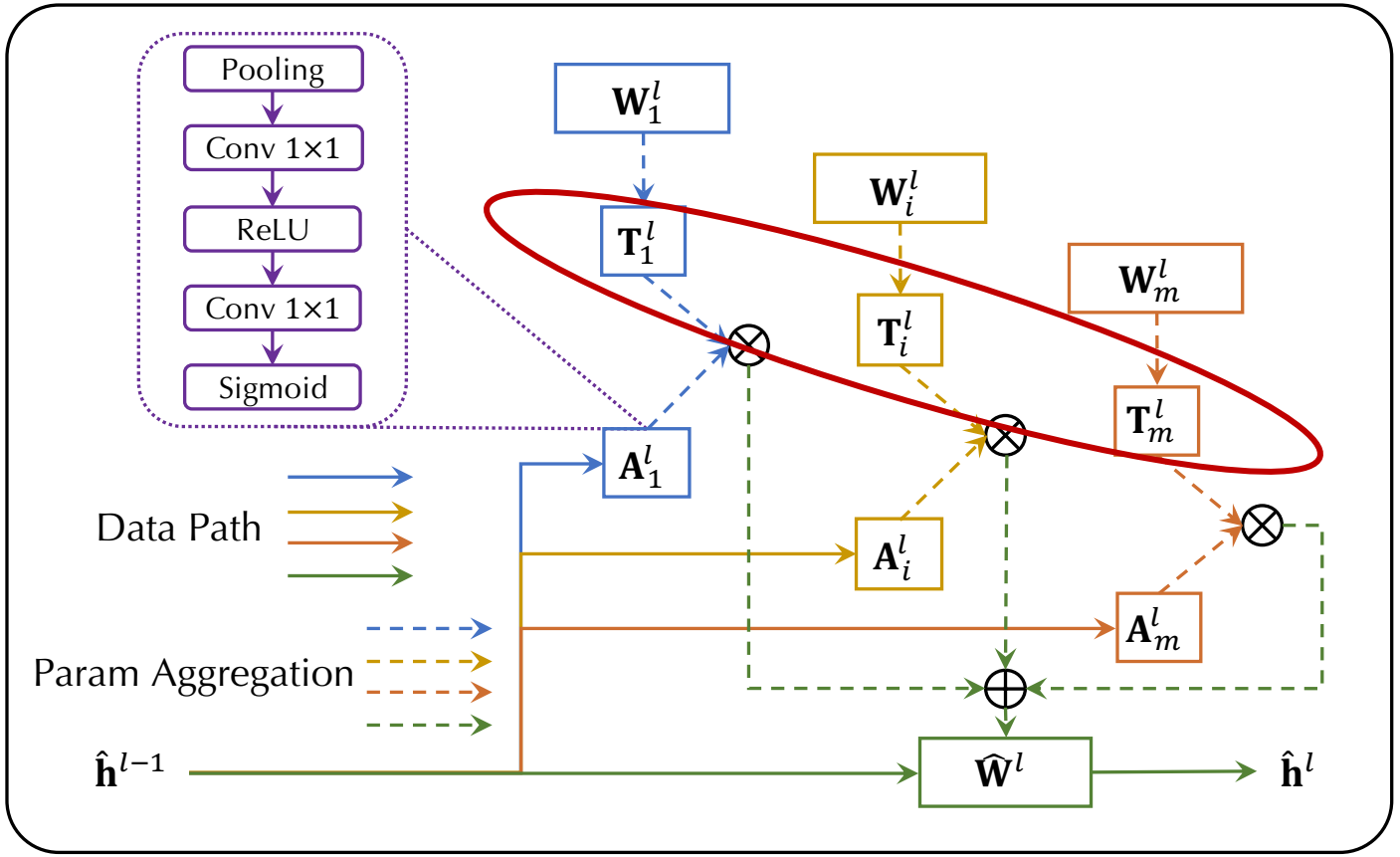
Adaptively aggregate source model parameters to derive target model.



# Adaptive Aggregation



# Adaptive Aggregation



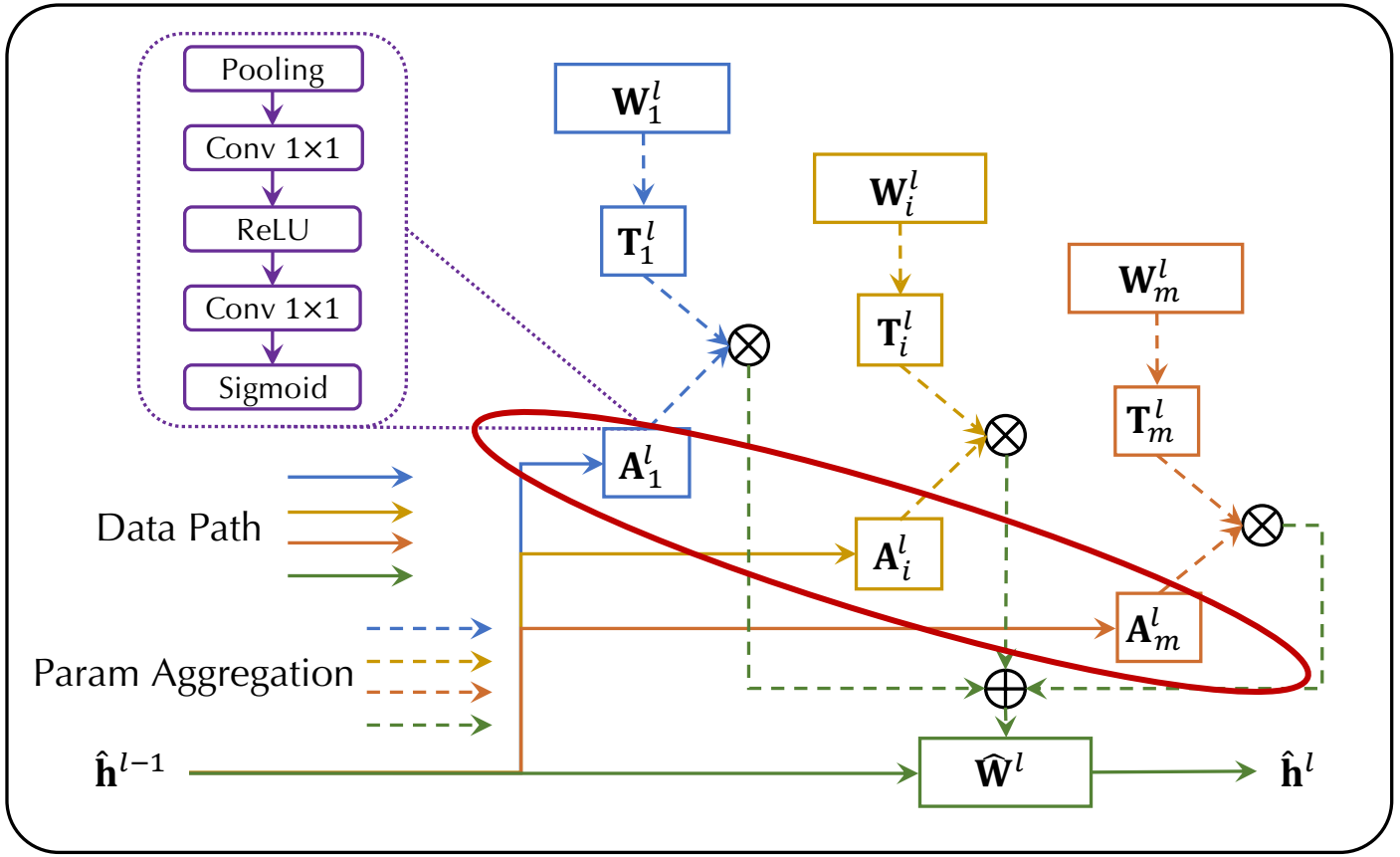

## Channel alignment

- Channels in different pre-trained models may have different semantic meanings.

$$\widetilde{\mathbf{W}}_i^l = \mathbf{T}_i^l * \mathbf{W}_i^l$$

Transformed parameter

# Adaptive Aggregation




## Data-dependent gating

- Each datum of each task has a different aggregation.

$$\widehat{\mathbf{W}}^l = \sum_{i=1}^{m} a_i^l \widetilde{\mathbf{W}}_i^l = \sum_{i=1}^{m} \mathbf{A}_i^l(\hat{h}^{l-1}) \left(\mathbf{T}_i^l * \mathbf{W}_i^l\right)$$

Target parameter

Gating value

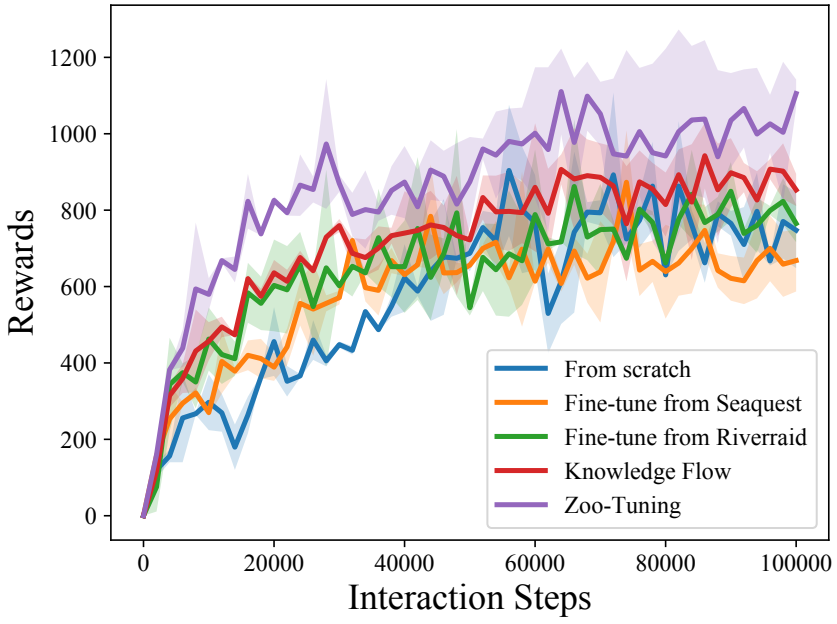
# Experiments

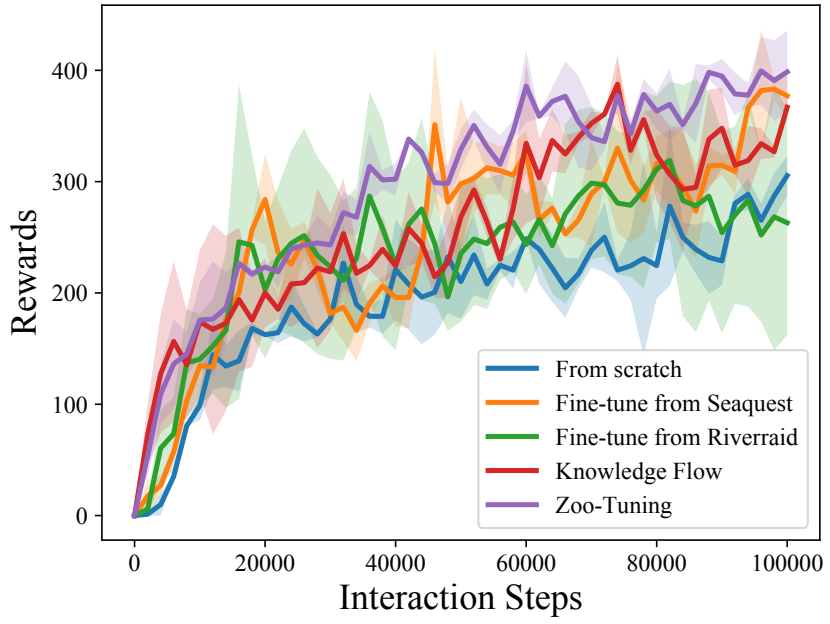
- Adaptive transfer from multiple models → Better accuracy.
- Adaptive aggregation of model parameters → More efficient than ensemble.

| Model | General | | Fine-grained | | | Specialized | | | Train | | Inference | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CIFAR-100 | COCO-70 | AIRCRAFT | CARS | INDOORS | DMLAB | EUROSAT | AVG. ACC | GFLOPS | PARAMS | GFLOPS | PARAMS |
| IMAGENET SUP. | 81.18 | 81.97 | 84.63 | 89.38 | 73.69 | 74.57 | 98.43 | 83.41 | 4.12 | 23.71M | 4.12 | 23.71M |
| MOCO PT. | 75.31 | 75.66 | 83.44 | 85.38 | 70.98 | 75.06 | 98.82 | 80.66 | 4.12 | 23.71M | 4.12 | 23.71M |
| MASKRCNN PT. | 79.12 | 81.64 | 84.76 | 87.12 | 73.01 | 74.73 | 98.65 | 82.72 | 4.12 | 23.71M | 4.12 | 23.71M |
| DEEPLAB PT. | 78.76 | 80.70 | 84.97 | 88.03 | 73.09 | 74.34 | 98.54 | 82.63 | 4.12 | 23.71M | 4.12 | 23.71M |
| KEYPOINT PT. | 76.38 | 76.53 | 84.43 | 86.52 | 71.35 | 74.58 | 98.34 | 81.16 | 4.12 | 23.71M | 4.12 | 23.71M |
| ENSEMBLE | 82.26 | 82.81 | **87.02** | **91.06** | 73.46 | **76.01** | 98.88 | 84.50 | 20.60 | 118.55M | 20.60 | 118.55M |
| DISTILL | 82.32 | 82.44 | 85.00 | 89.47 | 73.97 | 74.57 | 98.95 | 83.82 | 24.72 | 142.28M | 4.12 | 23.71M |
| KNOWLEDGE FLOW | 81.56 | 81.91 | 85.27 | 89.22 | 73.37 | 75.55 | 97.99 | 83.55 | 28.83 | 169.11M | 4.12 | 23.71M |
| LITE ZOO-TUNING | 83.39 | 83.50 | 85.51 | 89.73 | 75.12 | 75.22 | **99.12** | 84.51 | 4.53 | 130.43M | 4.12 | 23.71M |
| ZOO-TUNING | **83.77** | **84.91** | 86.54 | 90.76 | **75.39** | 75.64 | **99.12** | **85.16** | 4.53 | 130.43M | 4.18 | 122.54M |

# Applied to RL tasks

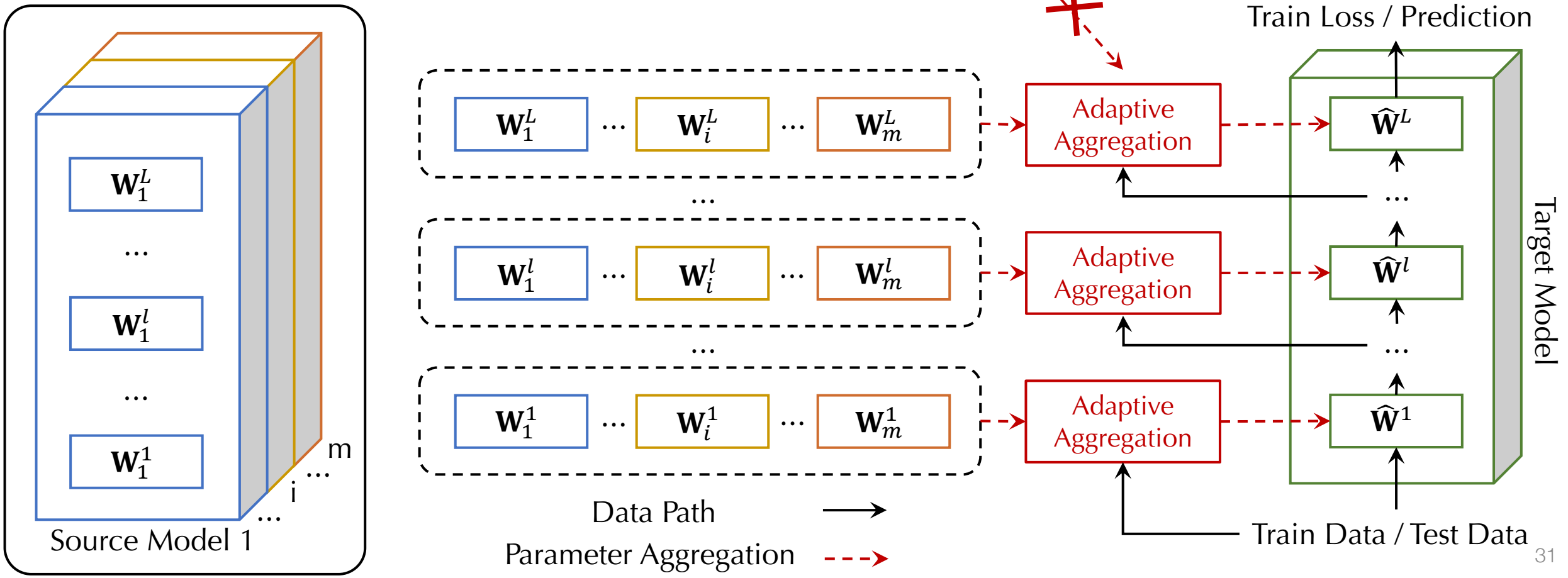
- Reinforcement Learning: Atari Games.
- Pre-trained Models: Models trained from other games.

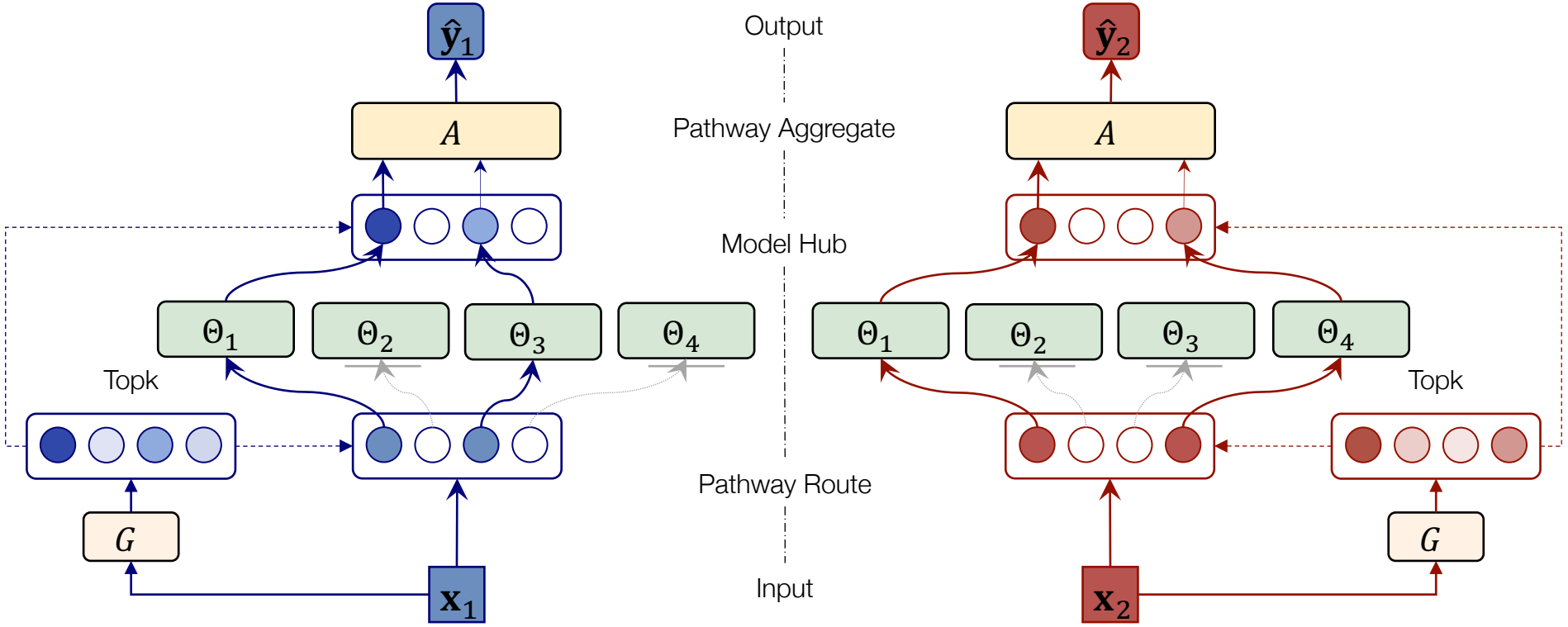
# Towards A More General Framework

- Parameter aggregation: efficient and adaptive.
- Not general enough for diverse models or even heterogeneous architectures.



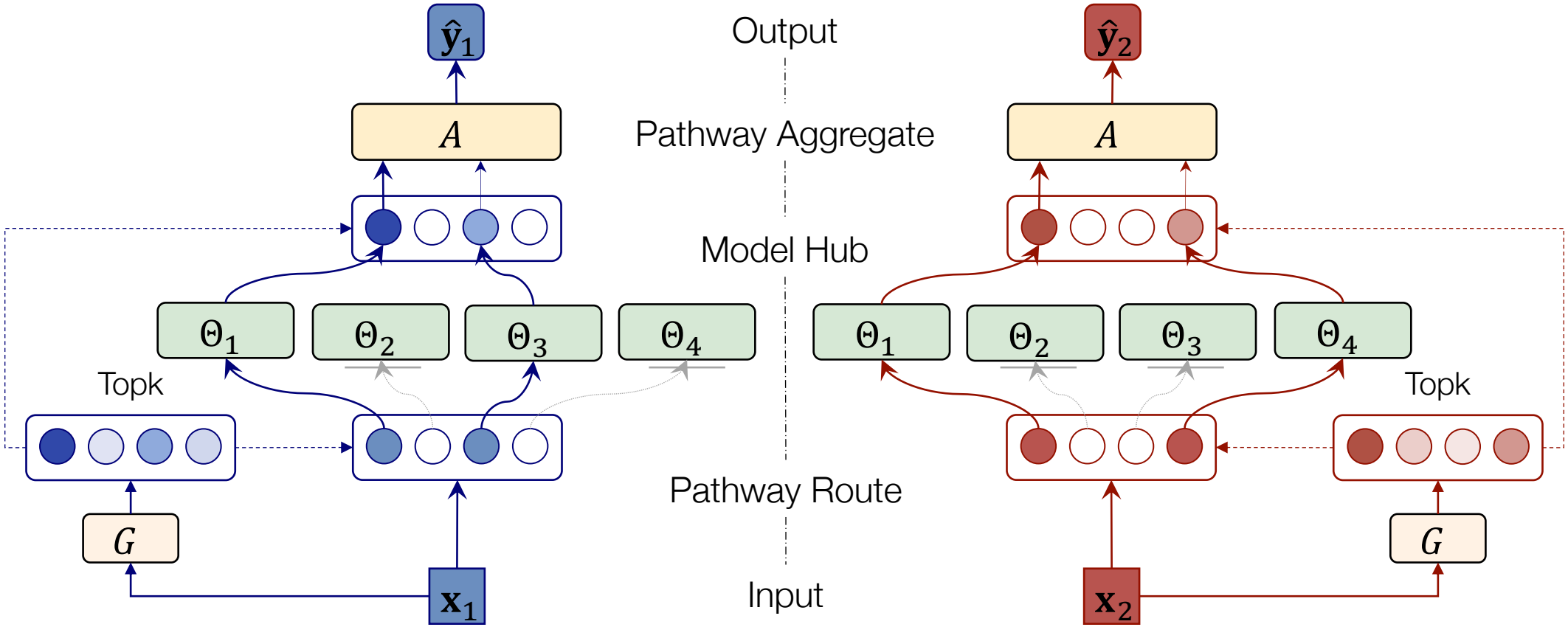
# Hub-Pathway Framework

Design data-dependent pathways throughout the Model Hub.

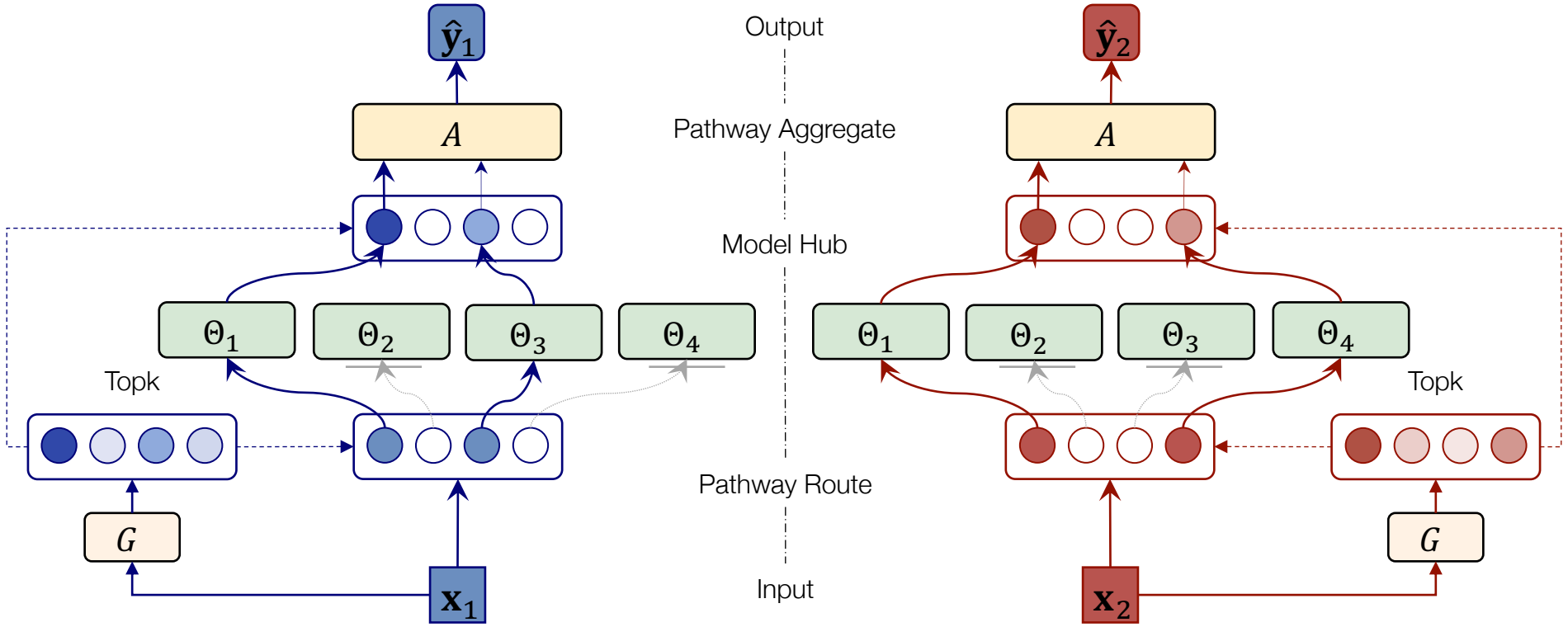


# Hub-Pathway Framework

- **Input level:** route different data to different models.
- **Output level:** aggregate transferred knowledge to make predictions.
- **Pathway flow:** control training and inference costs with Top-K activation.



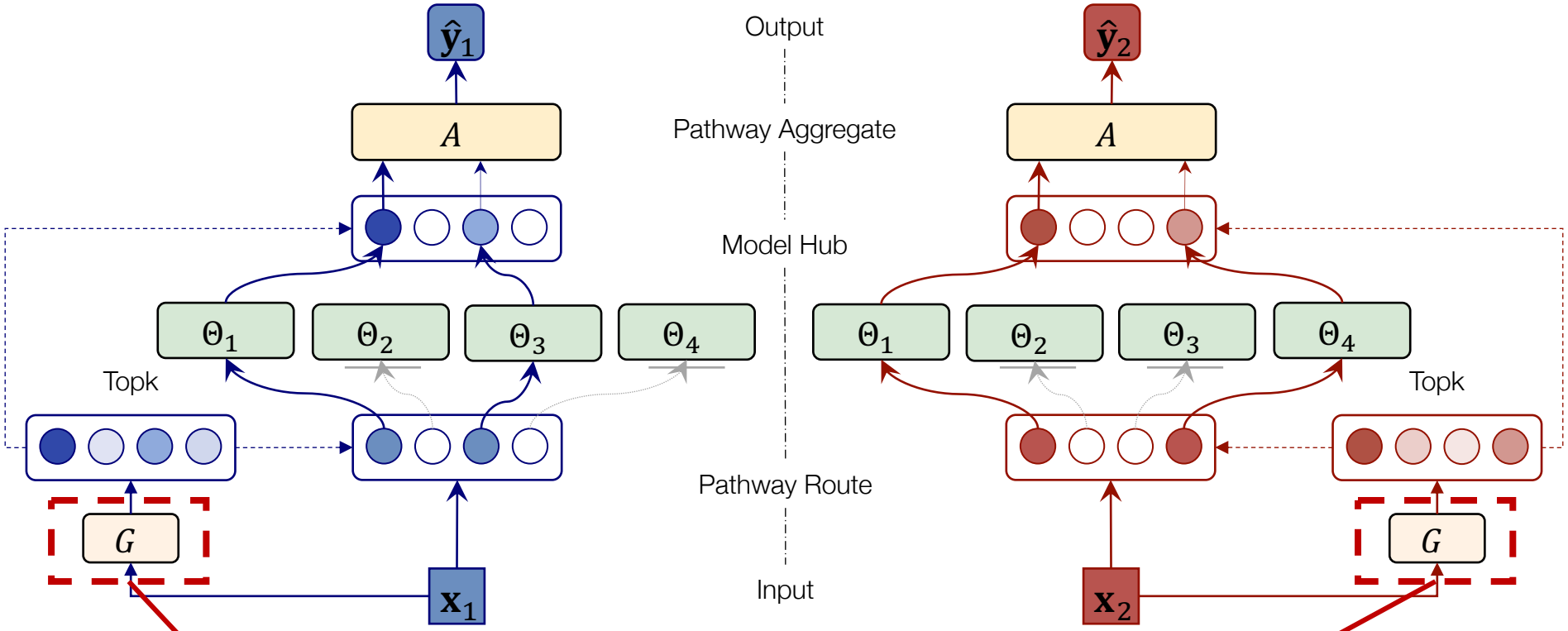
# Exploration and Exploitation




- How to explore better pathway configurations?
- How to fully exploit and transfer knowledge in pre-trained models?

# Enhance Exploration

- Direct training may activate only one or a few models → hub collapse.

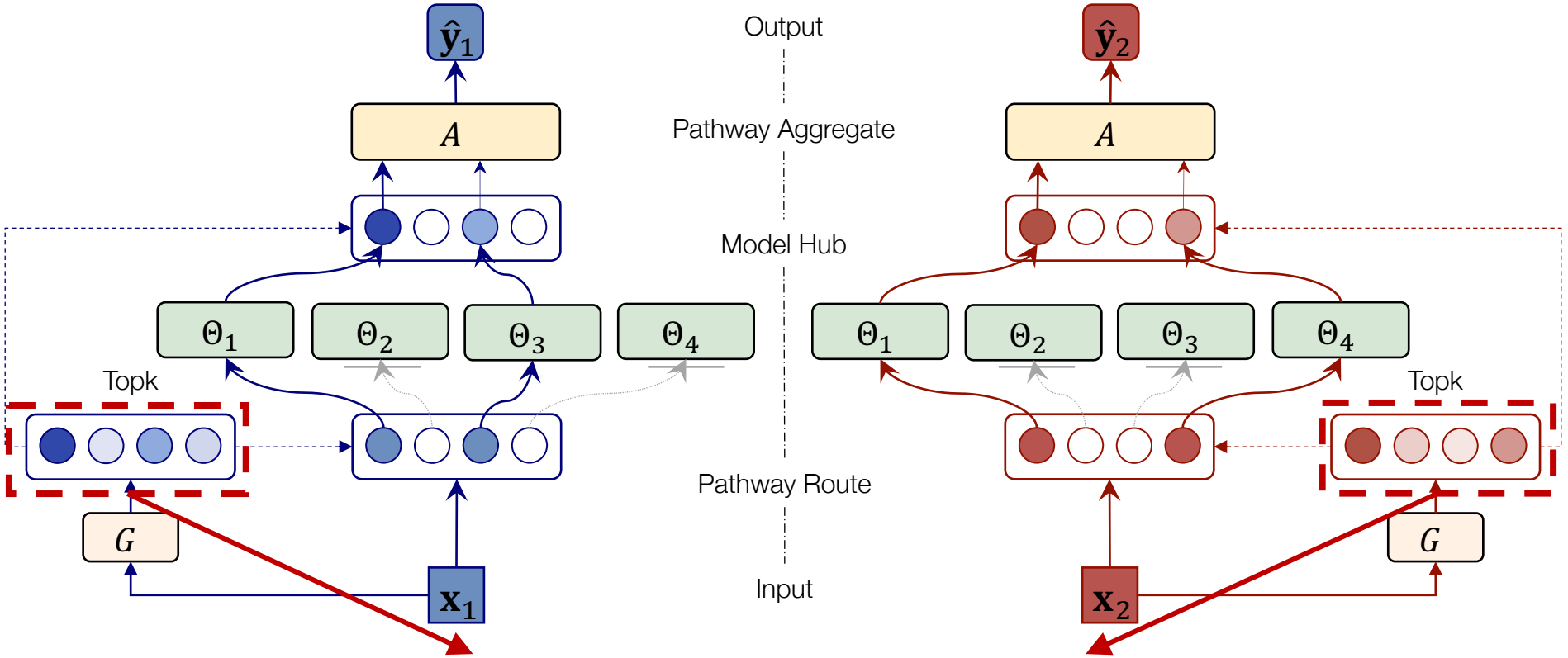



$$G(\mathbf{x}) = \mathbf{Softmax}\big(G_p(\mathbf{x}) + \epsilon \cdot \mathbf{Softplus}(G_n(\mathbf{x}))\big), \epsilon \sim \mathcal{N}(0,1)$$

- A noisy pathway generator to add randomness.

# Enhance Exploration

- Direct training may activate only one or a few models → hub collapse.

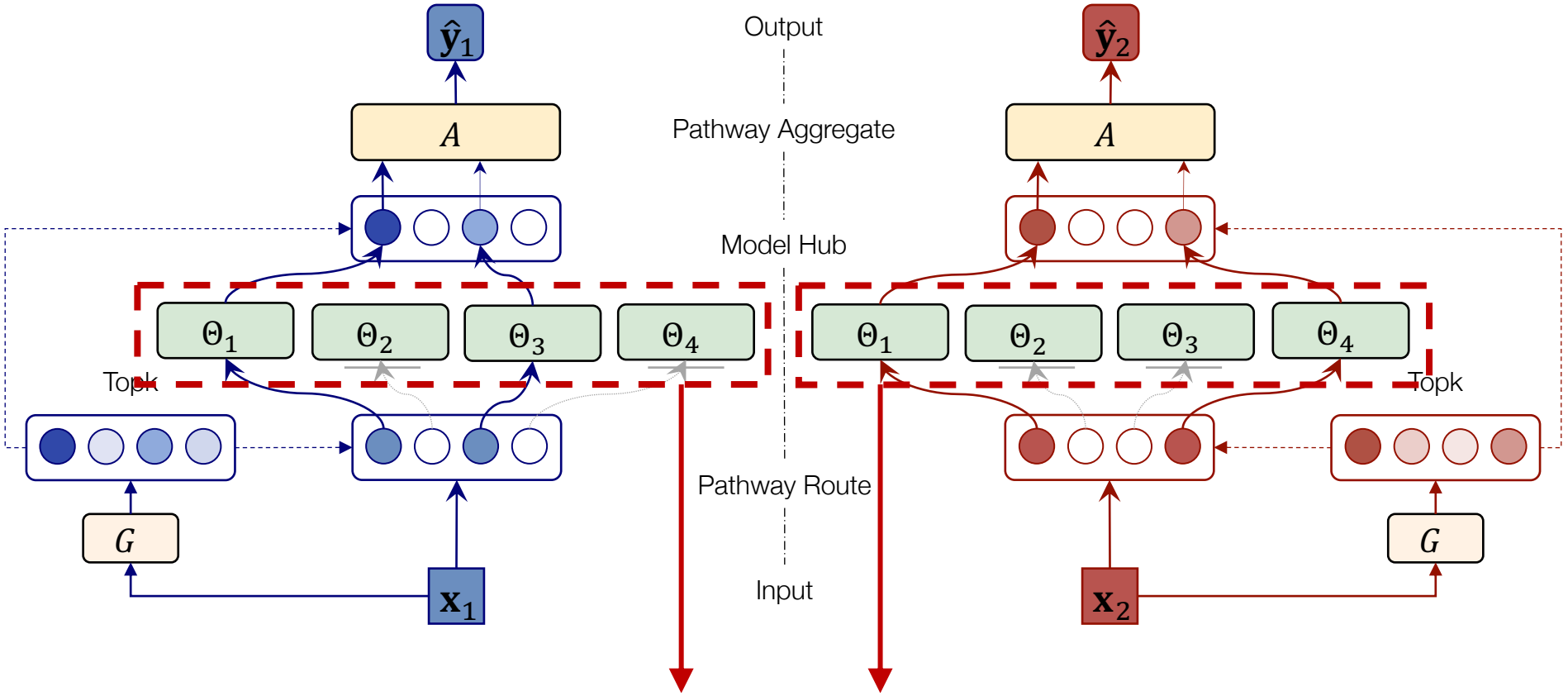



$$\mathcal{L}_{\text{explore}} = -\mathcal{H}\left(\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}} G(\mathbf{x})\right) \qquad \mathcal{H}(\mathbf{z}) = -\sum_i z_i \cdot \log(z_i)$$

- Inspired from RL: maximum-entropy regularization to encourage exploration.

# Enhance Exploitation

- Additional tuning with specific data to enhance knowledge transfer.




$$\mathcal{L}_{\text{exploit}} = \sum_{i=1}^{m} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}} \mathbb{I}\left(\bar{G}(\mathbf{x})_i > 0\right) \ell\left(\Theta_i(\mathbf{x}), \mathbf{y}\right)$$

# Experiments

- Data dependent pathways → General for heterogenous models.

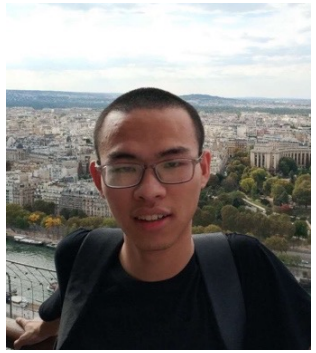
| Model | General | | Fine-Grained | | | Specialized | | Avg. |
|---|---|---|---|---|---|---|---|---|
| | CIFAR | COCO | Aircraft | Cars | Indoors | DMLab | EuroSAT | |
| MaskRCNN | 79.12 | 81.64 | 84.76 | 87.12 | 73.01 | 74.73 | 98.65 | 82.72 |
| MobileNetV3 | 83.14 | 83.28 | 80.26 | 86.37 | 75.09 | 70.09 | 98.95 | 82.45 |
| EffNet-B3 | 87.28 | 86.97 | 83.99 | 89.34 | 78.16 | 72.69 | 99.13 | 85.37 |
| Swin-T | 84.37 | 84.12 | 80.82 | 89.10 | 73.39 | 72.22 | 98.69 | 83.24 |
| ConvNeXt-T | 86.96 | 87.15 | 84.23 | 90.67 | 81.66 | 73.80 | 98.65 | 86.16 |
| Ensemble | 87.72 | 88.04 | 87.11 | 92.68 | 82.79 | **74.86** | 99.23 | 87.49 |
| Distill | 87.33 | 88.09 | 85.26 | 91.39 | 81.51 | 74.75 | 99.24 | 86.80 |
| Hub-Pathway | **89.01** | **89.14** | **88.12** | **92.93** | **84.40** | 74.80 | **99.26** | **88.24** |

- Control the costs with top-k activation→ More efficient than ensemble.

| Model | Acc (%) | Params (M) | FLOPs (G) | Train (iters/s) | Inference (samples/s) |
|---|---|---|---|---|---|
| ImageNet | 83.41 | 23.71 | 4.11 | 10.87 | 484.92 |
| Ensemble | 84.50 | 118.55 | 20.55 | 2.30 | 98.64 |
| Hub-Pathway | 85.63 | 128.43 | 9.11 | 4.68 | 240.48 |

# Adaptive Transfer with Generated Pathways

# Thank You!



游凯超

树扬

刘雍

龙明盛

王建民

大数据系统软件国家工程研究中心

清華大學 Tsinghua University