Transferability in Deep Learning

Mingsheng Long

School of Software, Tsinghua University

MLA 2022

Tsinghua University

Outline

Transferability in Deep Learning

Suggested Citation: Junguang Jiang, Yang Shu and Mingsheng Long (2022), "Transferability in Deep Learning", : Vol. xx, No. xx, pp 1–18. DOI: 10.1561/XXXXXXXXX.

Junguang Jiang
Tsinghua University
jiangjunguang1123@outlook.com

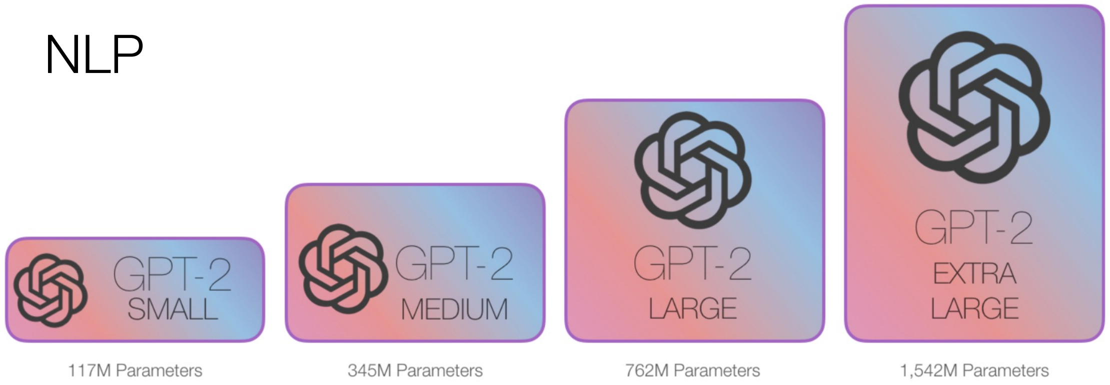
Yang Shu
Tsinghua University
shu-y18@mails.tsinghua.edu.cn

Mingsheng Long
Tsinghua University
mingsheng@tsinghua.edu.cn

now
the essence of knowledge
Boston — Delft

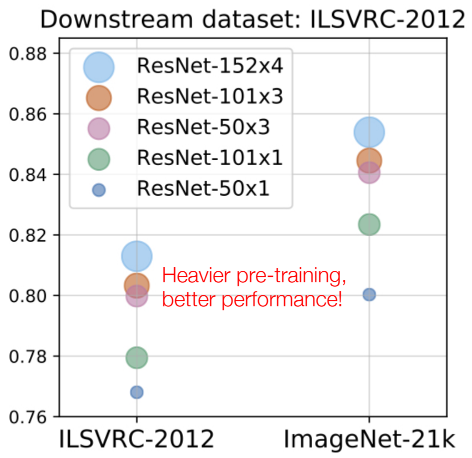
Pre-training

NLP



GPT: Large-scale Corpus Pre-training

CV



BiT: General Visual Representation Learning

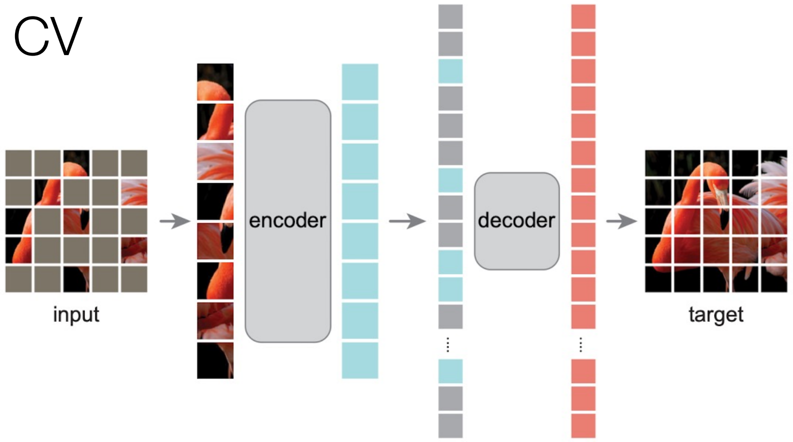
GPT — 2018

GPT-2, BERT — 2019

GPT-3, BiT, MoCo — 2020

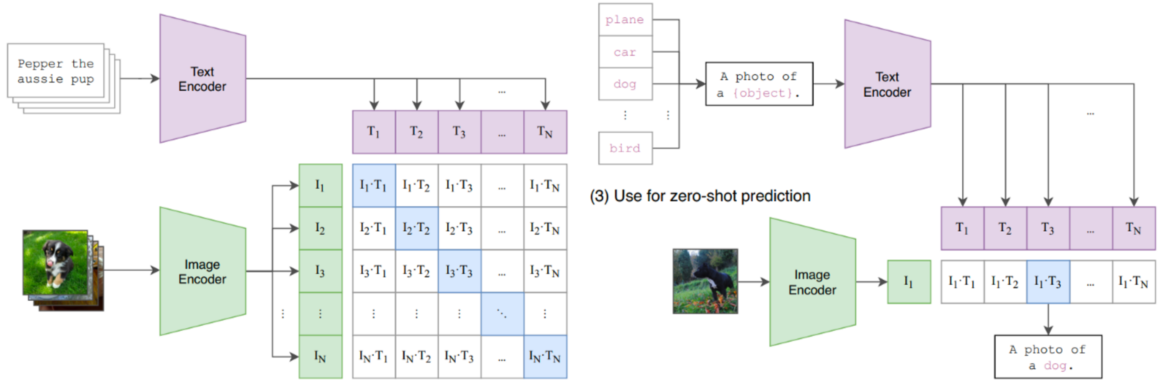
MAE, CLIP — 2021

CV



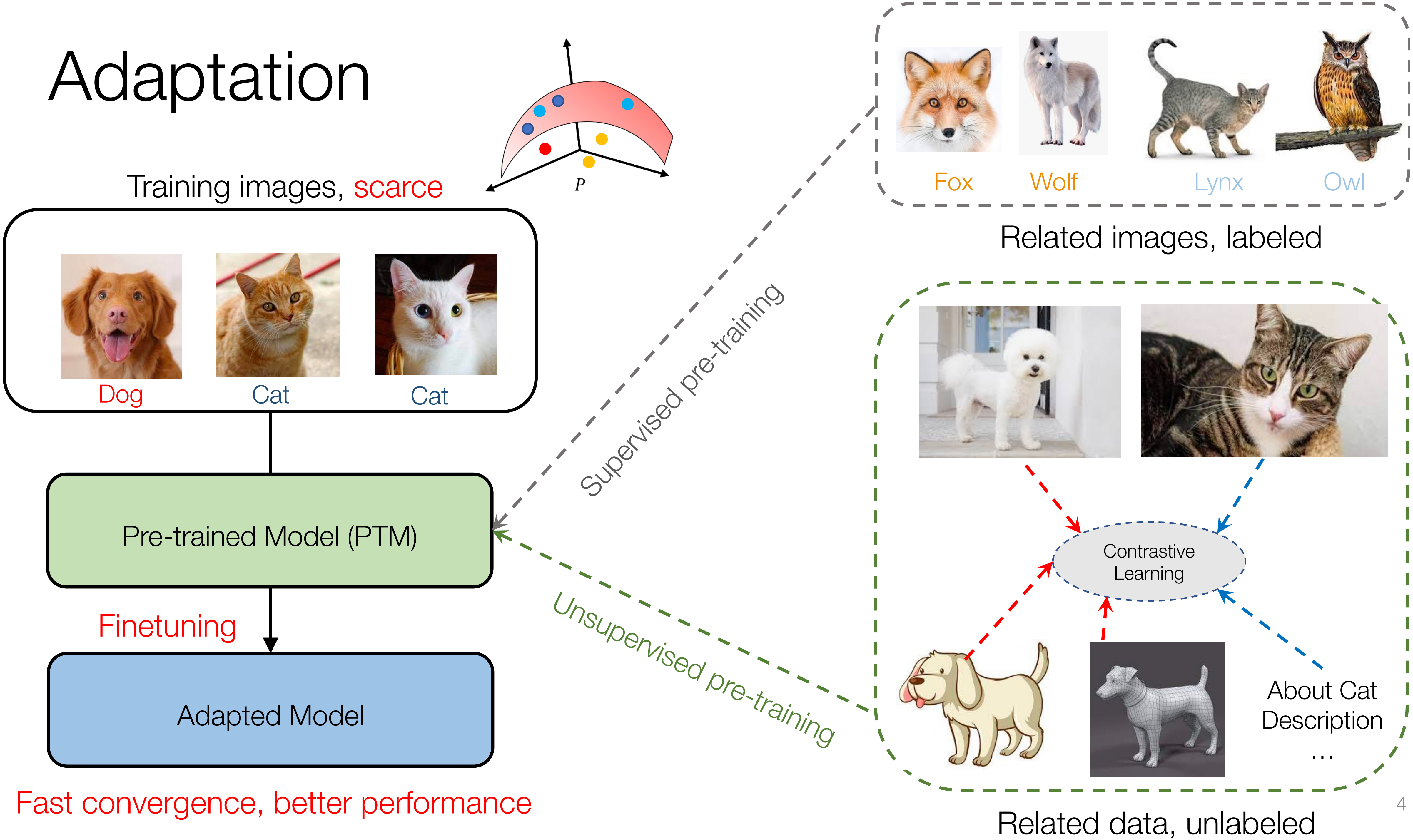
MAE: Masked AutoEncoder as Self-supervised Learner

All In One



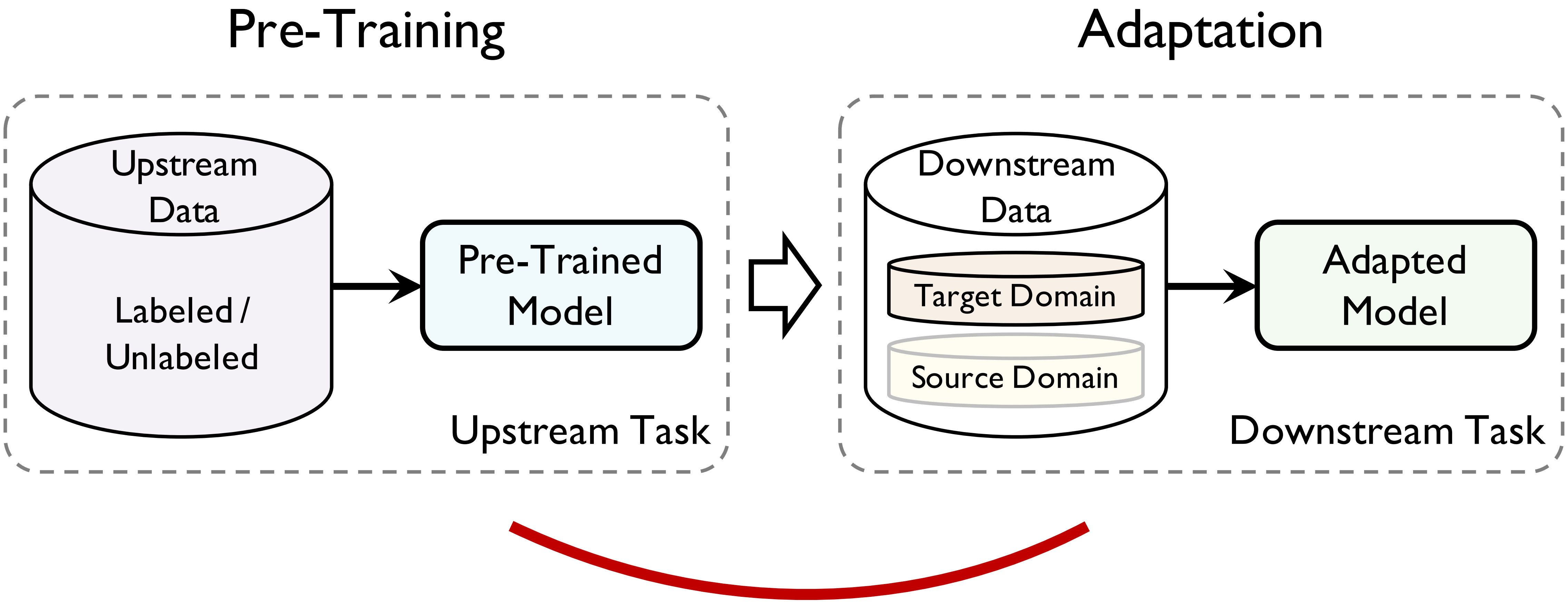
CLIP: Contrastive Language-Image Pre-Training

Adaptation



P
Training images, scarce
Dog
Cat
Cat
Pre-trained Model (PTM)
Finetuning
Adapted Model
Fast convergence, better performance
Supervised pre-training
Unsupervised pre-training
Fox
Wolf
Lynx
Owl
Related images, labeled
Contrastive Learning
About Cat Description
...
Related data, unlabeled

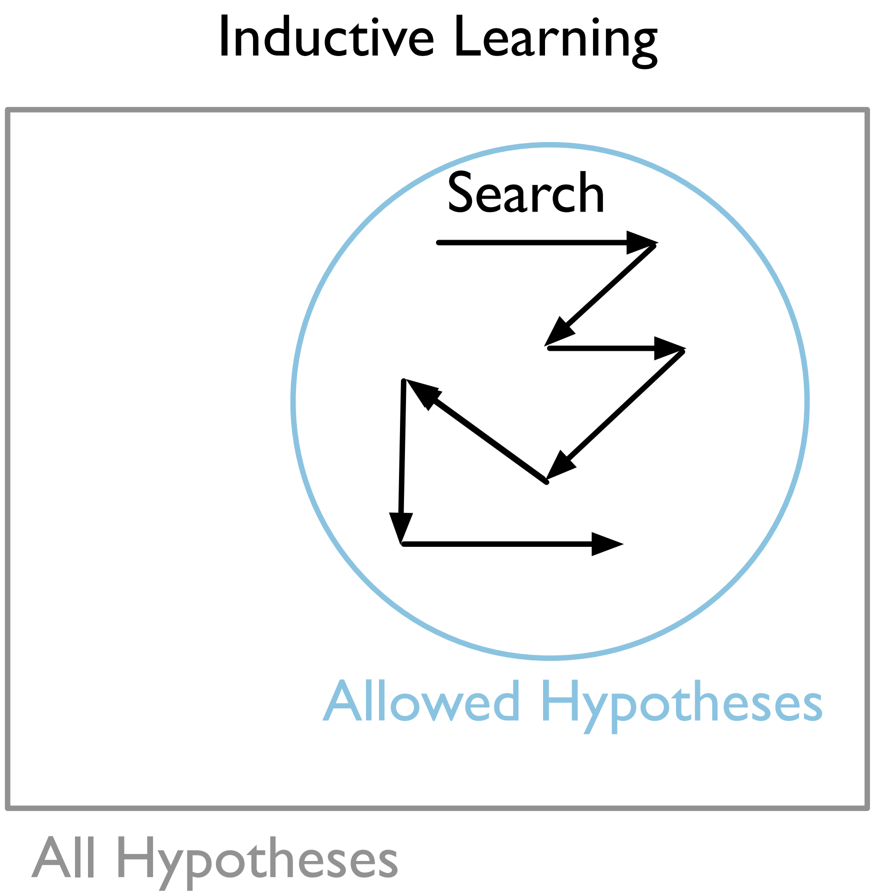
Pre-training and Adaptation

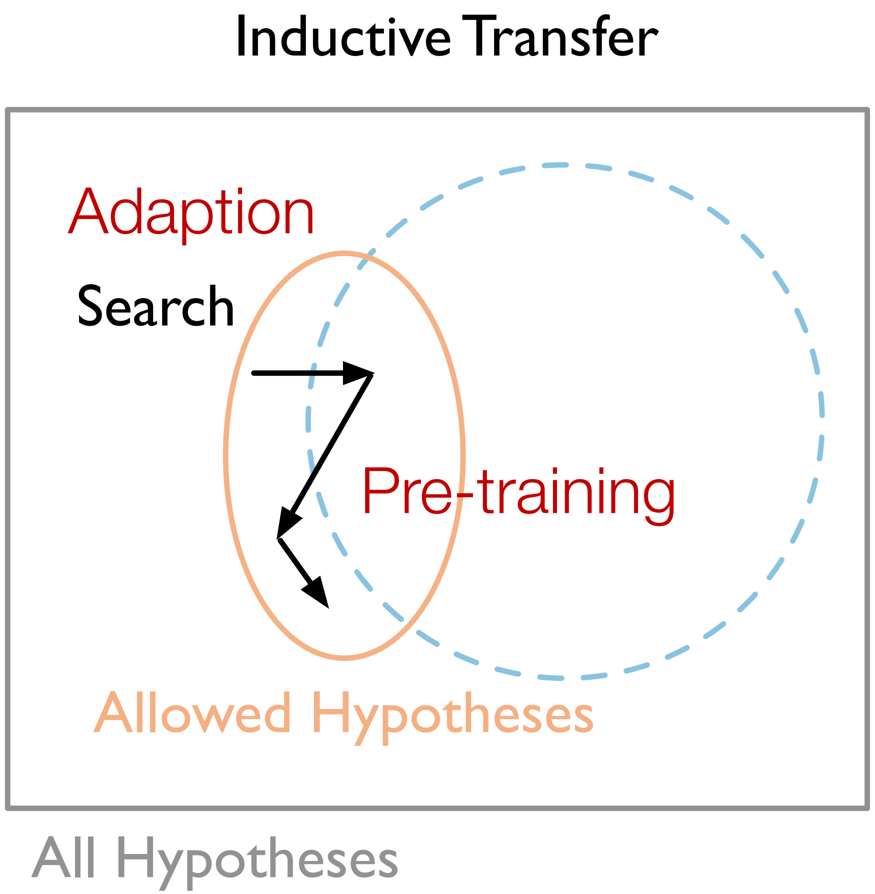


Pre-training → Adaptation

A Paradigm for Deep Learning Application

Pre-training and Adaptation

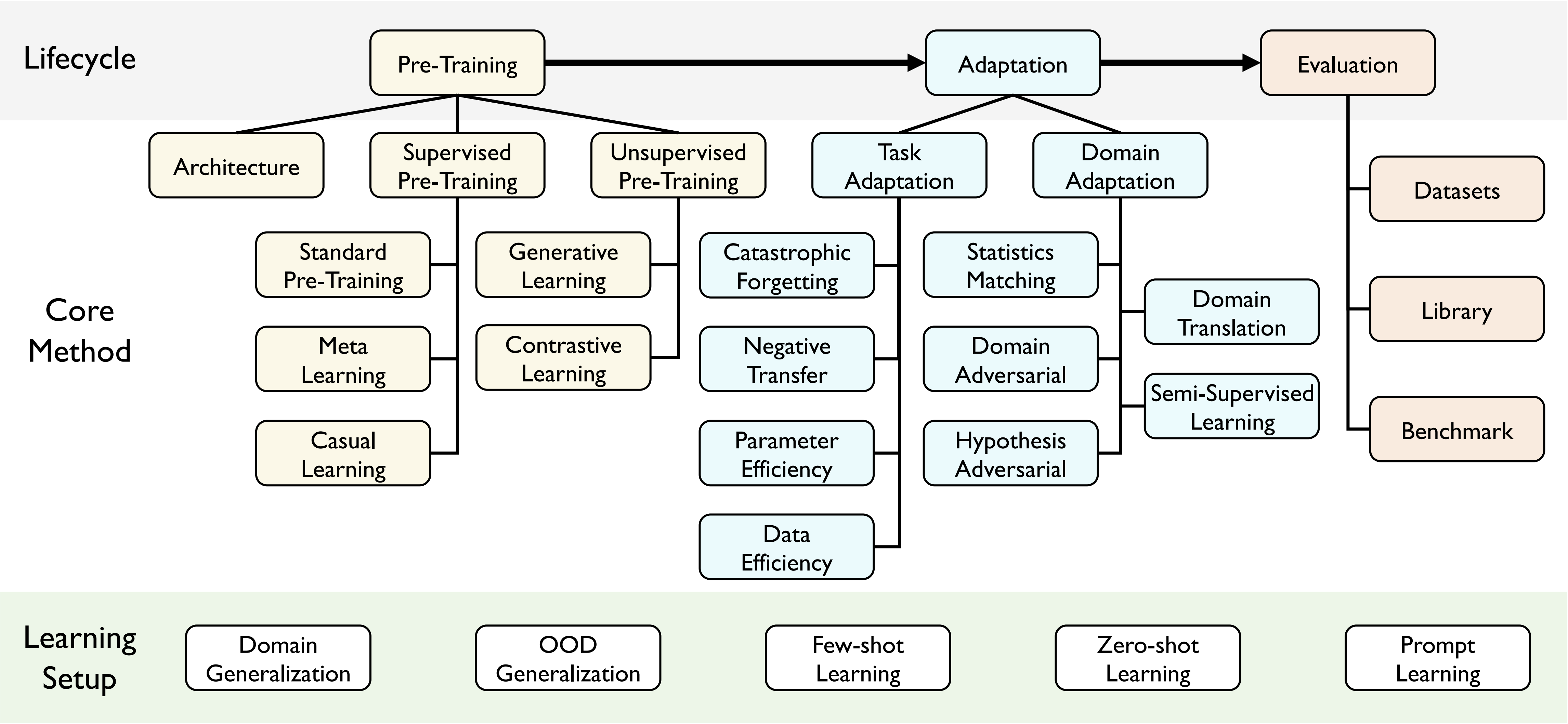




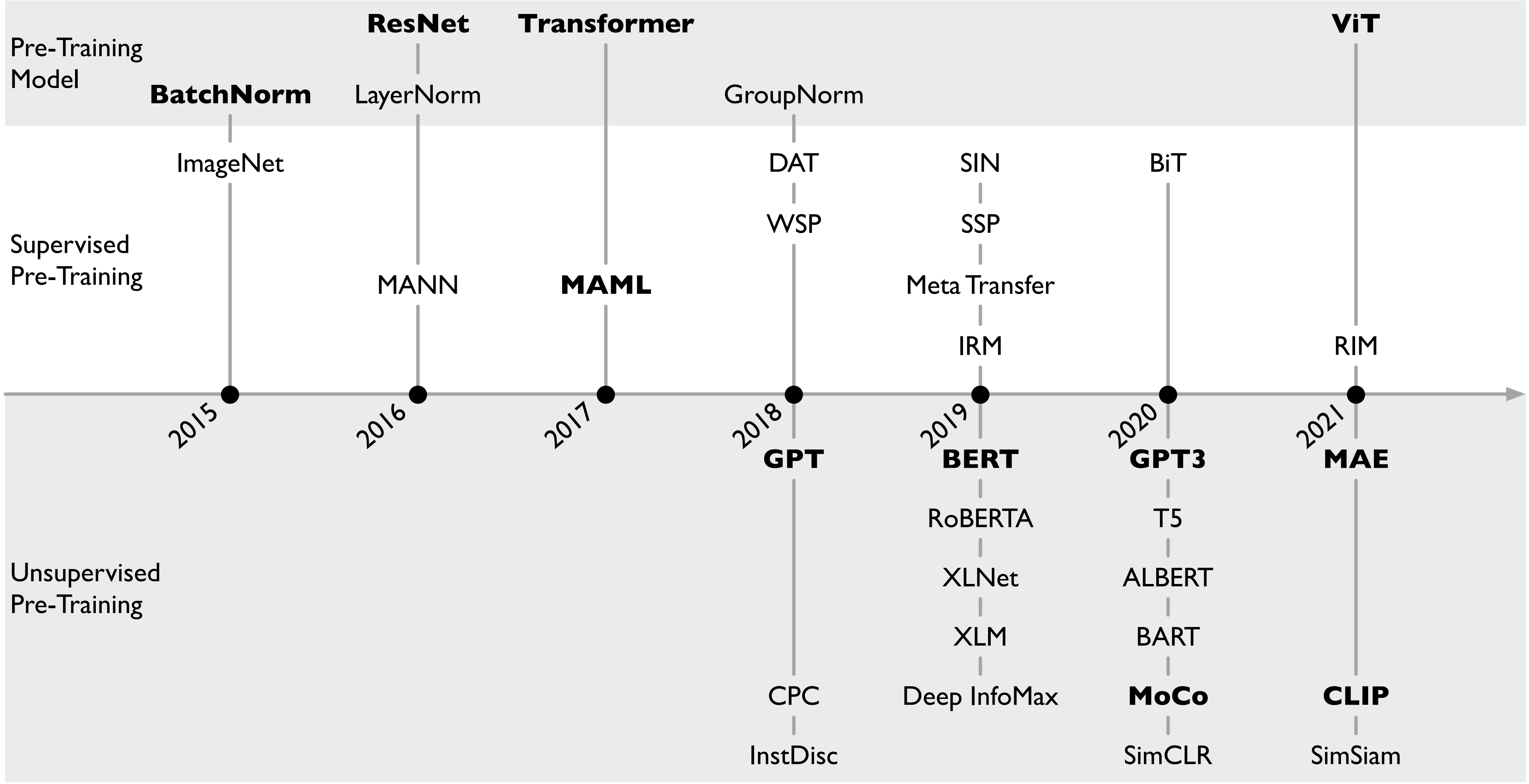
Pre-training → Adaptation

A Paradigm for Deep Learning Application

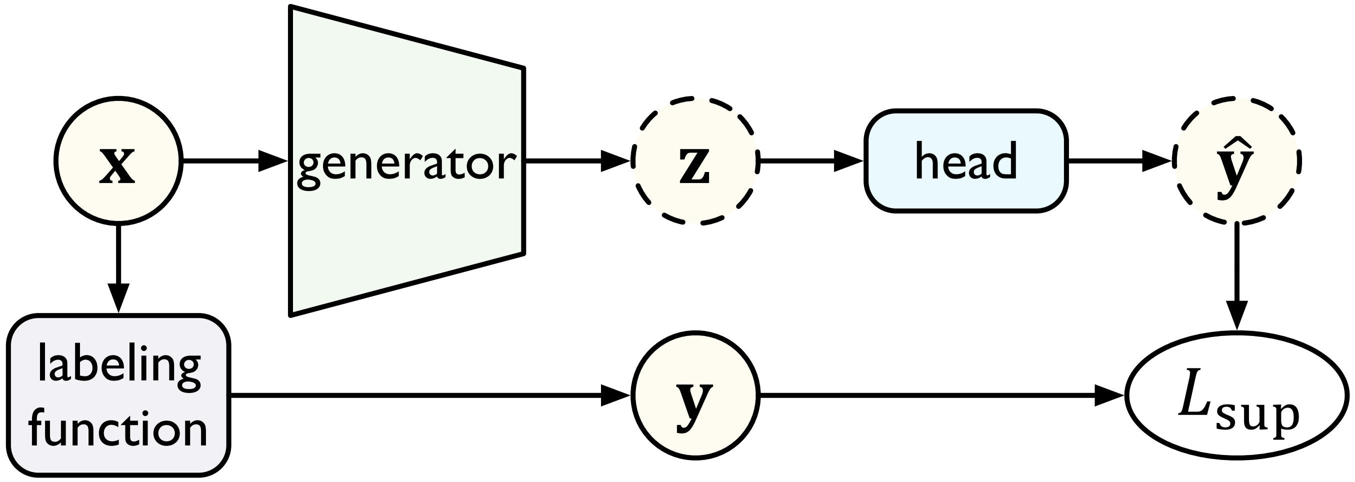
Transferability in the Lifecycle



Pre-training

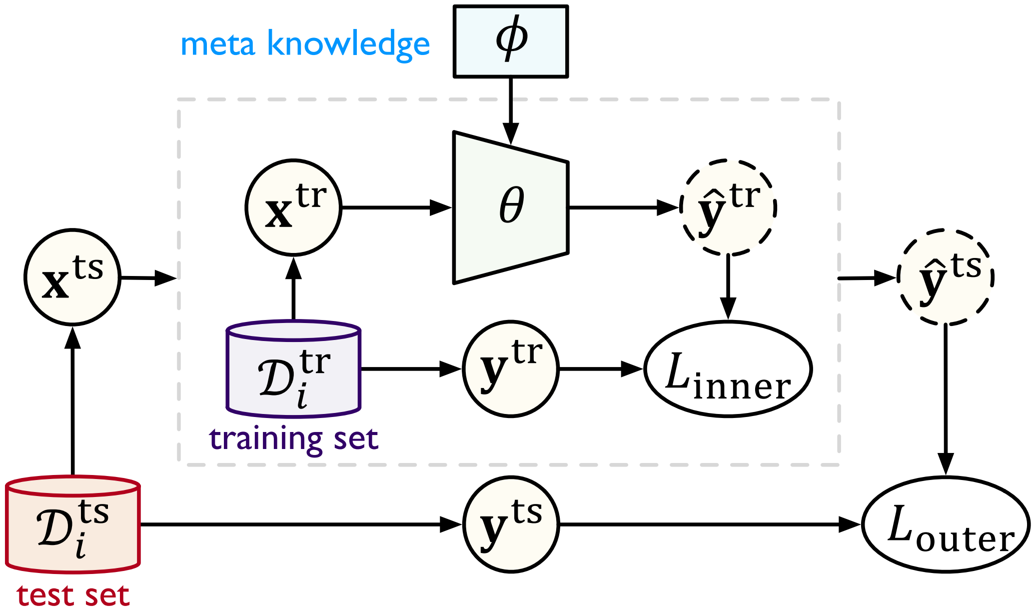


Supervised Pre-training



- Big Transfer (BiT) (Kolesnikov et al., 2020) emphasizes that training on larger datasets is vital for better transferability.
- Domain Adaptive Transfer (DAT) (Ngiam et al., 2018) uses importance weighting to carefully choose the pre-training data that are most relevant to the target task.

Meta-Learning



Task $i \in [1, \dots, n]$

$$\phi^* = \arg\max_{\phi} \sum_{i=1}^{n} \log P(\theta_i(\phi)|\mathcal{D}_i^{\text{ts}}),$$

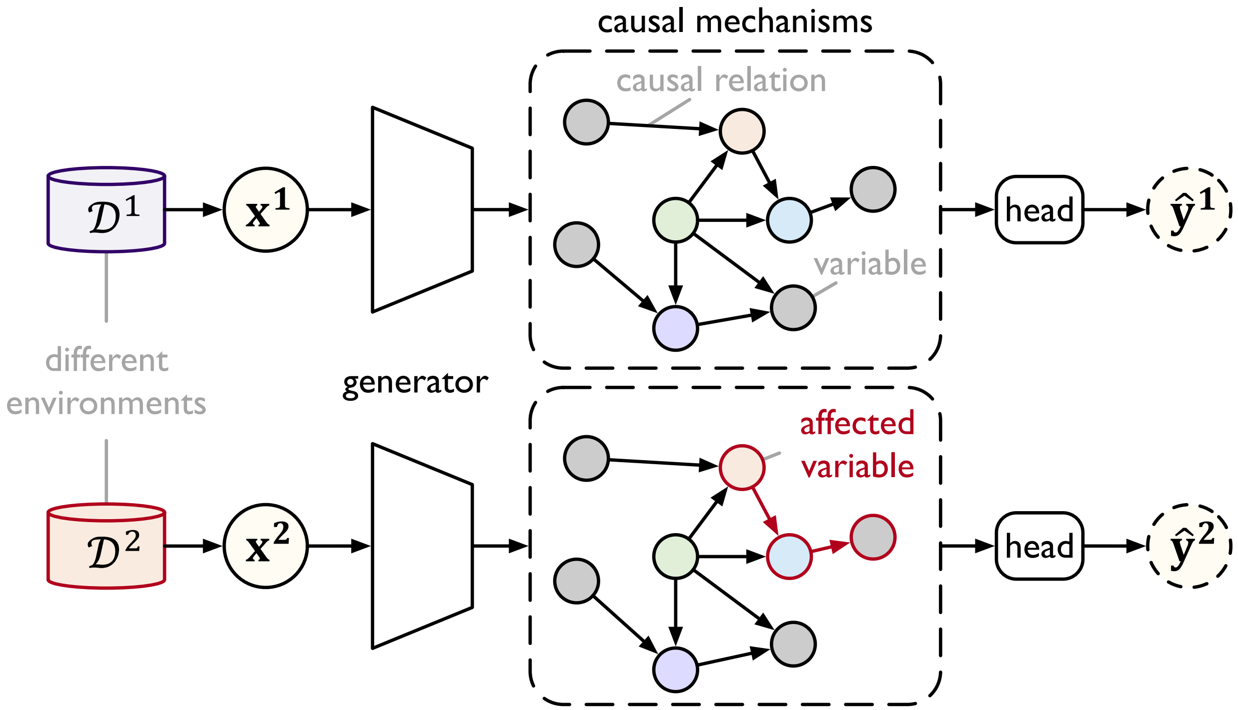
$$\text{where } \theta_i(\phi) = \arg\max_{\theta} \log P(\theta|\mathcal{D}_i^{\text{tr}}, \phi).$$

Model-Agnostic Meta-Learning **(MAML)**

for fast adaptation

$$\theta_i = \phi - \alpha \nabla_{\phi} L(\phi, \mathcal{D}_i^{\text{tr}})$$

Causal Learning



Invariant Risk Minimization **(IRM)**

for OOD generalization

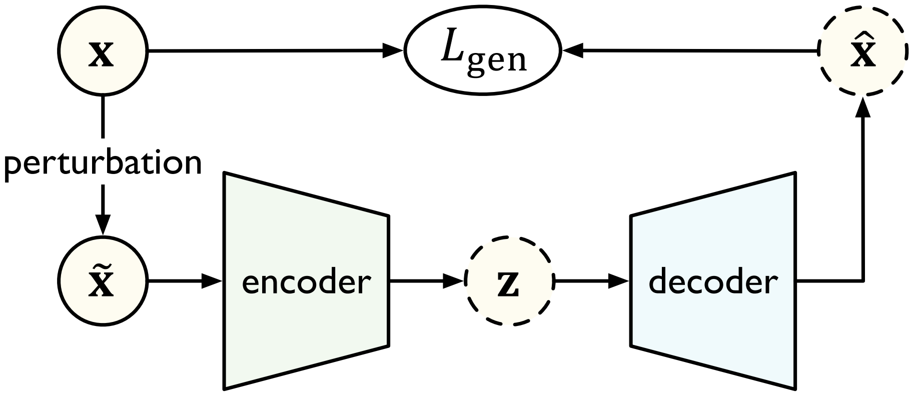
$$\min_{\psi:\mathcal{X}\to\mathcal{Z},h:\mathcal{Z}\to\mathcal{Y}} \sum_{e\in\mathcal{E}^{\mathrm{tr}}} \epsilon^e(h\circ\psi),$$

$$\text{subject to } h\in\arg\min_{\bar{h}:\mathcal{Z}\to\mathcal{Y}} \epsilon^e(\bar{h}\circ\psi), \text{ for all } e\in\mathcal{E}^{\mathrm{tr}}$$

Essentially, this implies invariance to **data augmentation**!

- Causal learning seeks a model with causal mechanisms, and if the environment or distribution changes, only part of the causal mechanisms will be affected.

Generative Learning

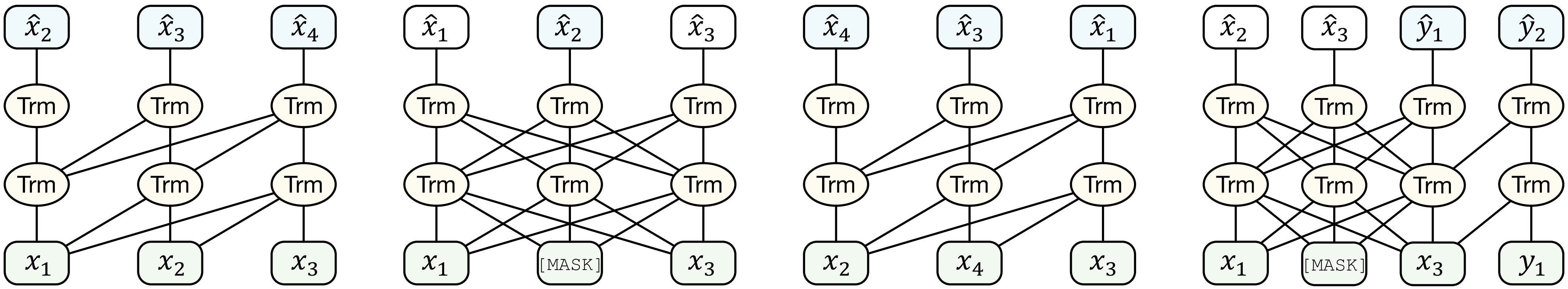


Autoregression
GPT, Image-GPT

$$\max_{\theta} \sum_{t=1}^{T} \log P_{\theta}(x_t | x_{t-k}, \cdots, x_{t-1})$$

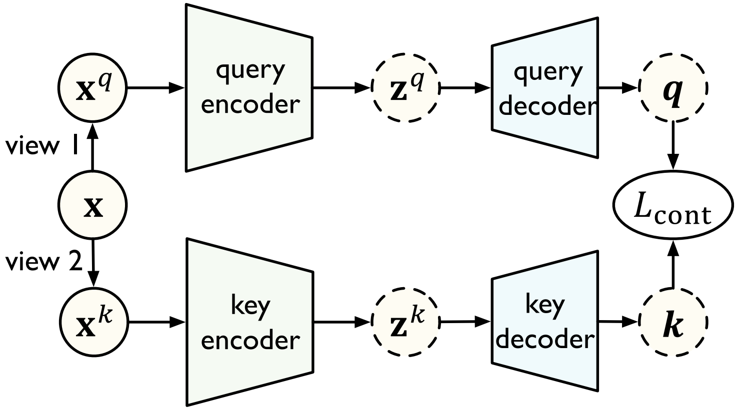
Autoencoding
BERT, MAE

$$\max_{\theta} \sum_{x \in m(\mathbf{x})} \log P_{\theta}(x | \mathbf{x}_{\setminus m(\mathbf{x})})$$



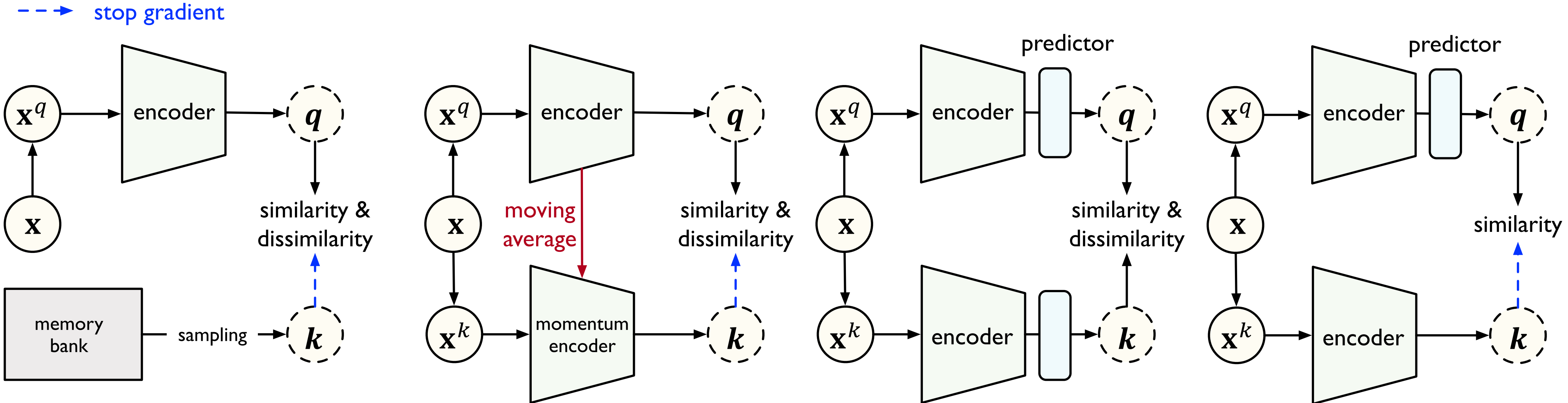
(a) LM (b) MLM (c) PLM (d) Seq2Seq MLM

Contrastive Learning



$$\min_{\psi} - \log \frac{\exp(\mathbf{q} \cdot \mathbf{k}_{+}/\tau)}{\sum_{j=0}^{K} \exp(\mathbf{q} \cdot \mathbf{k}_j/\tau)}$$

Supervised pre-training gains high-level semantic knowledge, while contrastive and generative pre-training gains mid-level & low-level representations



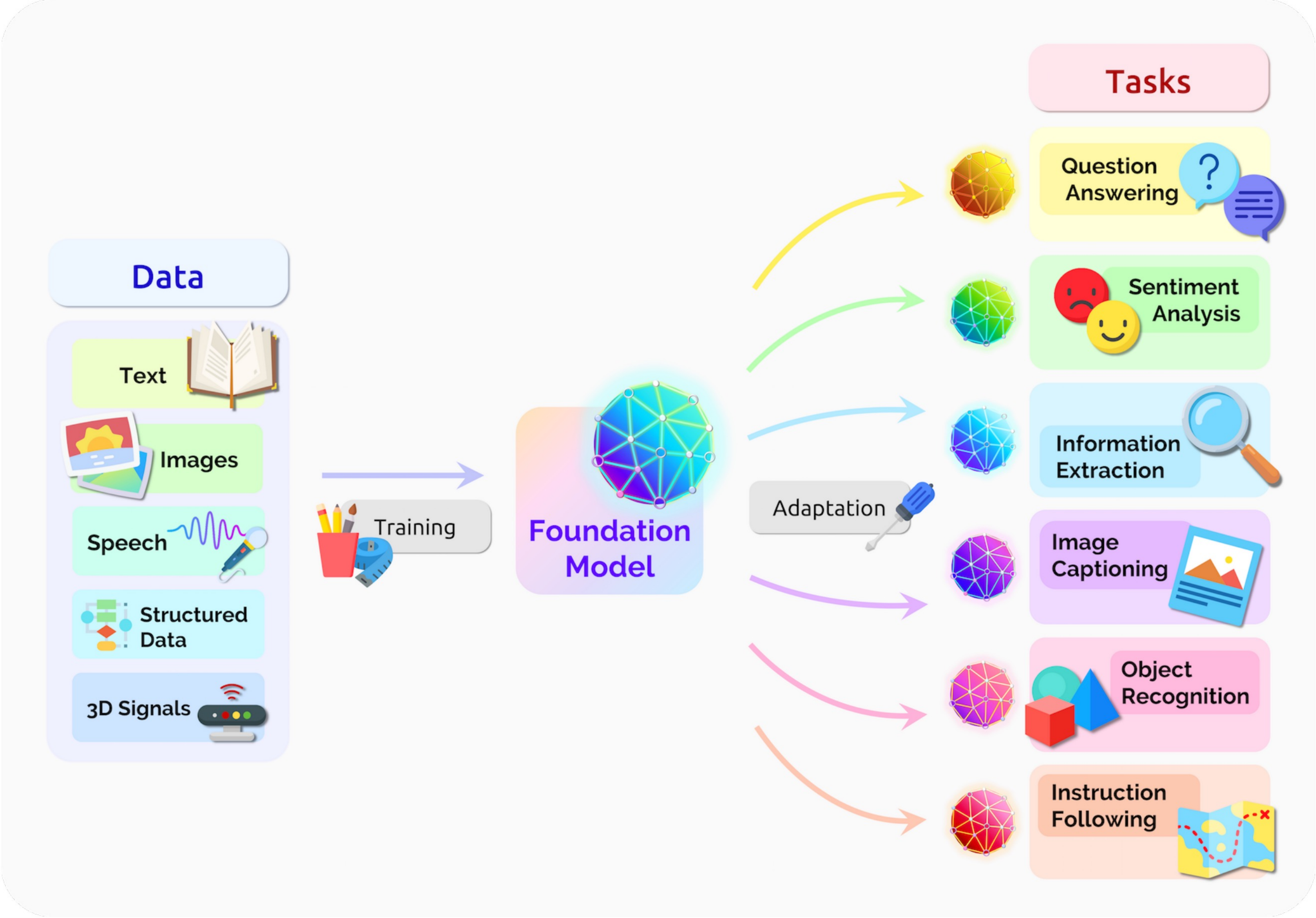
(a) InstDisc (b) MoCo (c) SimCLR (d) SimSiam

Remarks on Pre-training

Method	Modality Scalability[1]	Task Scalability[2]	Data Efficiency[3]	Label Efficiency[4]
Standard Pre-Training	★★★	★★	★★★	★
Meta-Learning	★★★	★	★	★
Causal Learning	★★	★	★	★
Generative Learning	★★	★★★	★★★	★★★
Contrastive Learning	★	★★★	★★★	★★★

[1] Whether models can be pre-trained on various modalities, such as text, graph.
[2] Whether pre-trained models can be easily transferred to many downstream tasks.
[3] Whether stronger transferability can be yielded from large-scale pre-training.
[4] Whether pre-training relies on manual data labeling.

Foundation Model



[Data Universal]

Learn from various modalities

[Task Universal]

Adapt to a wide range of downstream tasks

Bommasani et al. On the Opportunities and Risks of Foundation Models. Arxiv 2021.

General Relation Modeling in Transformers

Image

Relation among Image Patches

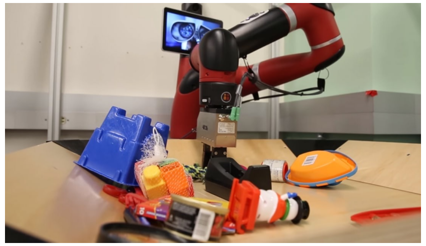
Language

Relation among Words

[SOS] Flowformer is a task-universal linear Transformer. [EOS]

Time Series

Relation among Time Points

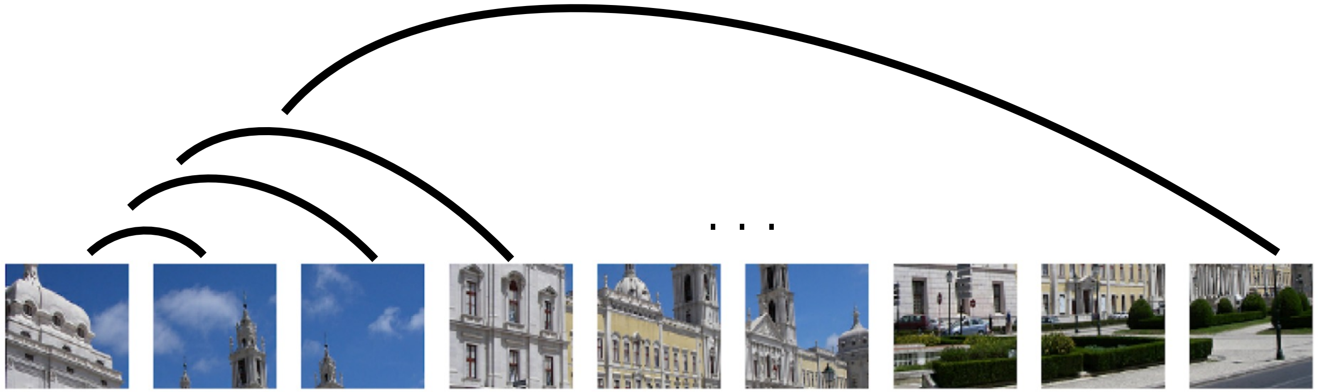


Agent Trajectory

Relation among Agent-Environment Interactions

… [S] [A] [R] …

Quadratic Complexity in Self-Attention



Pair-wise Relation Modeling: $\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$

$\Downarrow$

General Relation Modeling

[Quadratic Complexity]

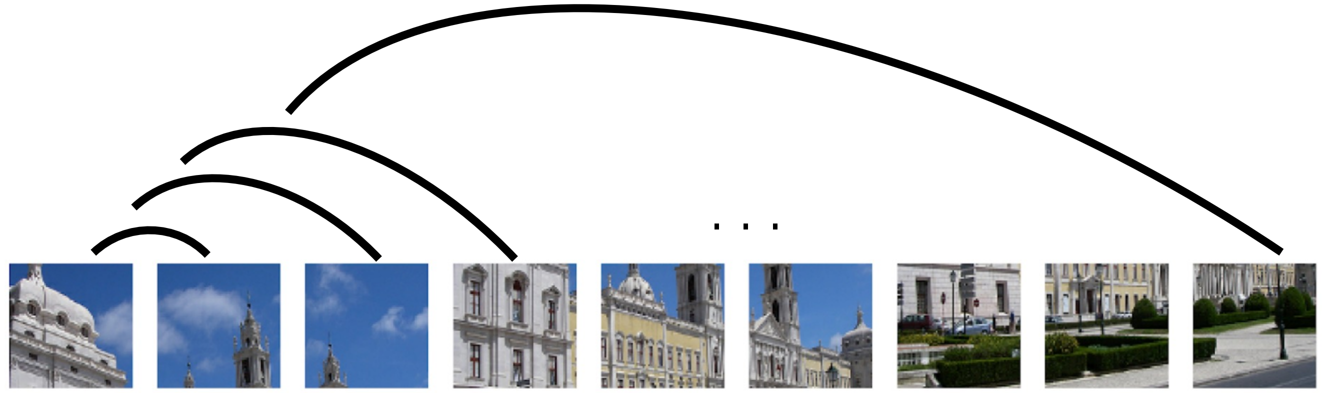
Long Sequence
Model Efficiency
Big Model ☹

Task & Data Universal

Foundation Model

?

Quadratic Complexity in Self-Attention



Pair-wise Relation Modeling: $\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$

$\mathcal{O}(n^2 d)$

Can we remove Softmax function?

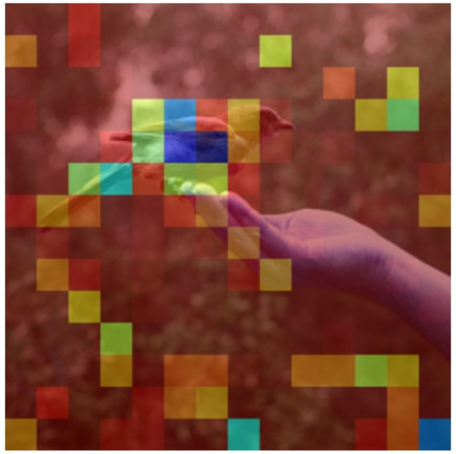
$(\boldsymbol{QK^T})\boldsymbol{V} = \boldsymbol{Q}(\boldsymbol{K^T V}) \Rightarrow \mathcal{O}(n^2 d) \rightarrow \mathcal{O}(nd^2)$

Recap: Softmax function

Softmax function is proposed as a differentiable generalization of the "winner-take-all" picking maximum operation.

Competition Mechanism

The key to avoid trivial attention



Bridle et al. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. NeurIPS 1989.

Recap: Softmax function

Softmax function is proposed as a differentiable generalization of the "winner-take-all" picking maximum operation.

$$\boldsymbol{\phi}(\boldsymbol{Q})(\boldsymbol{\phi}(\boldsymbol{K})^T\boldsymbol{V}) \quad \Longleftrightarrow \quad \textbf{Softmax}\left(\frac{\boldsymbol{QK}^T}{\sqrt{\boldsymbol{d}}}\right)\boldsymbol{V}$$

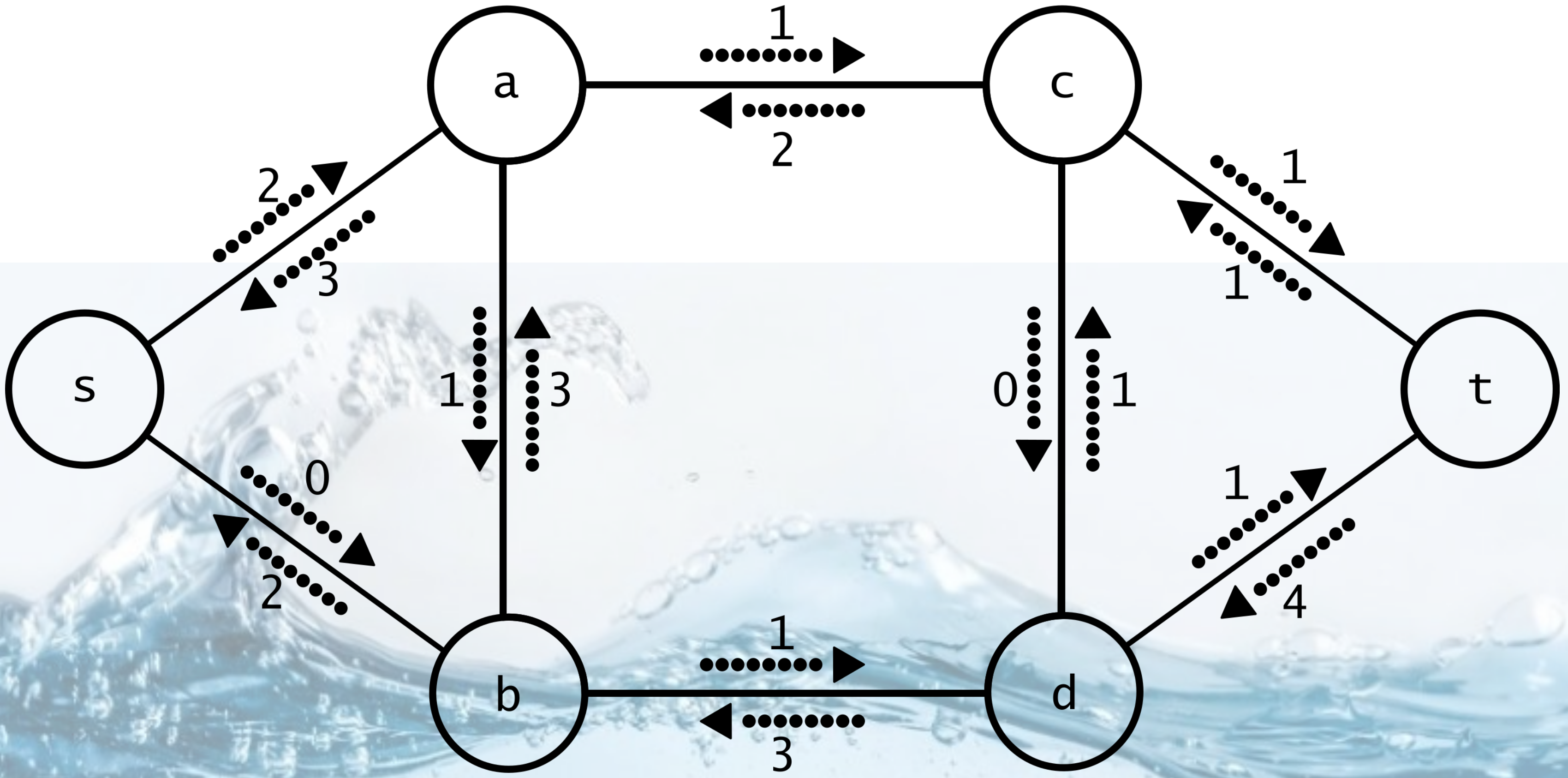
+

Competition Mechanism

"fixed resource will cause competition"

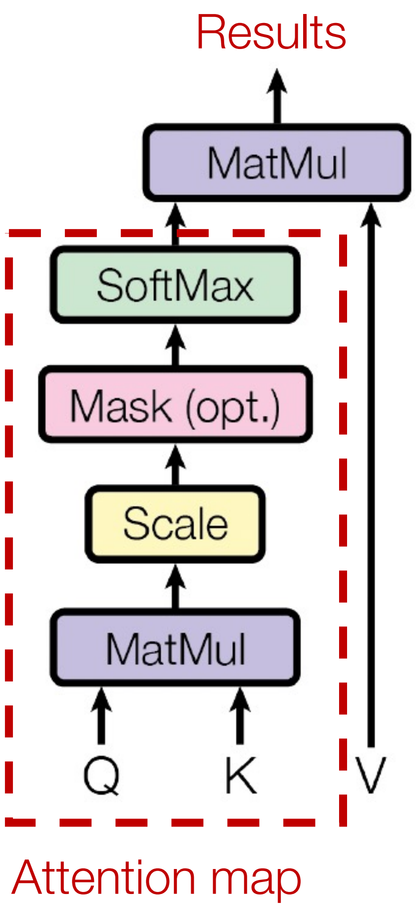
Bridle et al. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. NeurIPS 1989.

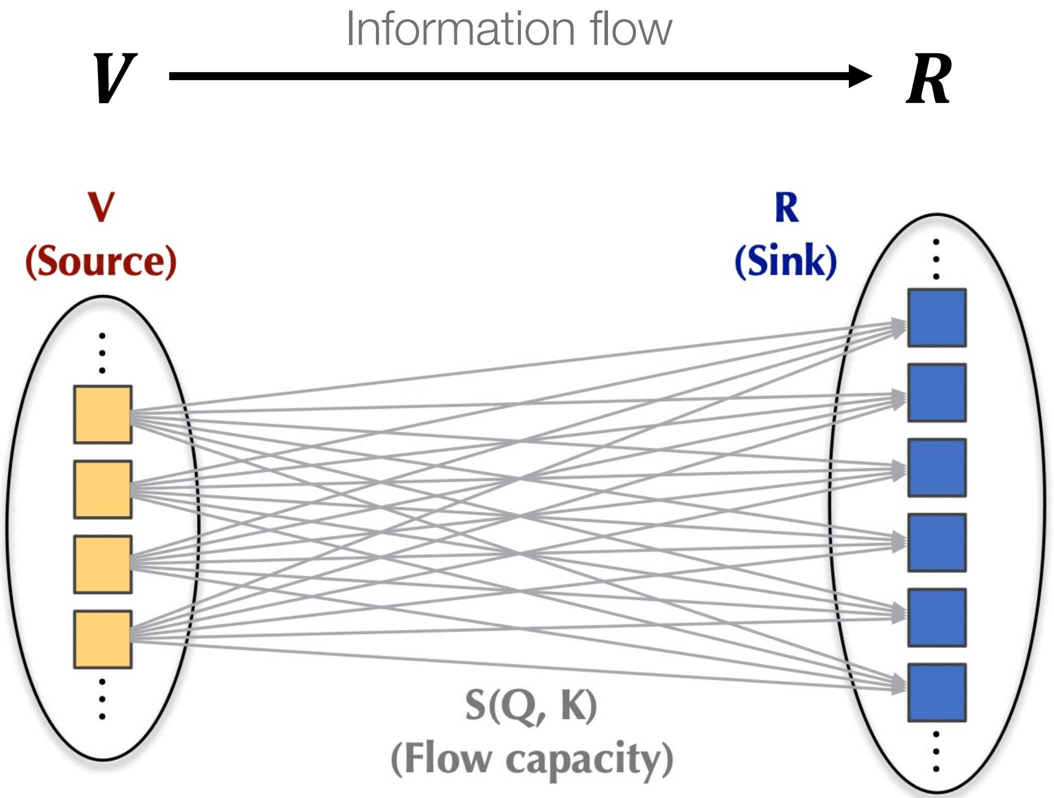
Flow Network Theory



[Conservation Property]: The incoming flow capacity of each node is equal to the outgoing flow.

Attention: A Flow Network View

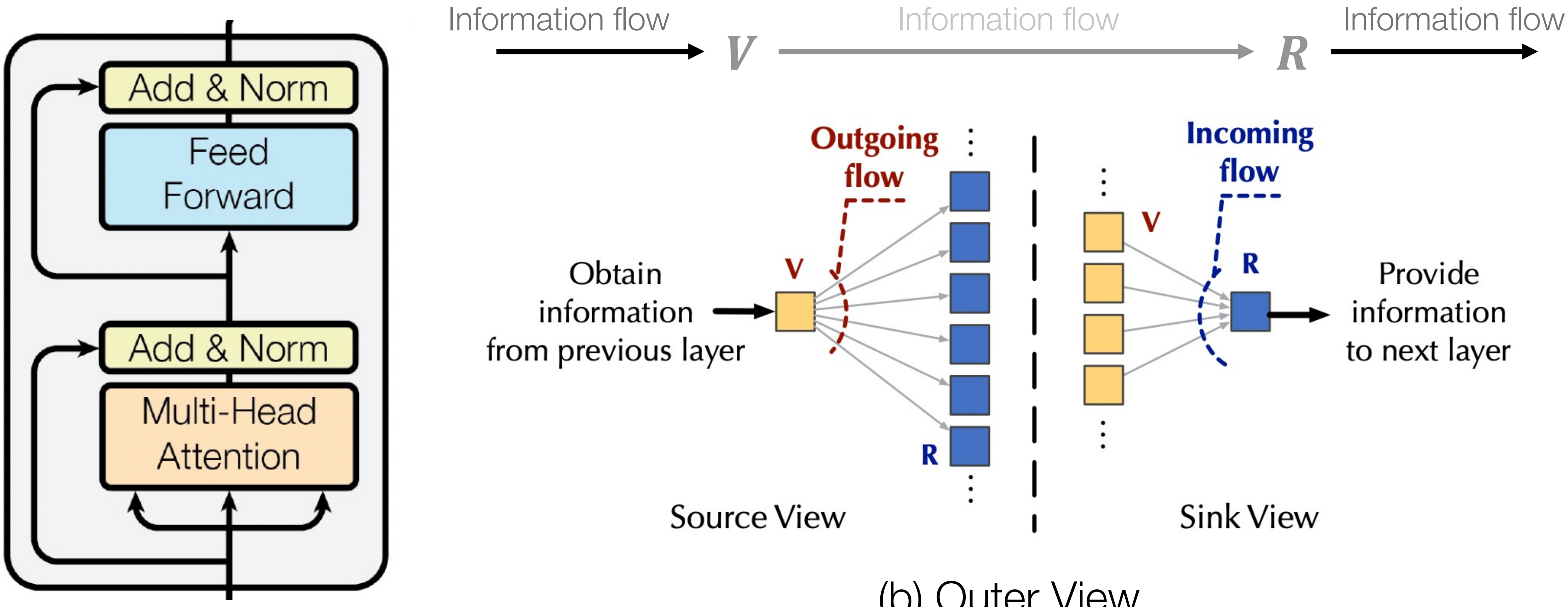




(a) Inner View

Haixu Wu, Jialong Wu, Jiehui Xu, Jianmin Wang, Mingsheng Long. Flowformer: Linearizing Transformers with Conservation Flows. **ICML**, 2022.

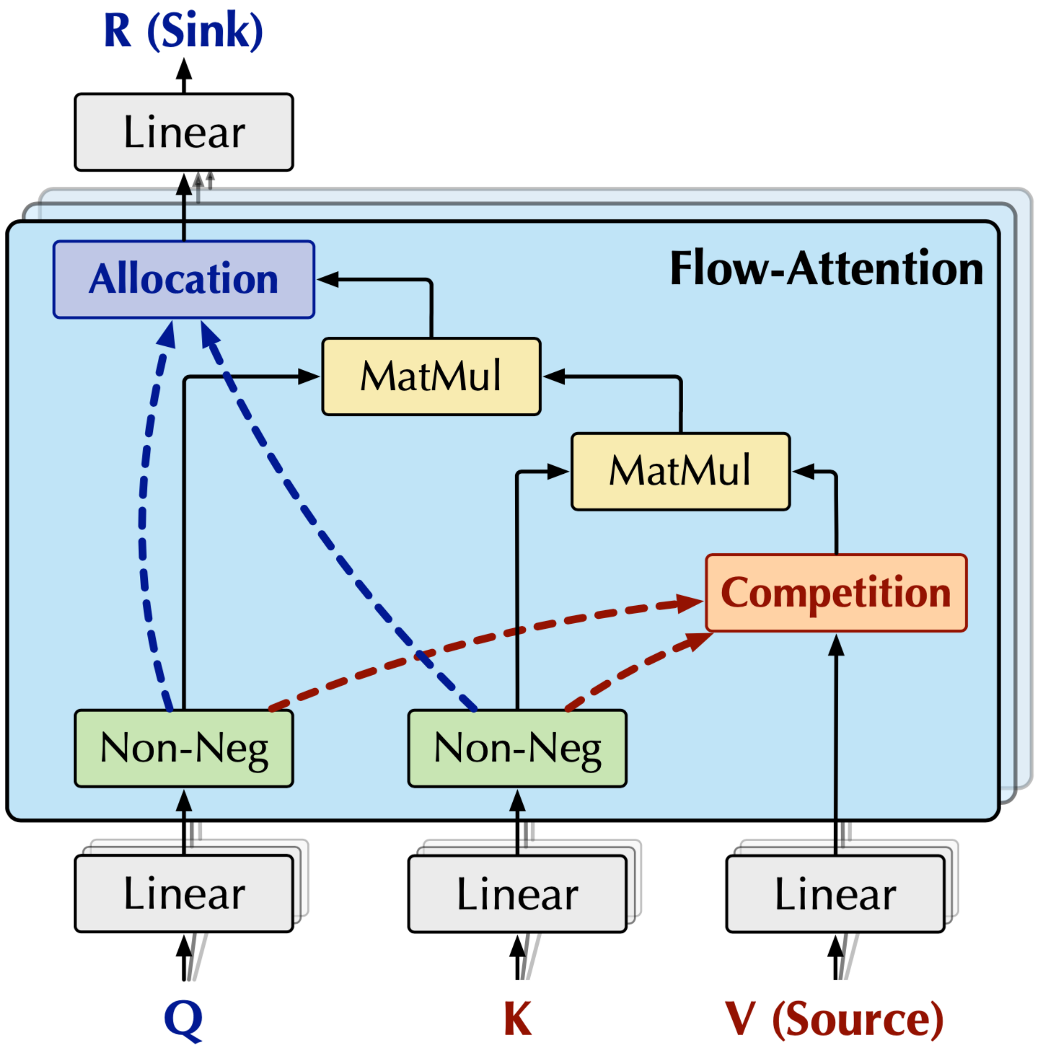
Attention: A Flow Network View



(b) Outer View

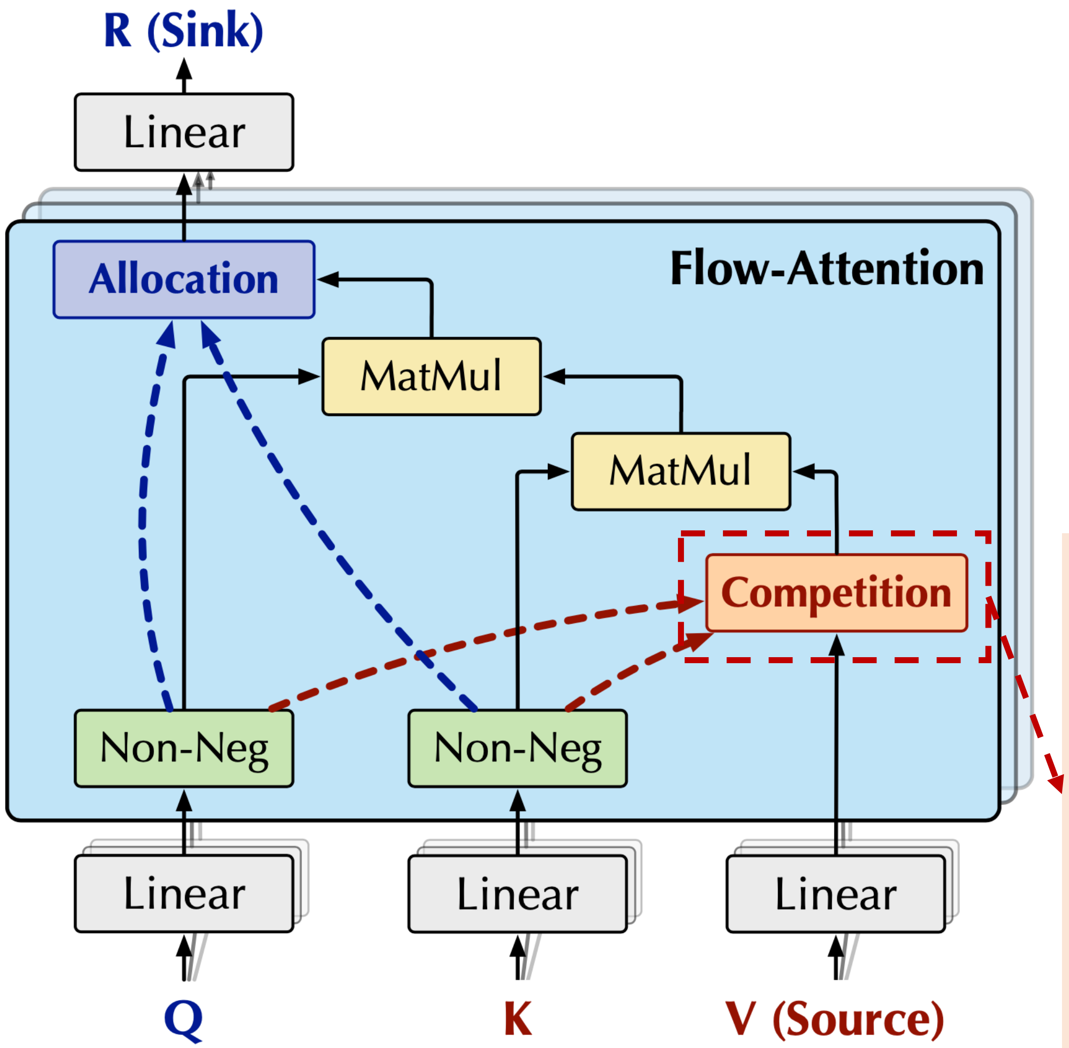
Haixu Wu, Jialong Wu, Jiehui Xu, Jianmin Wang, Mingsheng Long. Flowformer: Linearizing Transformers with Conservation Flows. **ICML**, 2022.

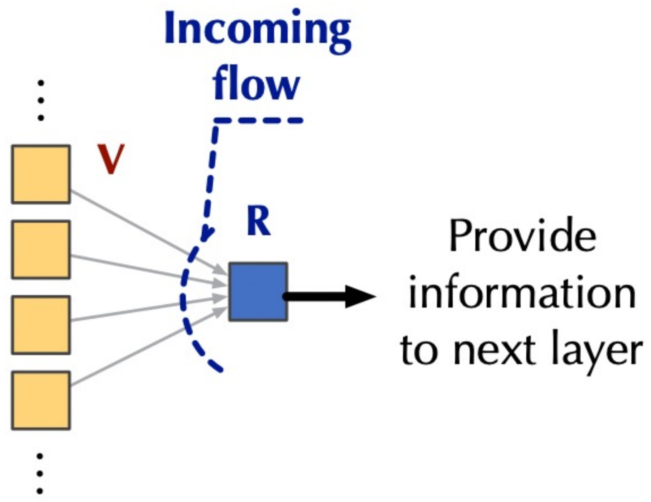
Flow-Attention



- [Incoming Flow Conservation]:
  - Competition among Source tokens
- [Outgoing Flow Conservation]:
  - Competition among Sink tokens

Flow-Attention



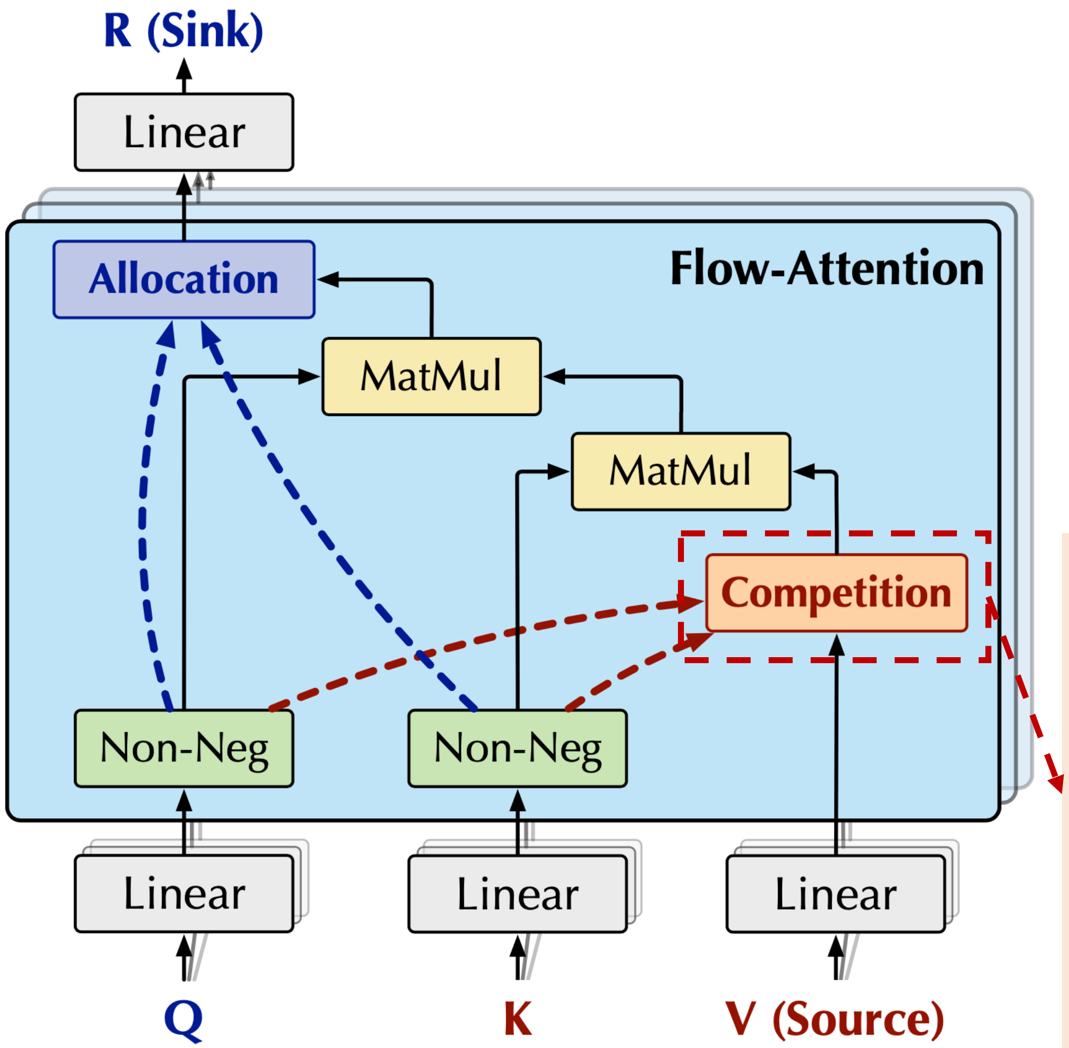


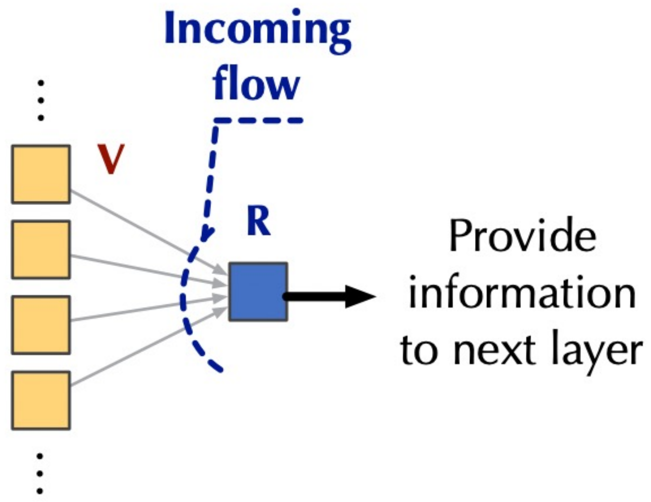
Incoming flow: $I_i = \phi(Q_i)\sum_j \phi(K_j)^T$

Incoming flow conservation: $\dfrac{\phi(\boldsymbol{Q})}{\boldsymbol{I}}$

Incoming flow: $\dfrac{\phi(Q_i)}{I_i}\sum_j \phi(K_j)^T = \dfrac{I_i}{I_i} = 1$

Flow-Attention



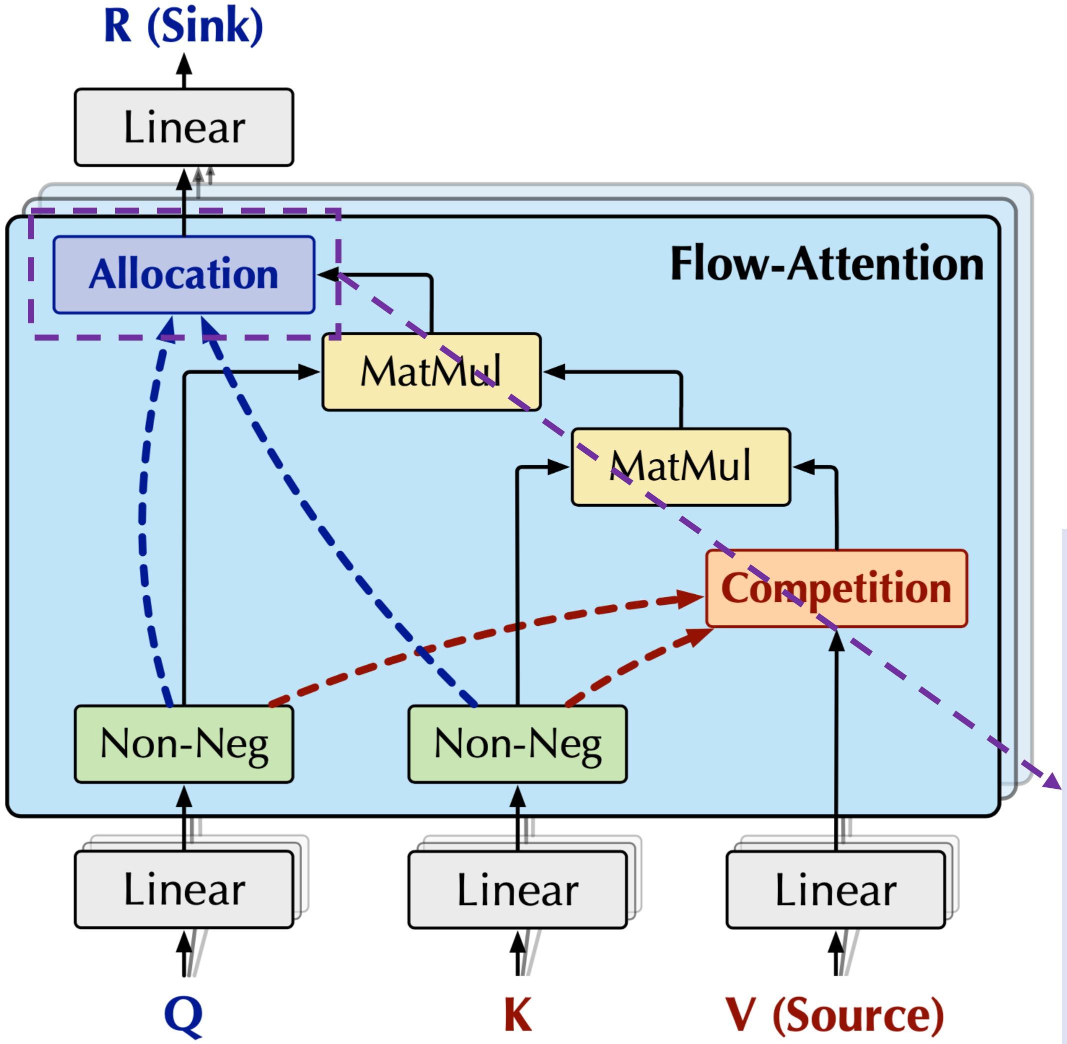


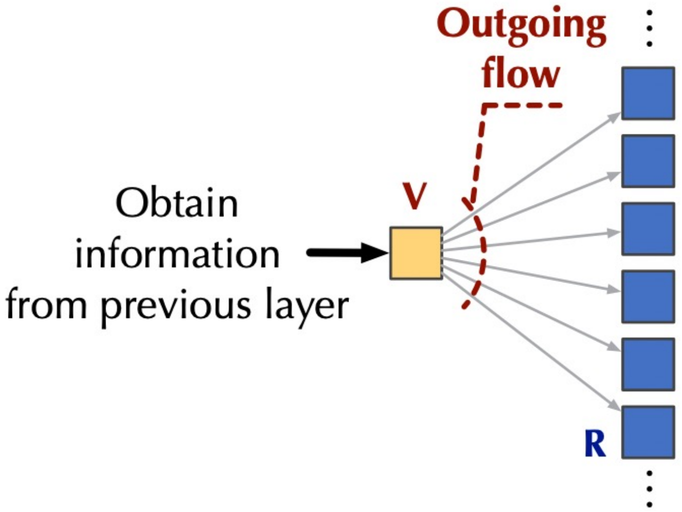
Incoming flow: $I_i = \phi(Q_i)\sum_j \phi(K_j)^T$

Incoming flow conservation: $\frac{\phi(\boldsymbol{Q})}{\boldsymbol{I}}$

Conserved outgoing flow: $\widehat{\boldsymbol{O}} = \phi(\boldsymbol{K})\sum_i \frac{\phi(Q_i)^T}{I_i}$

Flow-Attention



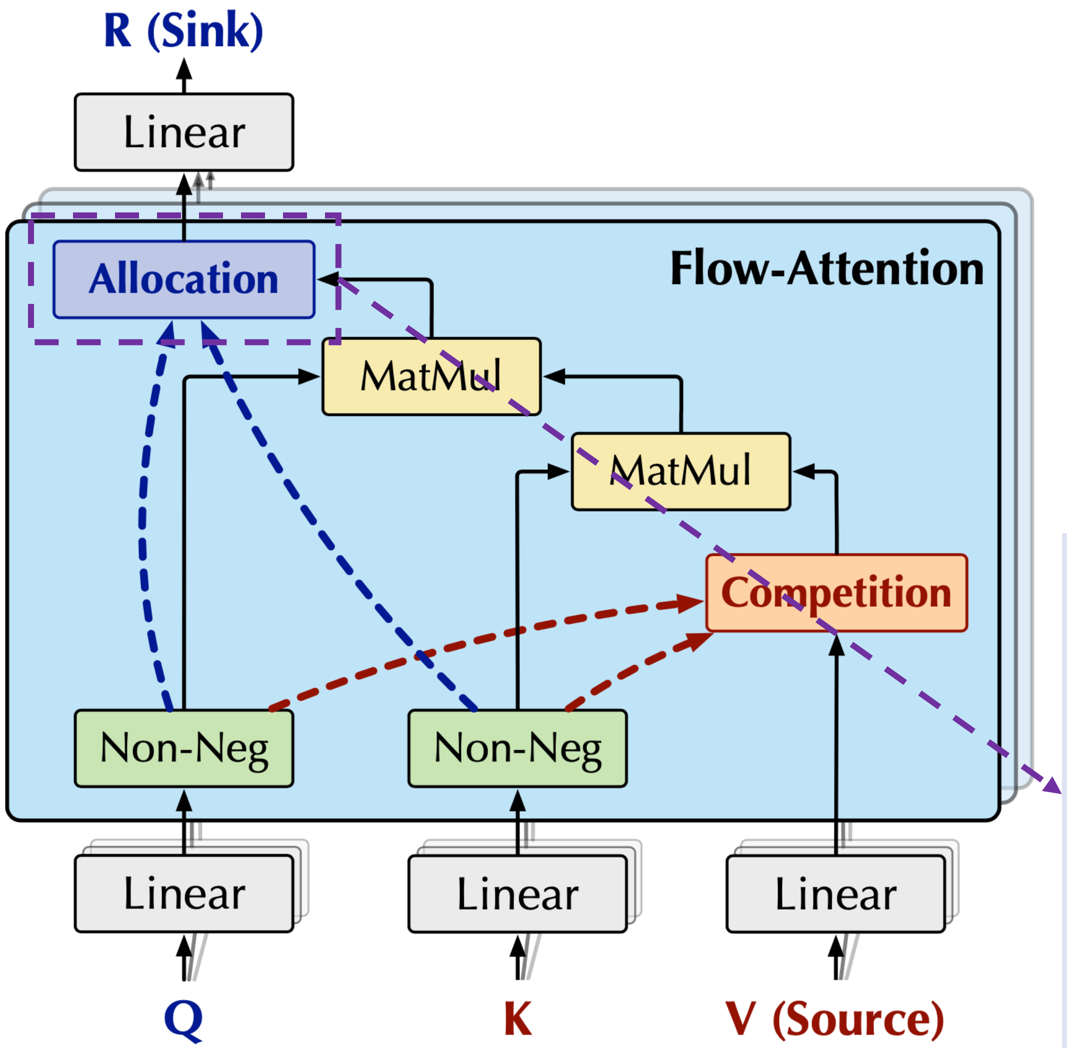


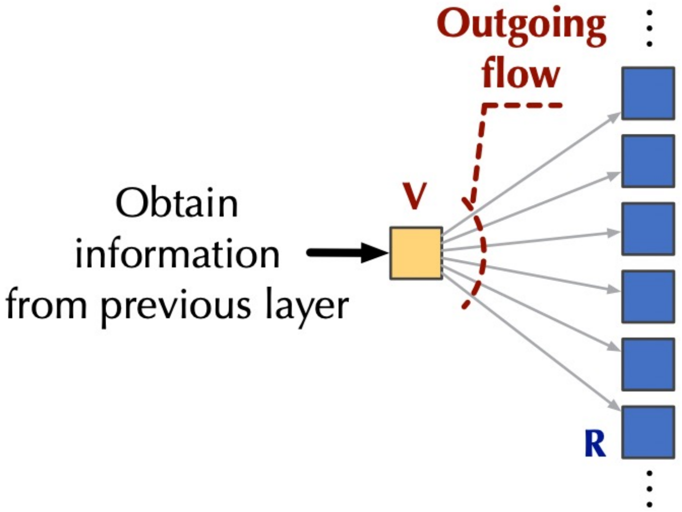
Outgoing flow: $O_i = \phi(K_i) \sum_j \phi(Q_j)^T$

Outgoing flow conservation: $\frac{\phi(\boldsymbol{K})}{\boldsymbol{O}}$

Outgoing flow: $\frac{\phi(K_i)}{O_i} \sum_j \phi(Q_j)^T = \frac{O_i}{O_i} = 1$

Flow-Attention



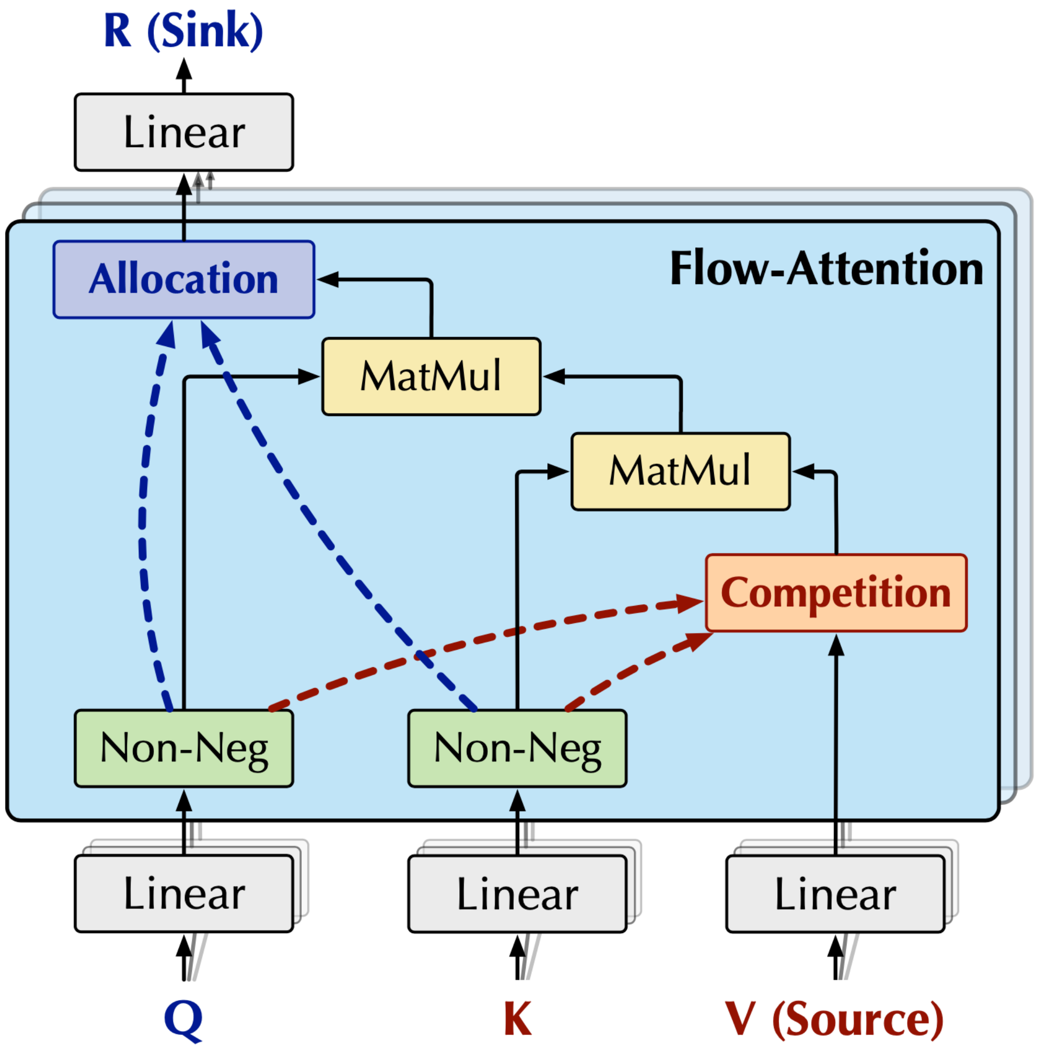


Outgoing flow: $O_i = \phi(K_i) \sum_j \phi(Q_j)^T$

Outgoing flow conservation: $\frac{\phi(\boldsymbol{K})}{\boldsymbol{O}}$

Conserved incoming flow: $\hat{\boldsymbol{I}} = \phi(\boldsymbol{Q}) \sum_j \frac{\phi(K_j)^T}{O_j}$

Flow-Attention



$$\text{Competition: } \widehat{\mathbf{V}} = \mathrm{Softmax}(\widehat{\mathbf{O}}) \odot \mathbf{V}$$

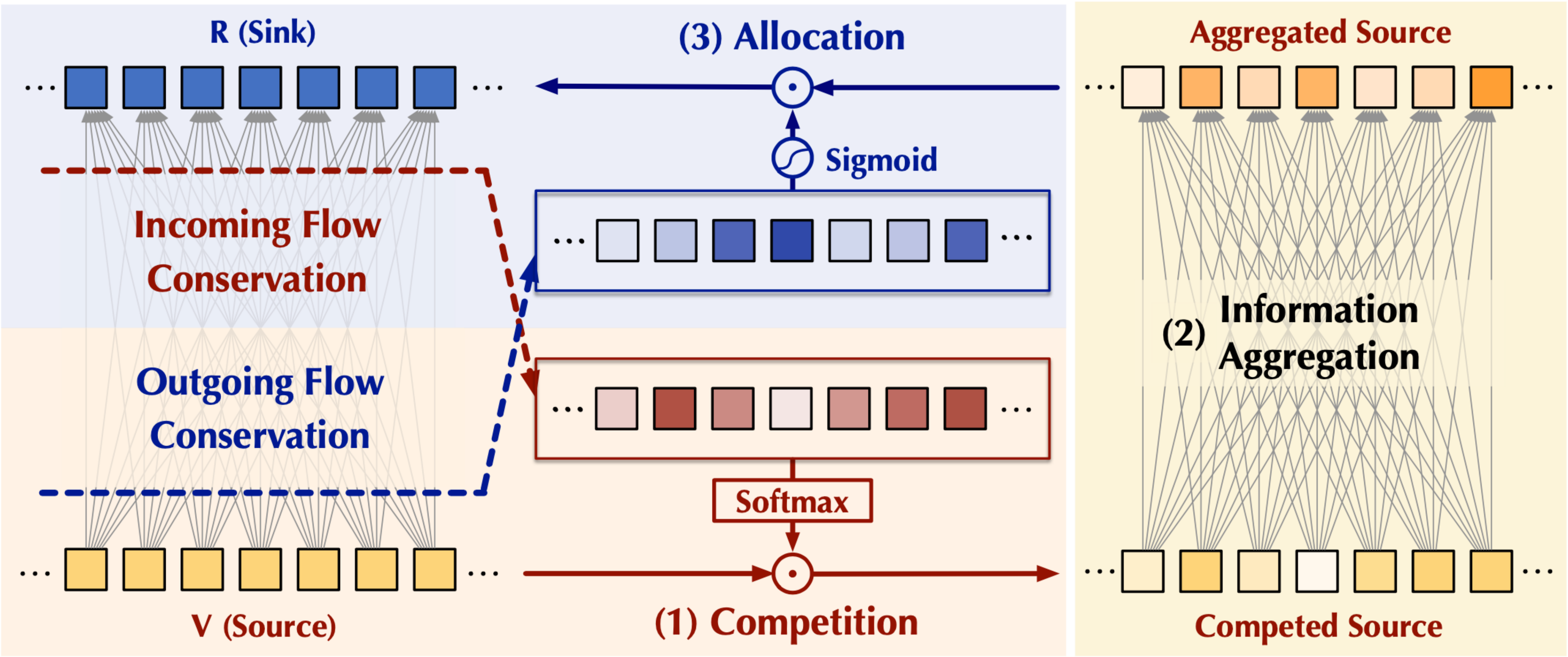
$$\text{Aggregation: } \mathbf{A} = \frac{\phi(\mathbf{Q})}{\mathbf{I}}(\phi(\mathbf{K})^{\mathsf{T}}\widehat{\mathbf{V}})$$

$$\text{Allocation: } \mathbf{R} = \mathrm{Sigmoid}(\widehat{\mathbf{I}}) \odot \mathbf{A},$$

Successfully bring the **Competition Mechanism**

Into Attention design to avoid trivial attention

Flowformer: Efficiency and Universality



[Efficiency]: All the calculations are in linear complexity.

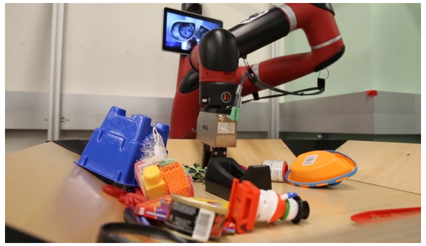
[Universality]: The whole design is based on flow network without specific inductive biases.

Flowformer Experiments

Image

Language

Time Series



Agent Trajectory

Benchmarks	Task	Version	Length
LRA (2020c)	Sequence	normal	1000∼4000
WikiText (2017)	Language	causal	512
ImageNet (2009)	Vision	normal	49∼3136
UEA (2018)	Time Series	normal	29∼1751
D4RL (2020)	Offline RL	causal	60

- Extensive tasks (covering 5 mainstream tasks)
- Normal and causal versions
- Various sequence lengths (29-4000)
- Extensive baselines (20+)

Flowformer Experiments

Task	Metrics	Flowformer	Performer	Reformer	Vanilla Transformer
Long Sequence Modeling (LRA)	Avg Acc (%) ↑	**56.48**	51.41	50.67	OOM
Vision Recognization (ImageNet-1K)	Top-1 Acc (%) ↑	**80.6**	78.1	79.6	78.7
Language Modeling (WikiText-103)	Perplexity ↓	**30.8**	37.5	33.6	33.0
Time series classification (UEA)	Avg Acc (%) ↑	**73.0**	71.5	71.9	71.9
Offline RL (D4RL)	Avg Reward ↑ Avg Deviation ↓	**73.5** $\pm$ 2.9	63.8 $\pm$ 7.6	63.9 $\pm$ 2.9	72.2 $\pm$ **2.6**

Strong performance on all five mainstream tasks within the linear complexity.

Flowformer

General Relation Modeling

Quadratic Complexity

Long Sequence
Model Efficiency
Big Model ☹

Task & Data Universal

Foundation Model

Flowformer

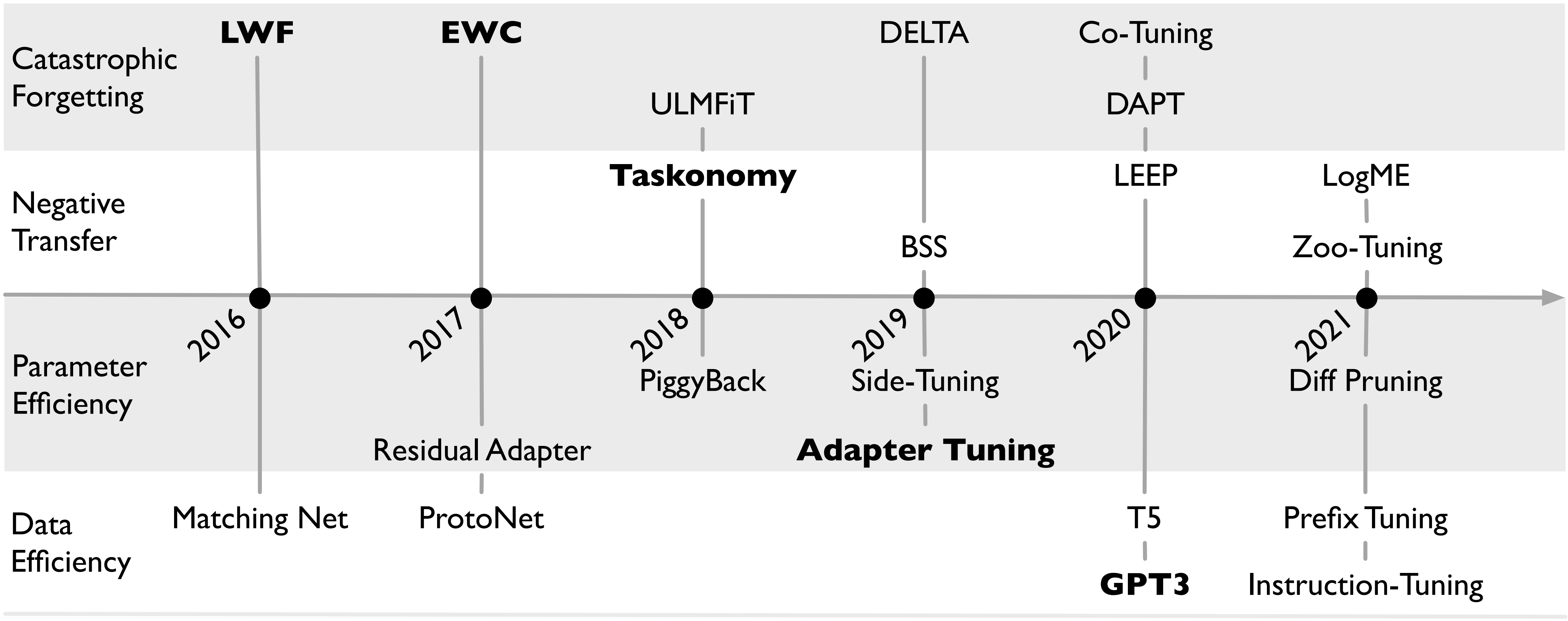
Linear complexity w.r.t. sequence length

Based on flow network & without specific inductive biases

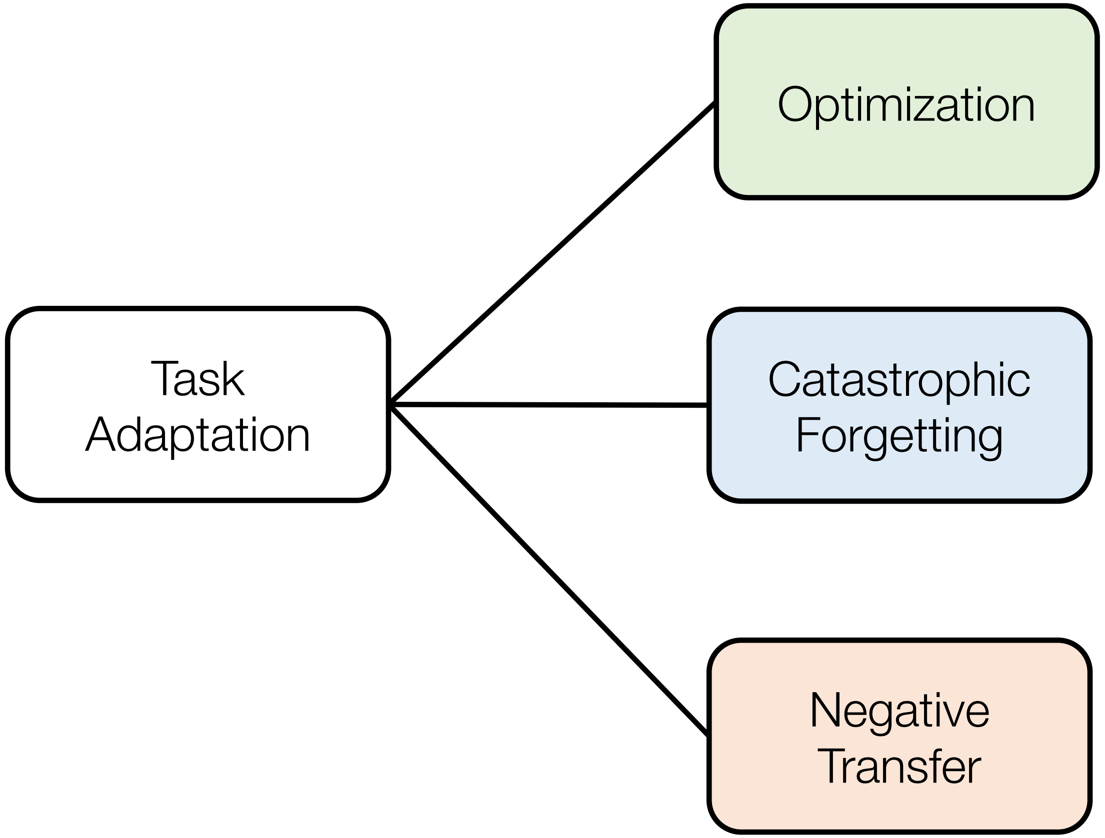
Strong performance in Long Sequence, CV, NLP, Time Series, RL

Haixu Wu, Jialong Wu, Jiehui Xu, Jianmin Wang, Mingsheng Long✉. Flowformer: Linearizing Transformers with Conservation Flows. **ICML**, 2022.

Task Adaptation

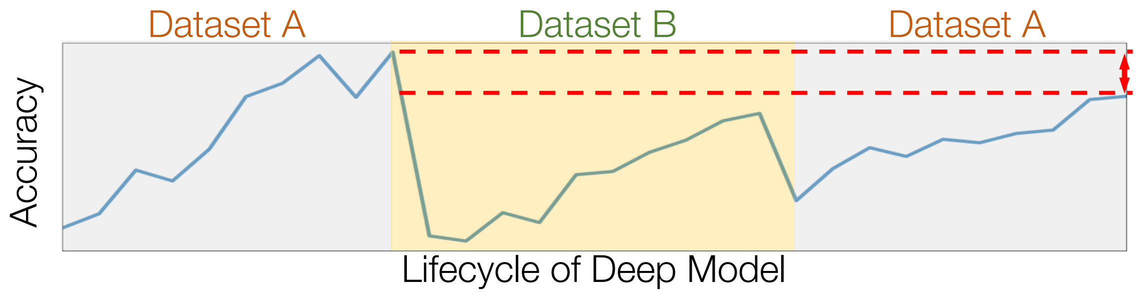


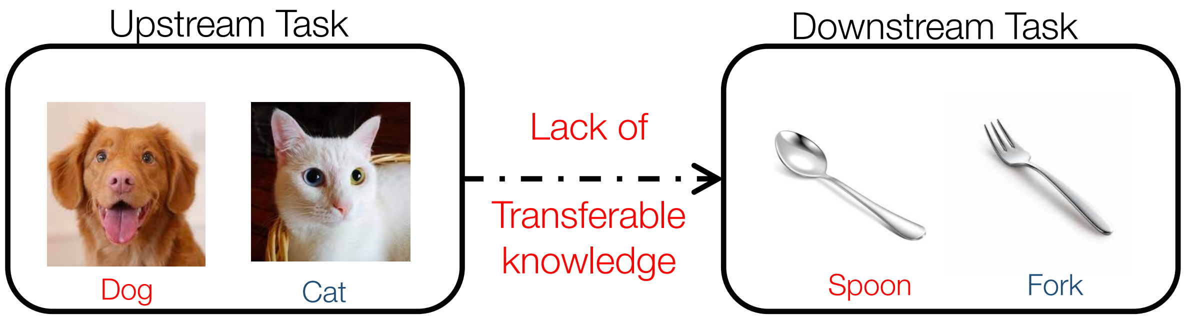
Foundation Problems

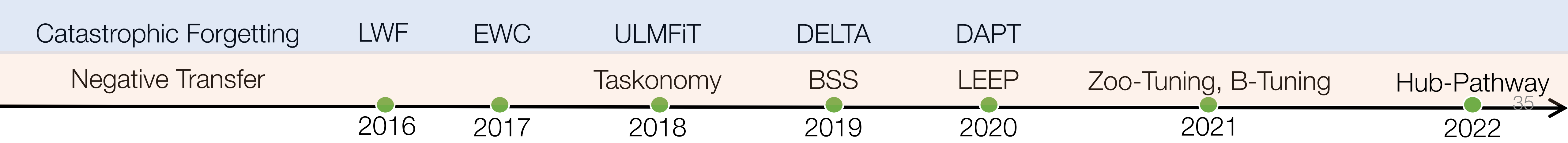


Training Strategies

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





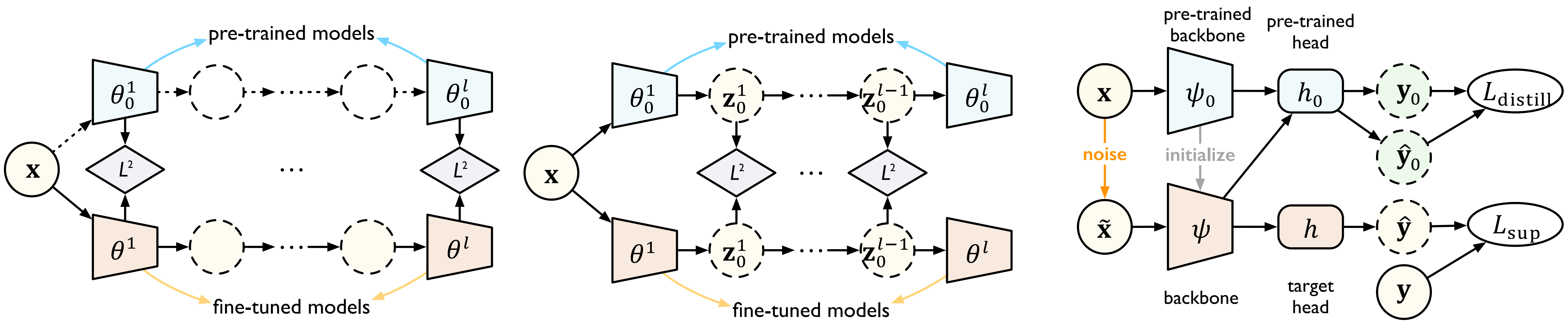


Catastrophic Forgetting

Regularization Tuning

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

Regularization term



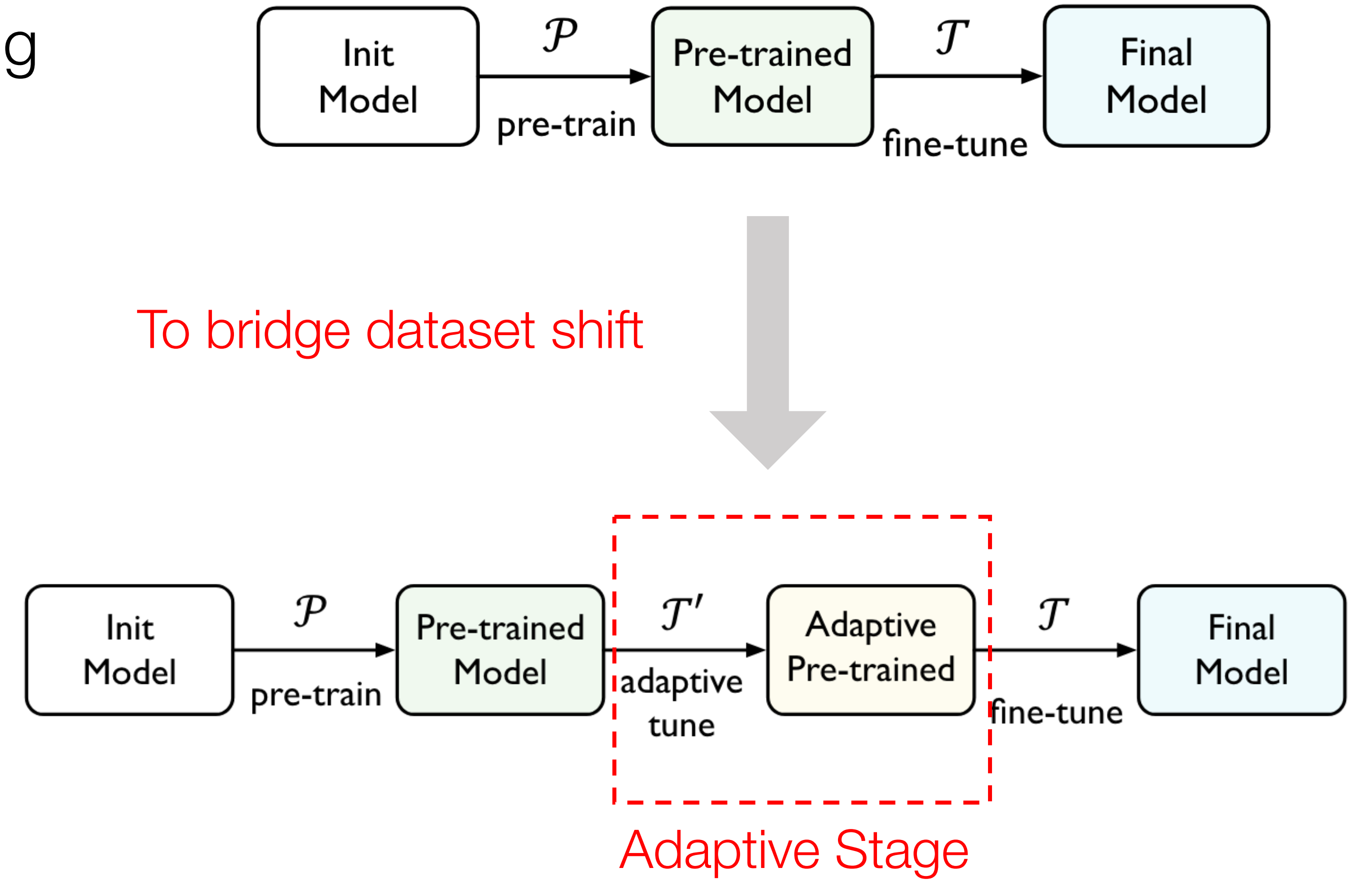
(a) EWC (b) DELTA (c) LWF

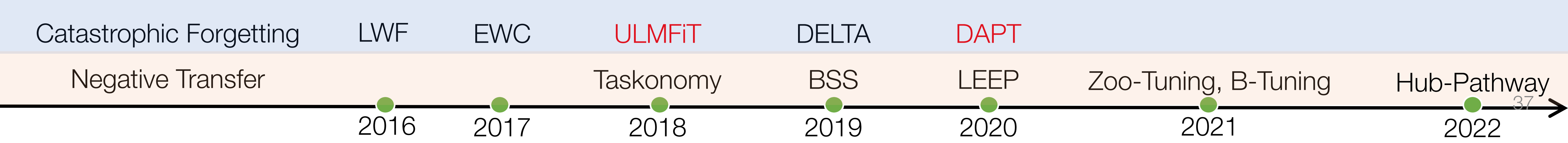
	2016	2017	2018	2019	2020	2021	2022
Catastrophic Forgetting	LWF	EWC	ULMFiT	DELTA	DAPT		
Negative Transfer			Taskonomy	BSS	LEEP	Zoo-Tuning, B-Tuning	Hub-Pathway

Catastrophic Forgetting

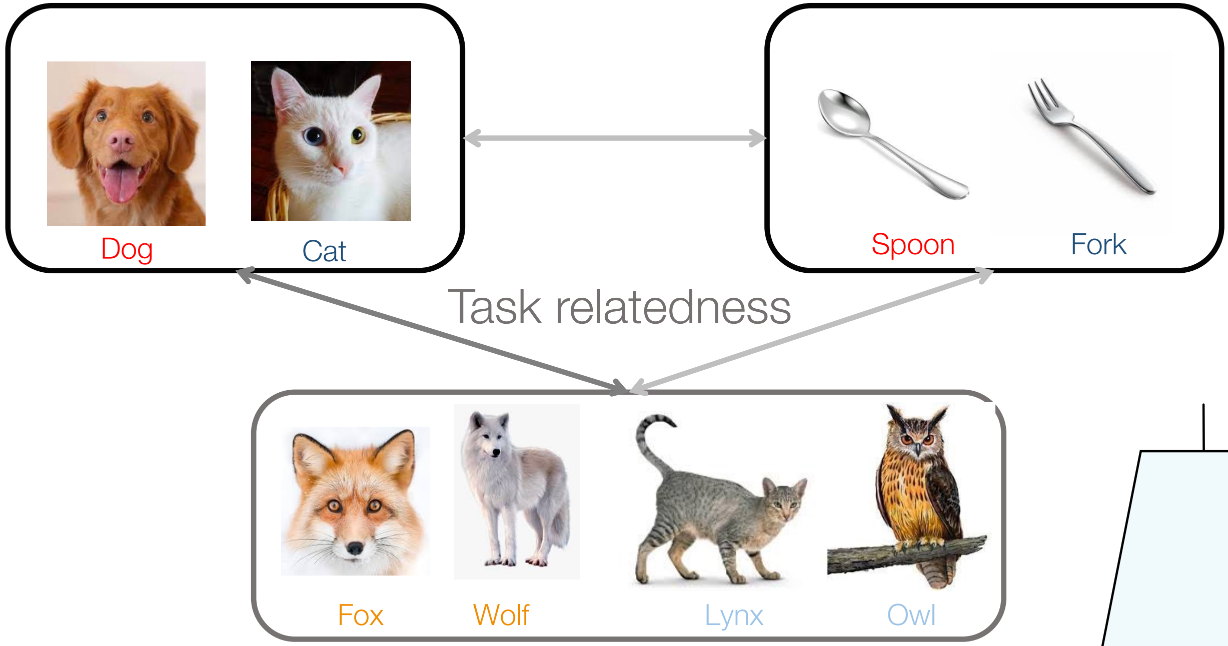
Domain Adaptive Tuning

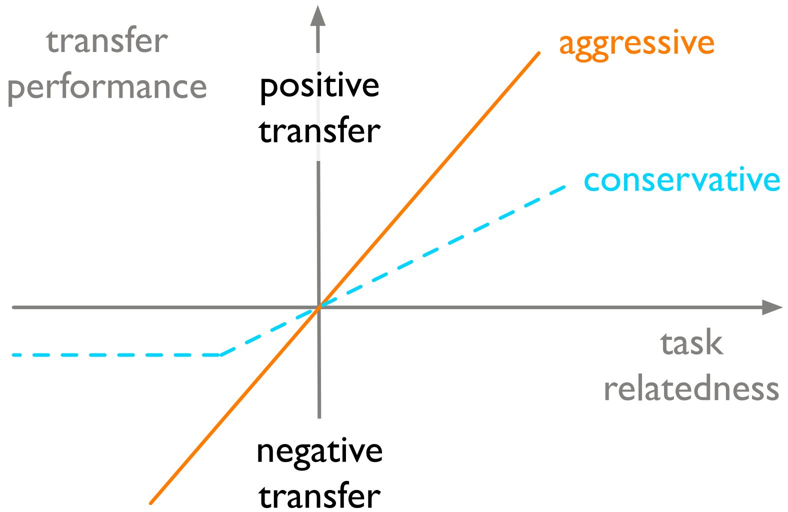
- ULMFiT
- DAPT
- SiATL

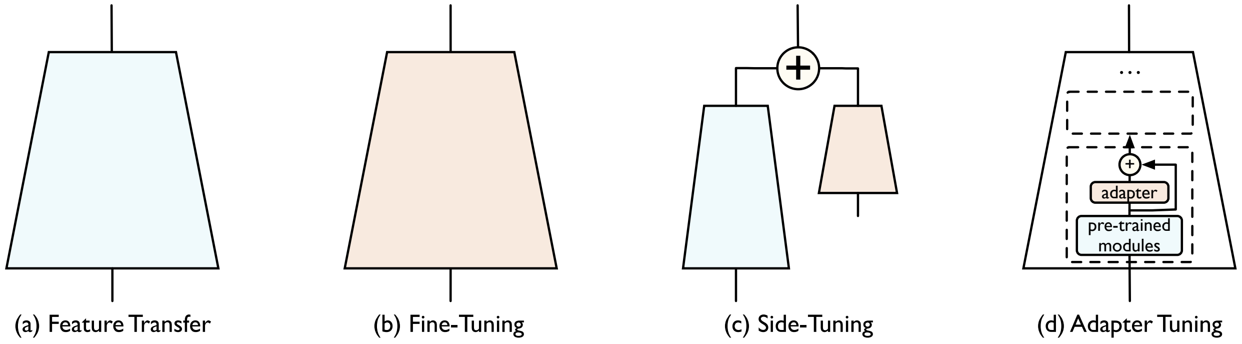


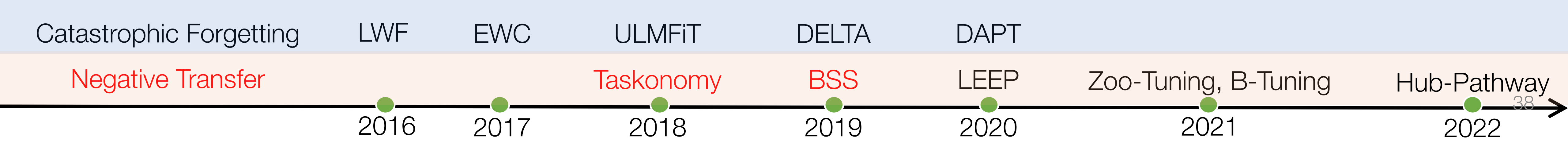


Negative Transfer





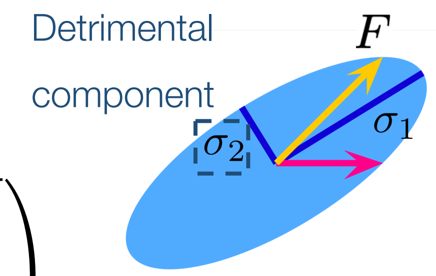


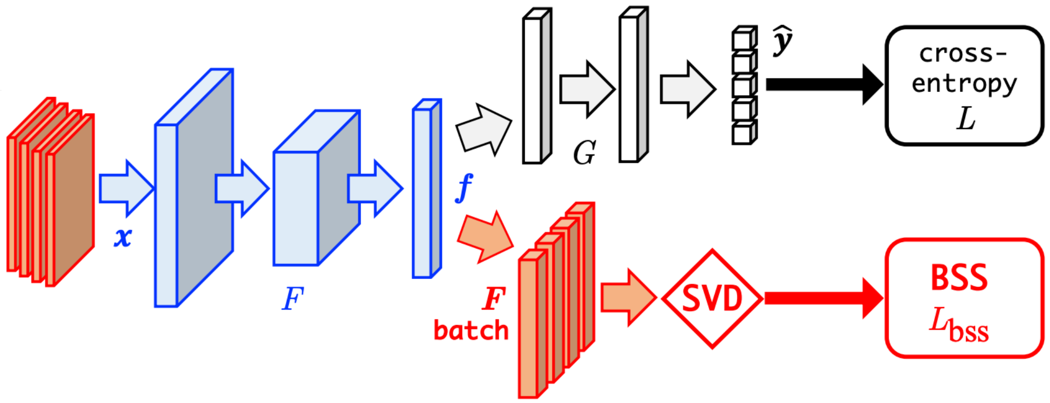


Negative Transfer

- Enhance Safe Transfer
  - BSS, Zoo-tuning

Feature Space





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

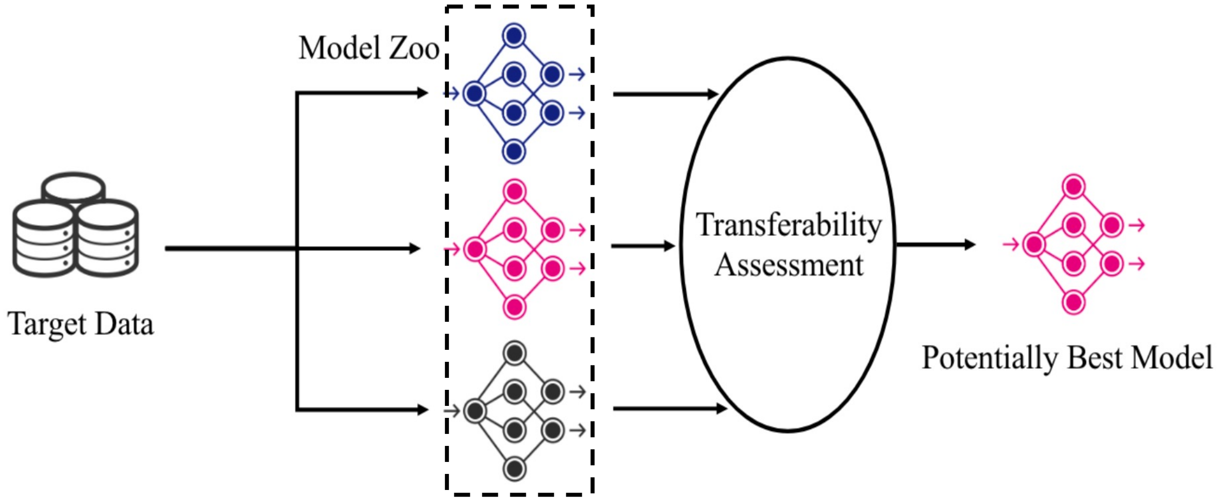
Penalize smallest singular values : $L_{\text{bss}}(F) = \eta \sum_{i=1}^{k} \sigma_{-i}^2$

- Choose Pre-trained Models
  - LEEP, LogME

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

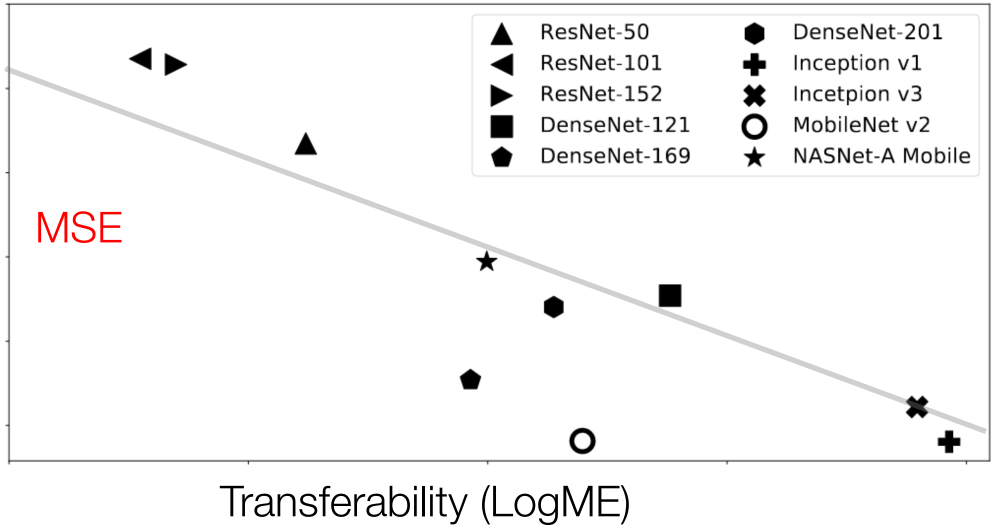
Negative Transfer

- Enhance Safe Transfer
  - BSS, Zoo-tuning



Transferable knowledge pool

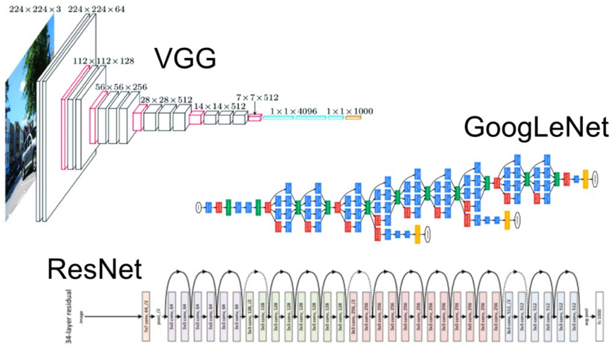
- Choose Pre-trained Models
  - LEEP, LogME



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

Pre-trained Model Hub

Various Models and Platforms



PyTorch

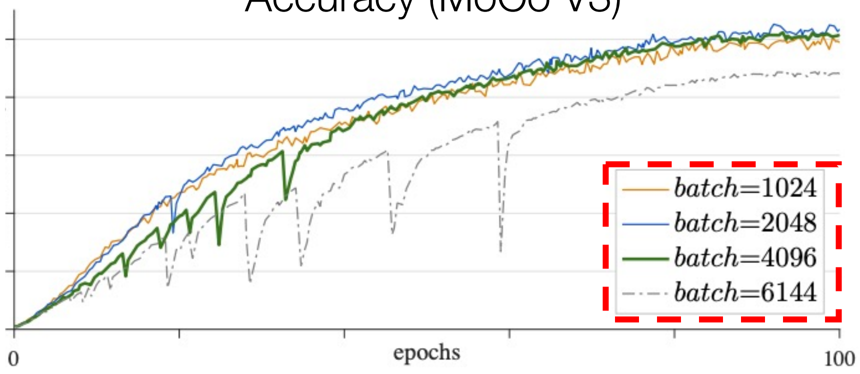
Hugging Face

TensorFlow Hub

Pytorch Image Models

Avoid Heavy Pre-training

Accuracy (MoCo V3)



Plenty of Transferable Knowledge

ImageNet sup.
MoCo pt.
MaskRCNN pt.
DeepLab pt.
Keypoint pt.

ResNet

Same architecture

Pre-trained differently

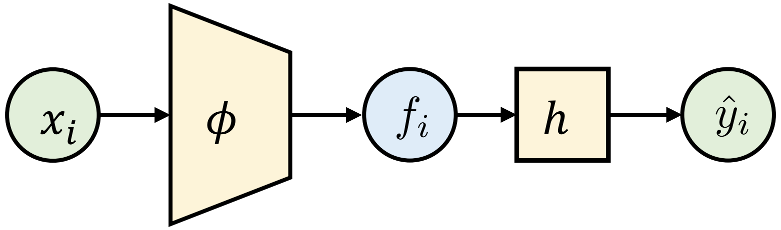
- Adapt one model
  - Which one is the best?

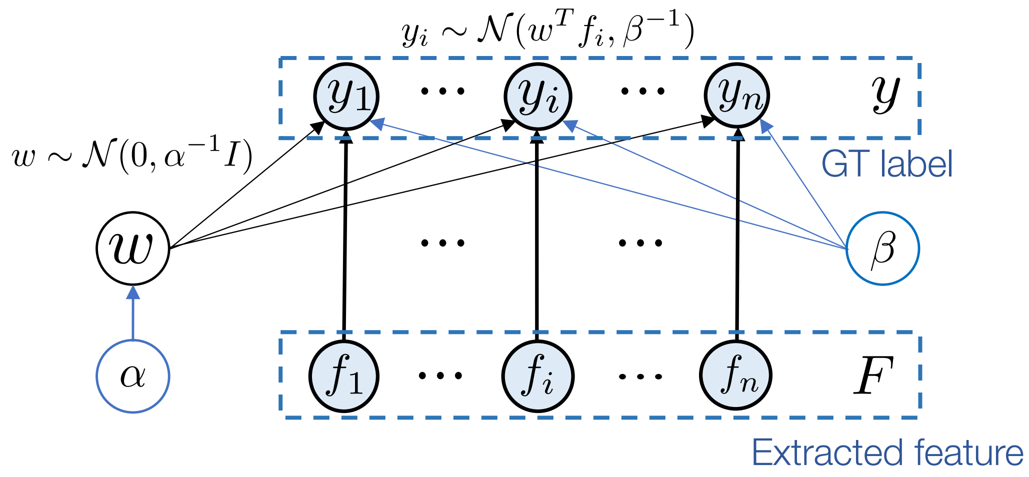
- Adapt multiple models
  - How to aggregate transferable knowledge?

Transferability Assessment by LogME

Estimate adaption performance of PTM on given dataset without finetuning.

LogME Approach



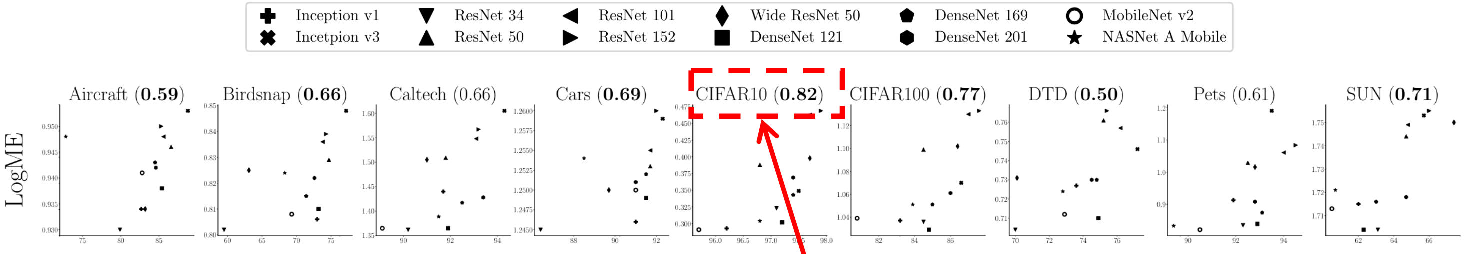


- Fixed PTM (as feature extractor).
- $P(y \mid F)$: Graphical modeling between extracted features and GT label.
- Parameterize $P(y \mid F)$ by prior α, β.
- Maximize evidence $P(y \mid F, \alpha, \beta)$.
  - MacKay algorithm with guarantee!

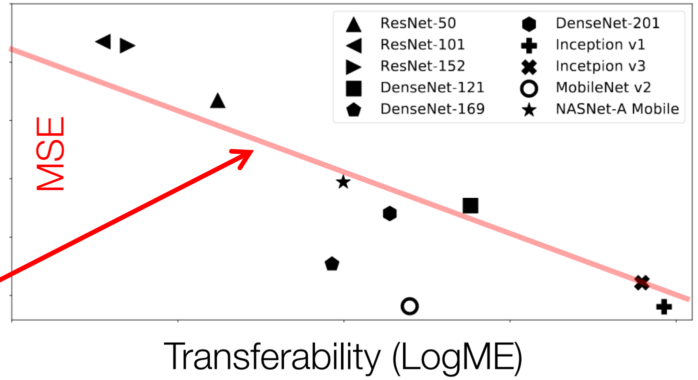
Effectiveness of LogME

General and Accurate

Vision tasks

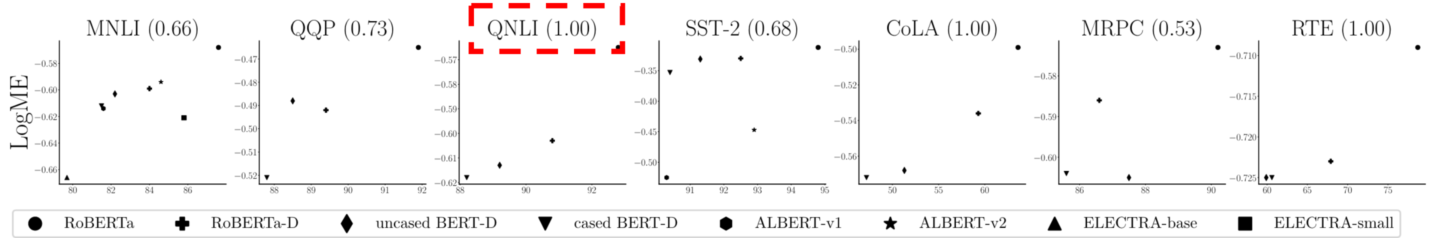


Regression task



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
	τ_w: 1.0		τ_w: 1.0	

Theoretical Guarantee of LogME

- MacKay algorithm (1992) is a heuristic method for solving the evidence maximization procedure of empirical Bayesian learning (Bishop, 1995).
- We provide the theoretical guarantee for MacKay algorithm.

Algorithm 4 One iteration of evidence maximization in Algorithm 2.

1: Input: α, β; Output: α', β' for the next iteration.
2: Compute $A = \alpha I + \beta F^T F, m = \beta A^{-1} F^T y, \gamma = \sum_{i=1}^{D} \frac{\beta \sigma_i^2}{\alpha + \beta \sigma_i^2}$
3: Return $\alpha' = \frac{\gamma}{m^T m}, \beta' = \frac{n - \gamma}{||Fm - y||_2^2}$

Theorem 1 Algorithm 4 induces a scalar function (Equation 3) with $t = \frac{\alpha}{\beta}$ and $t' = \frac{\alpha'}{\beta'}$.

$$t' = f(t) = \left(\frac{n}{n - \sum_{i=1}^{D} \frac{\sigma_i^2}{t + \sigma_i^2}} - 1 \right) t^2 \frac{\sum_{i=1}^{n} \frac{z_i^2}{(t + \sigma_i^2)^2}}{\sum_{i=1}^{n} \frac{\sigma_i^2 z_i^2}{(t + \sigma_i^2)^2}}. \tag{3}$$

Theorem 2 If $r < n$ and $\sum_{1 \le i,j \le n} (z_i^2 - z_j^2)(\sigma_i^2 - \sigma_j^2) > 0$, then $f(t)$ has a fixed point and thus MacKay's algorithm will converge.

Efficiency of LogME

Computation Efficient --- MacKay algorithm with improved 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 α, β 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 α, β 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 $\Sigma'_{ii} = \frac{\sigma_i}{t+\sigma_i^2} (1 \leq i \leq r)$.
Compute and return $\mathcal{L} = \frac{1}{n}\mathcal{L}(\alpha, \beta)$ using Equation 2

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

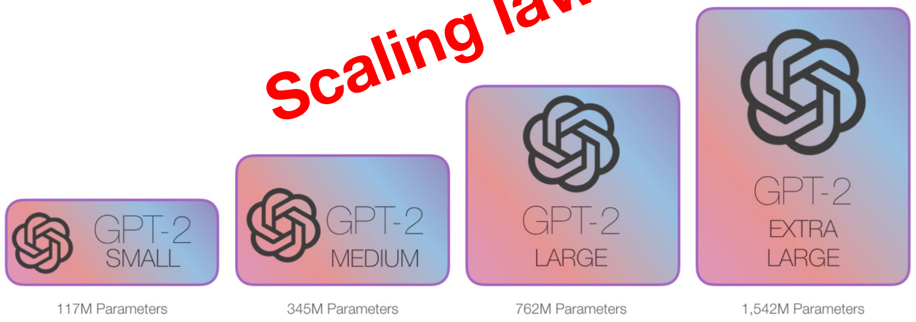
Cubic complexity with fewer terms

	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 ↑

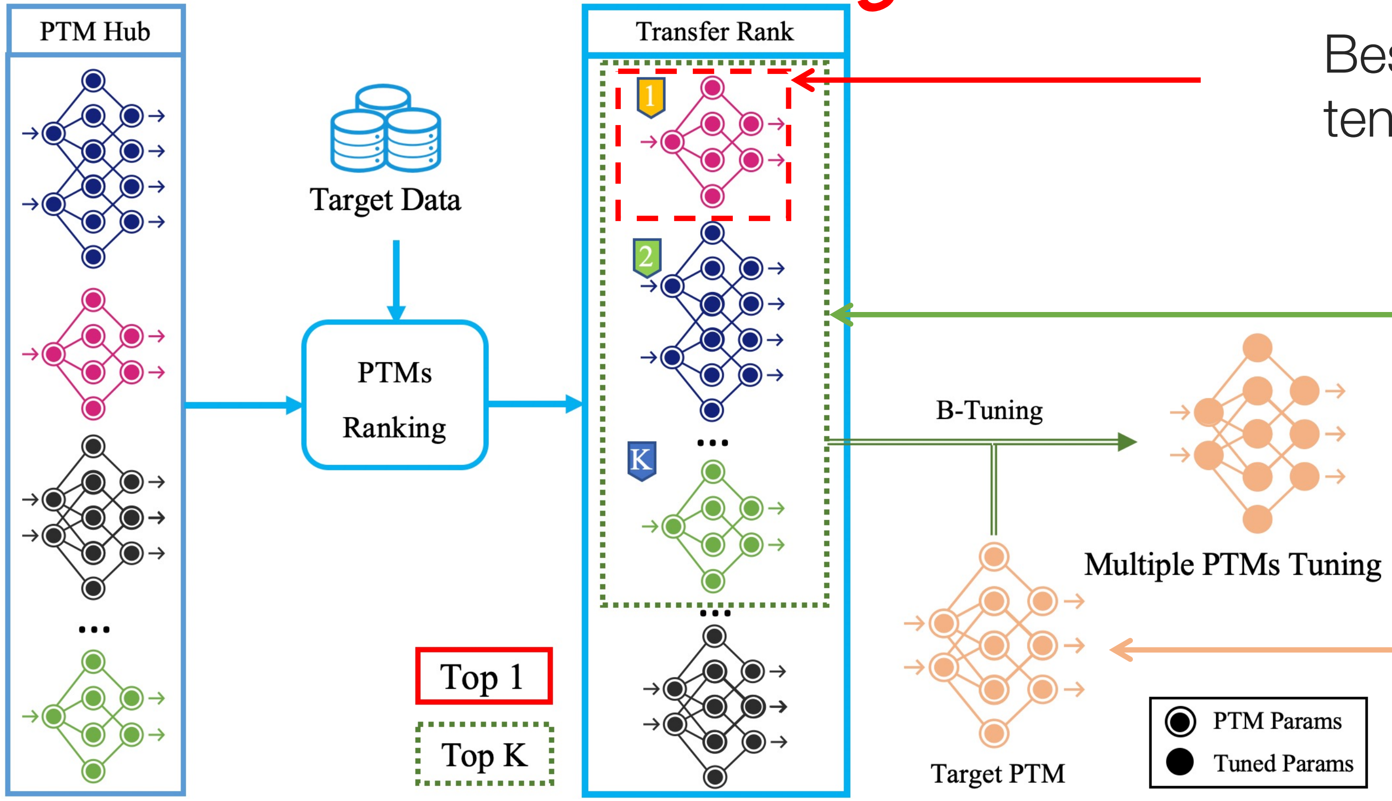
Tuning Pre-trained Models

Why adapt multiple models?

Scaling law I



Scaling law II

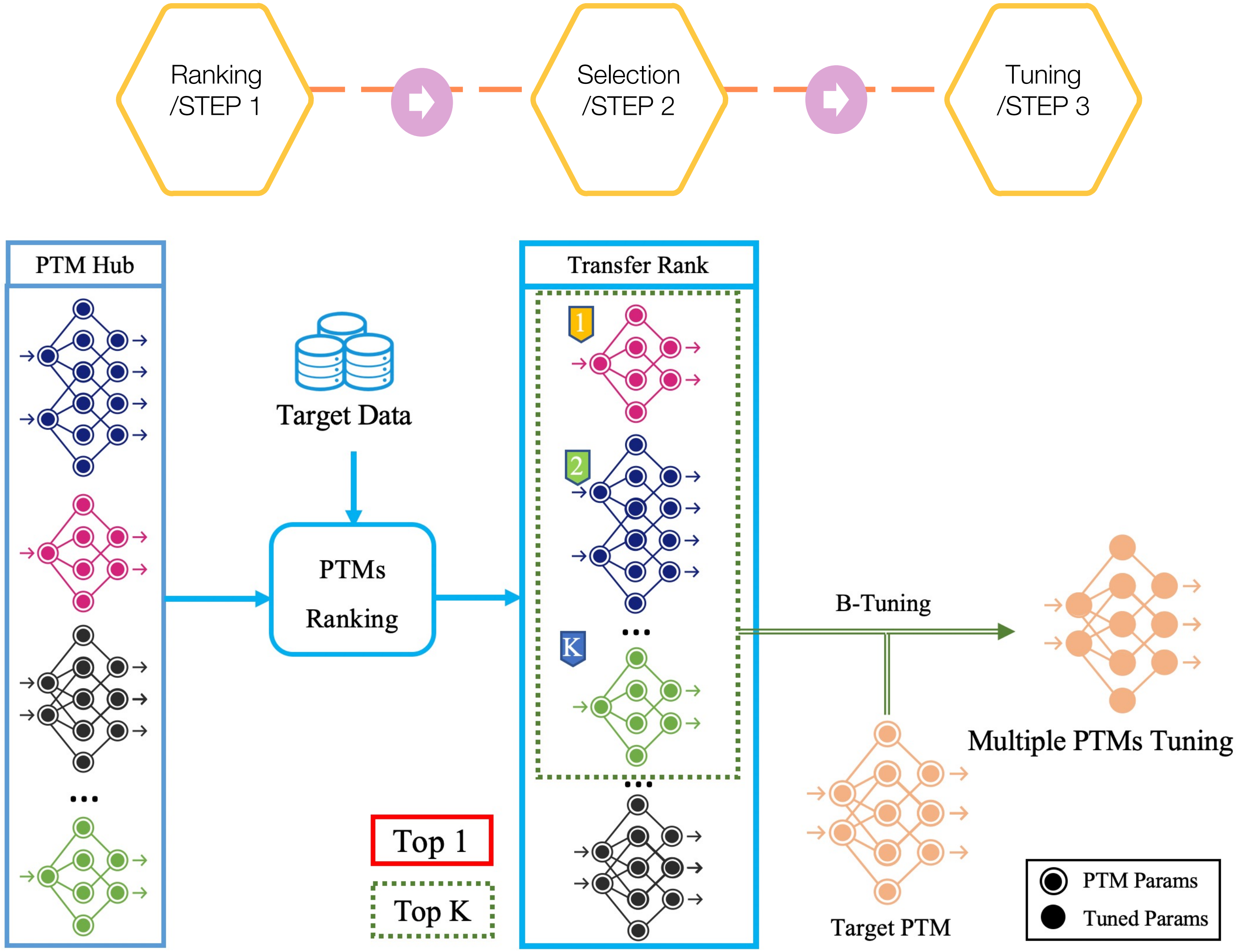


Best ranked models (e.g. DenseNet-201) tend to be large and expensive to deploy.

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

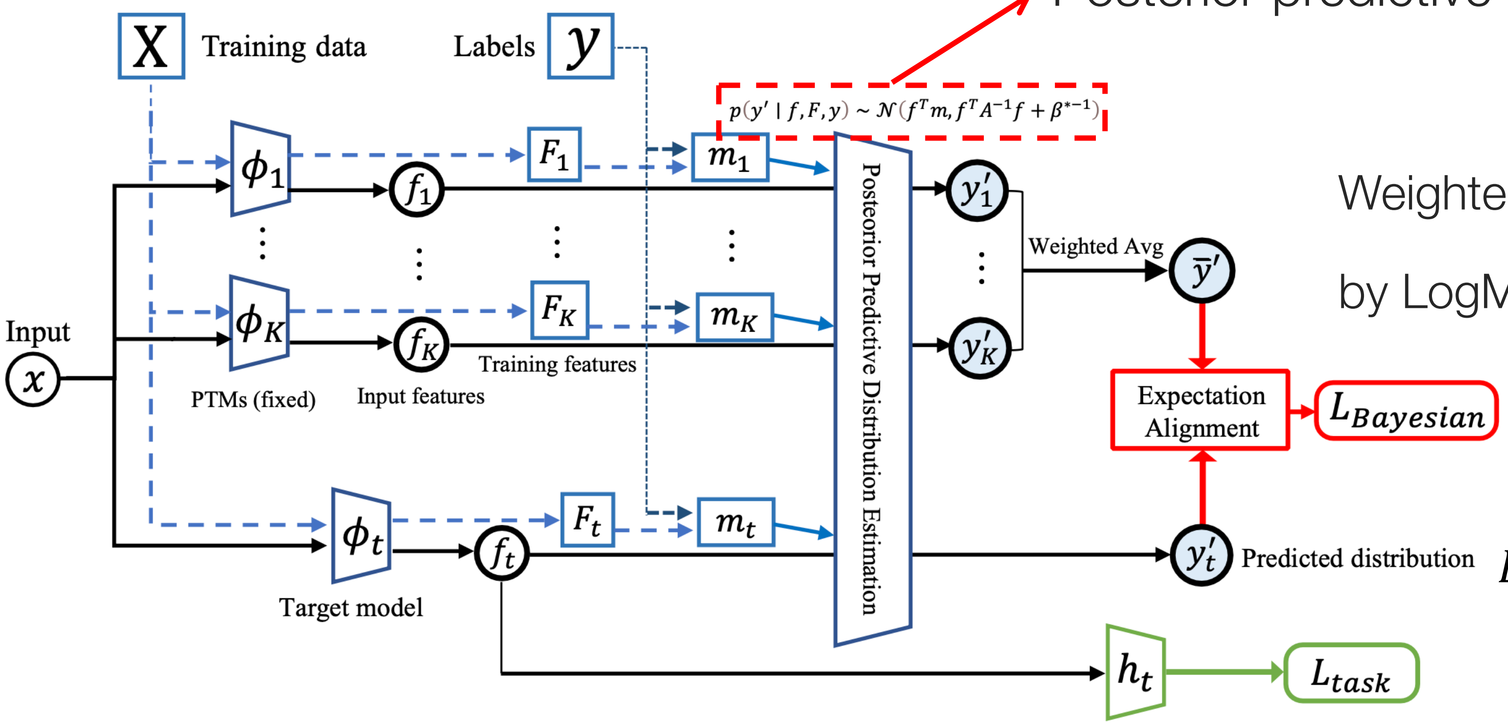
Consider simple Knowledge Distillation (KD):

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

- Needs additional learnable projection W_k for each teacher model.
- Treats all teacher models as equal:
  - No adaptive mechanism to transfer only useful knowledge.
- Violates the "Many could be better than all" theorem (Zhou et al. 2002).

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



Posterior predictive distributions $p(y'_k \mid f_k, F_k, y)$ from LogME

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

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

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

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

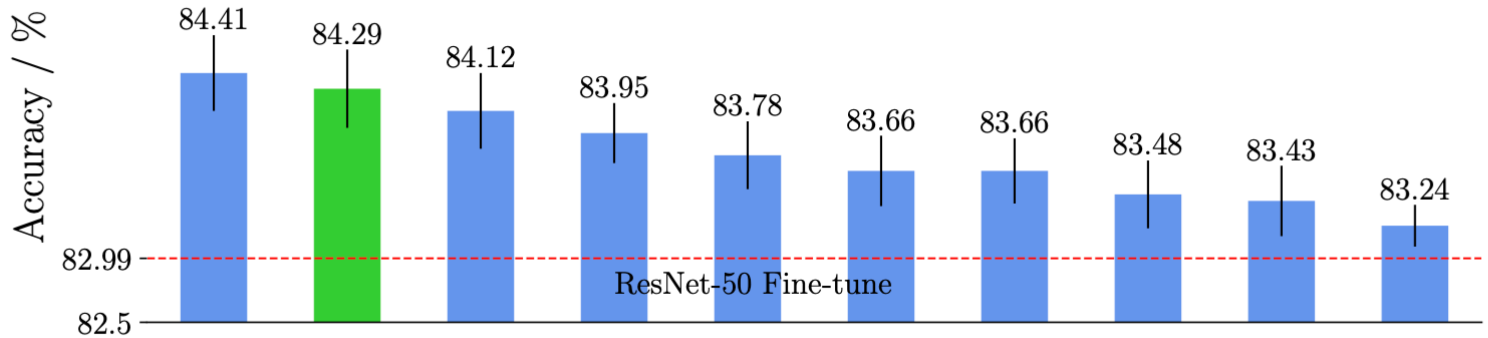
Intuition: encourage the target model to behave like the best top-K teachers.

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

Effectiveness of B-Tuning

Reduced burden of Selection and Adaptation.

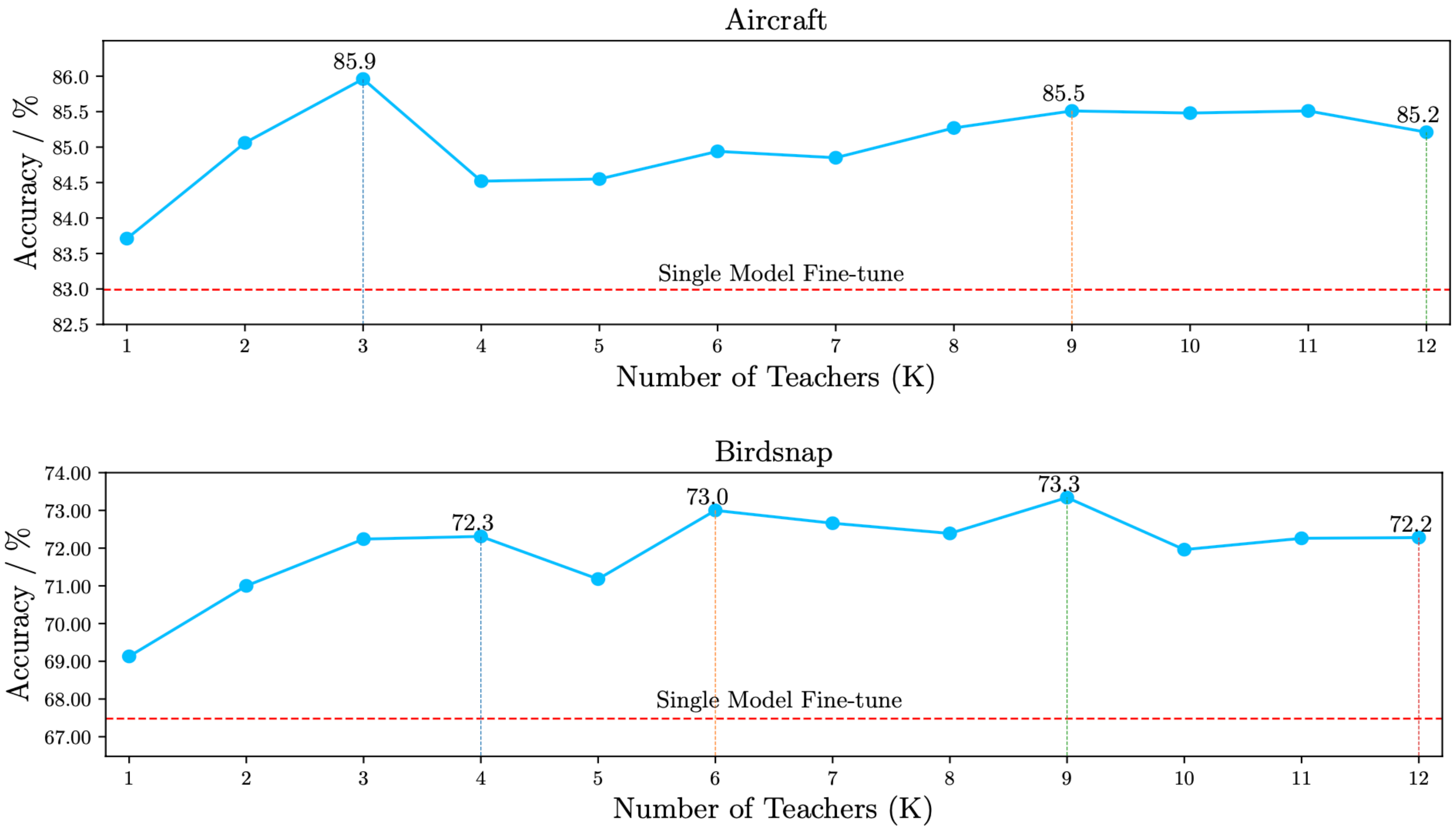
- Exhaustively fine-tune 10 times: 84.41% accuracy.
- Rank by LogME and fine-tune once: 84.29% accuracy.



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

Effectiveness of B-Tuning

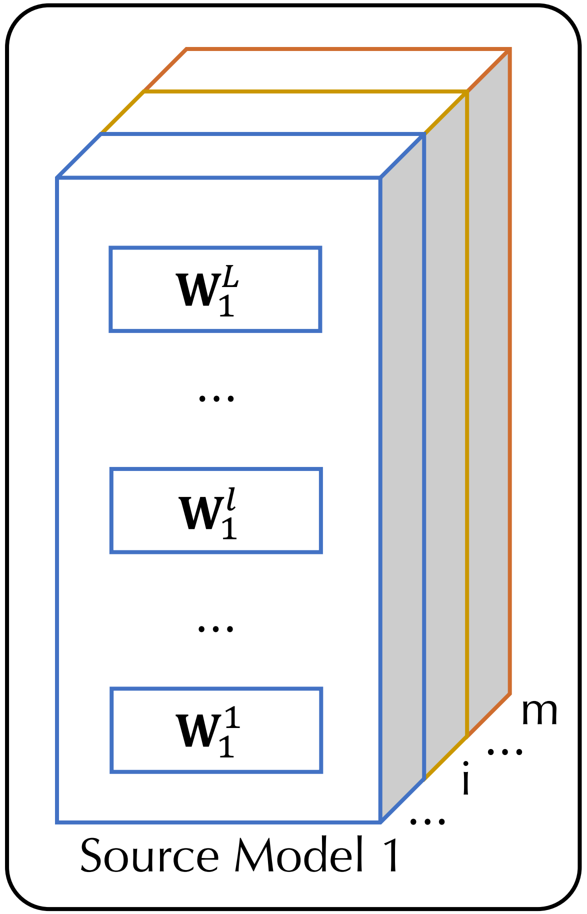
Fully utilization of transferable knowledge in Model Hub.

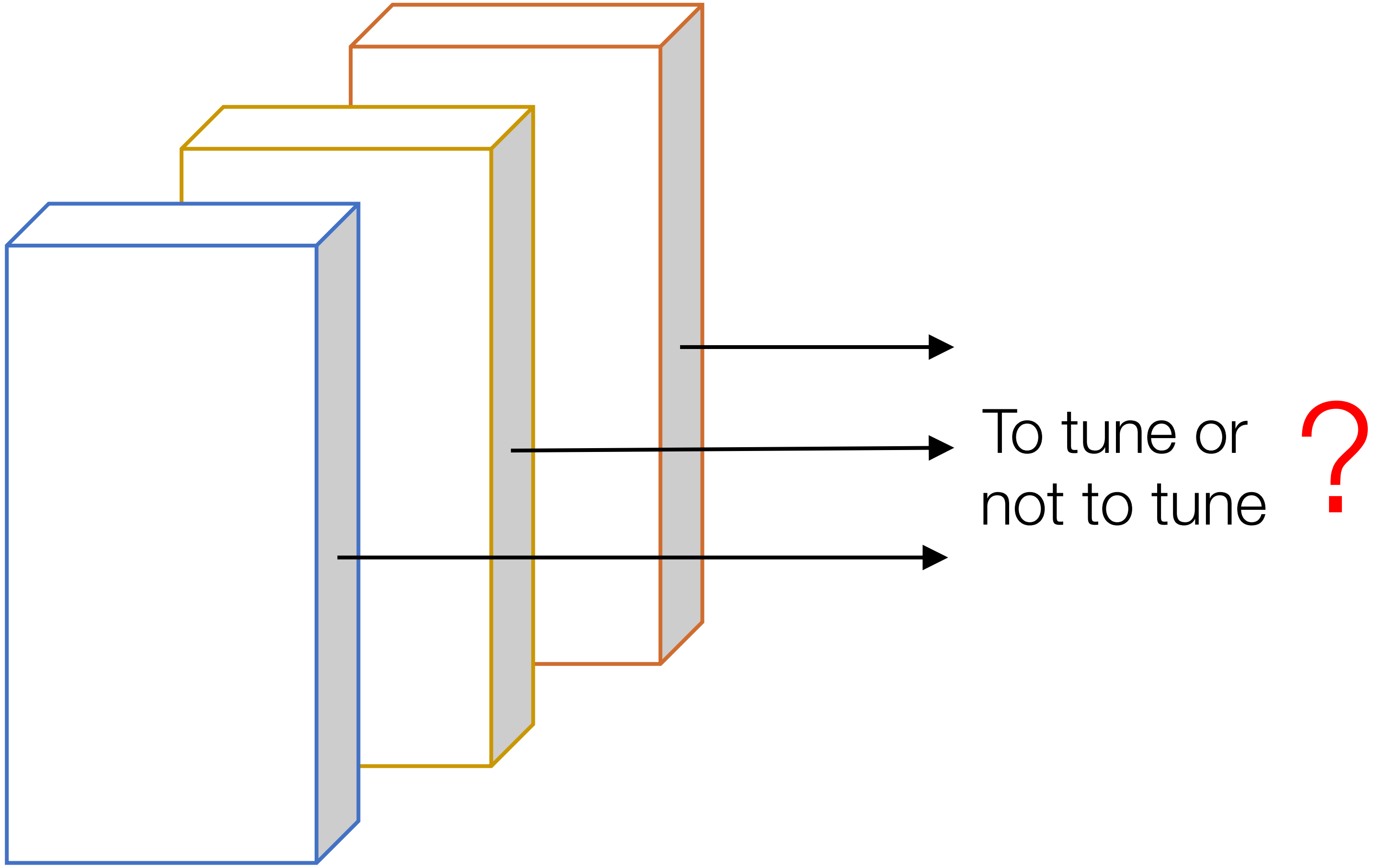


- Just fine-tune the most popular model is sub-optimal.
- The ranking and B-Tuning paradigm brings 3%~5% accuracy gain.

Homogeneous Model Zoo

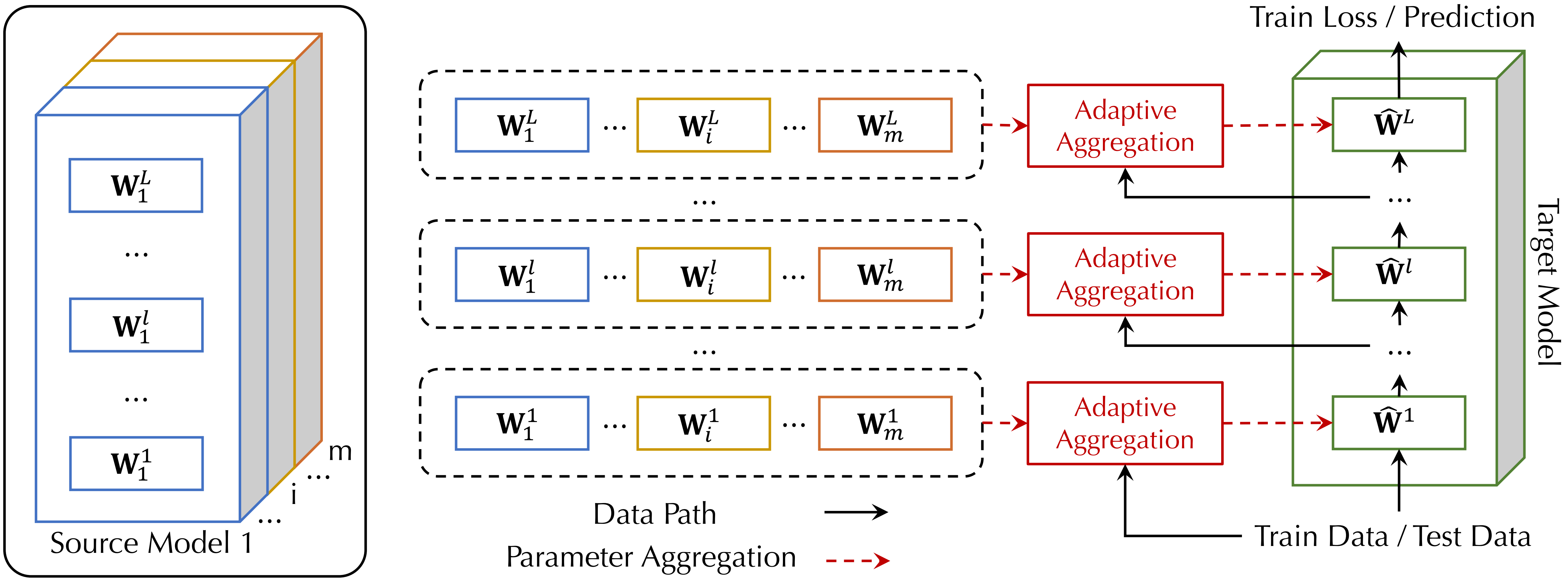
Considering models with same architecture but different knowledge.





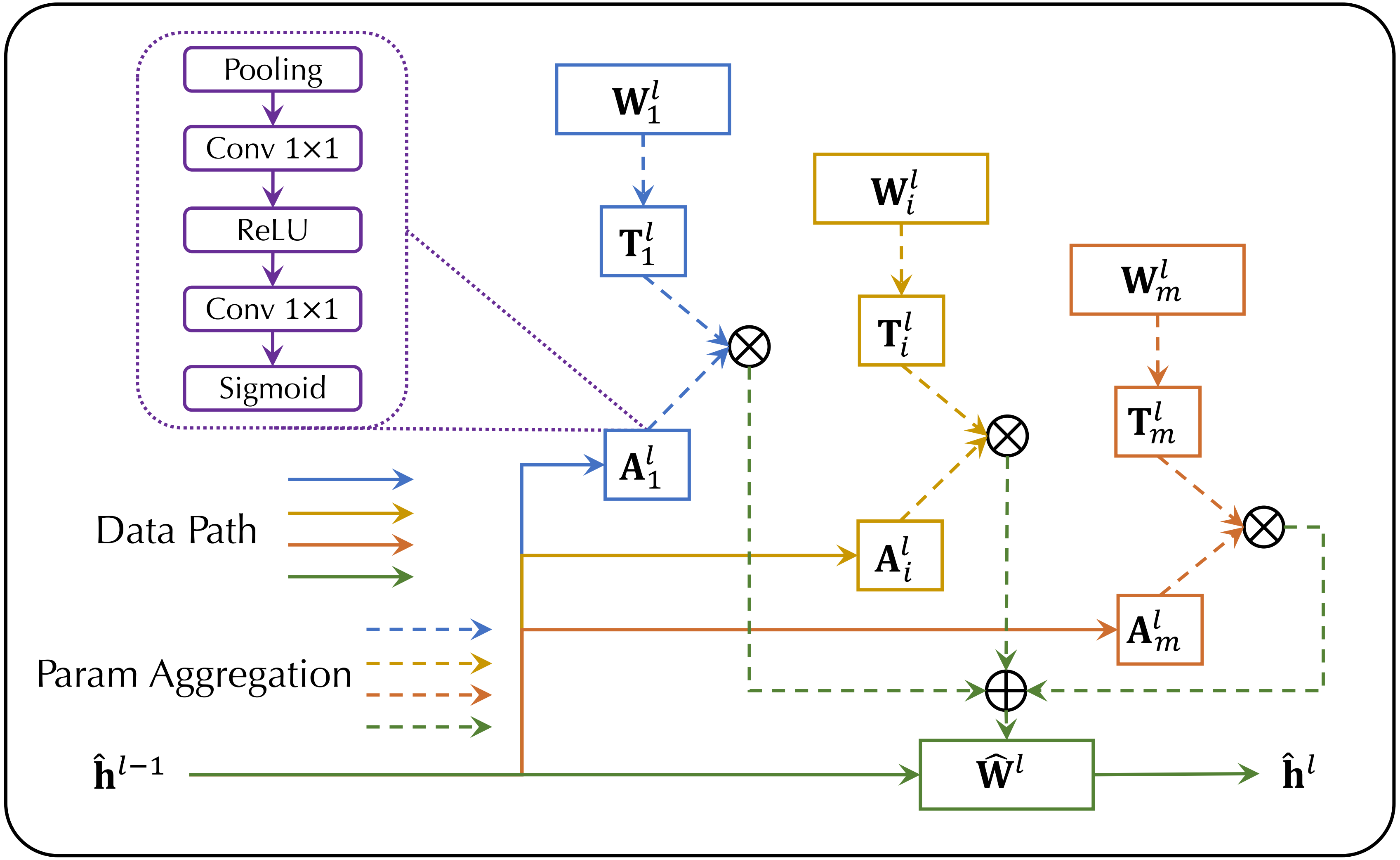
Zoo-Tuning

Adaptively aggregate source model parameters to derive target model.



Yang Shu, Zhi Kou, Zhangjie Cao, Jianmin Wang, **Mingsheng Long**. Zoo-Tuning: Adaptive Transfer from A Zoo of Models, **ICML** 2021

Adaptive Aggregation



Channel alignment

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

Data-dependent gating

$$\widehat{\mathbf{W}}^l = \sum_{i=1}^{m} a_i^l \widetilde{\mathbf{W}}_i^l$$

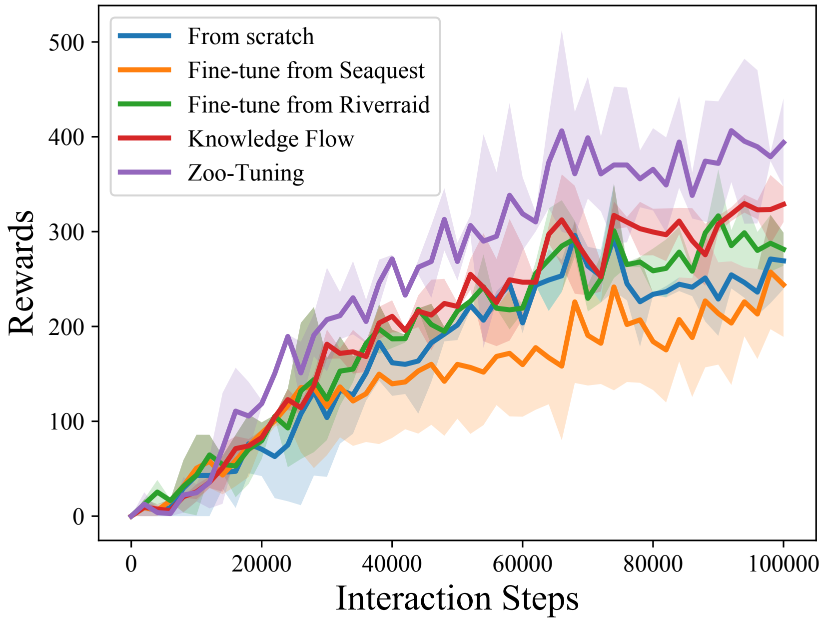
Experiments

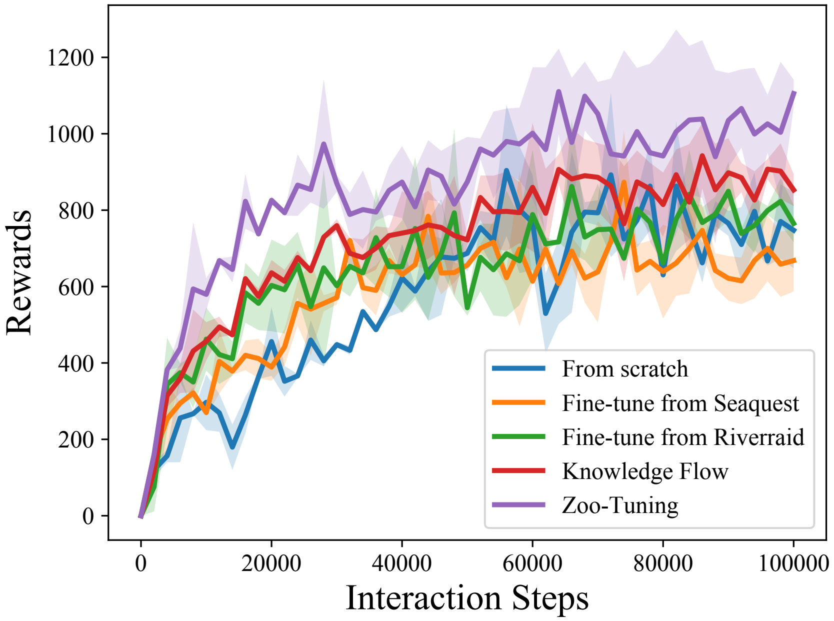
- Adaptive transfer from multiple models → Better accuracy.
- Adaptive aggregation of model parameters → More efficient than ensemble.

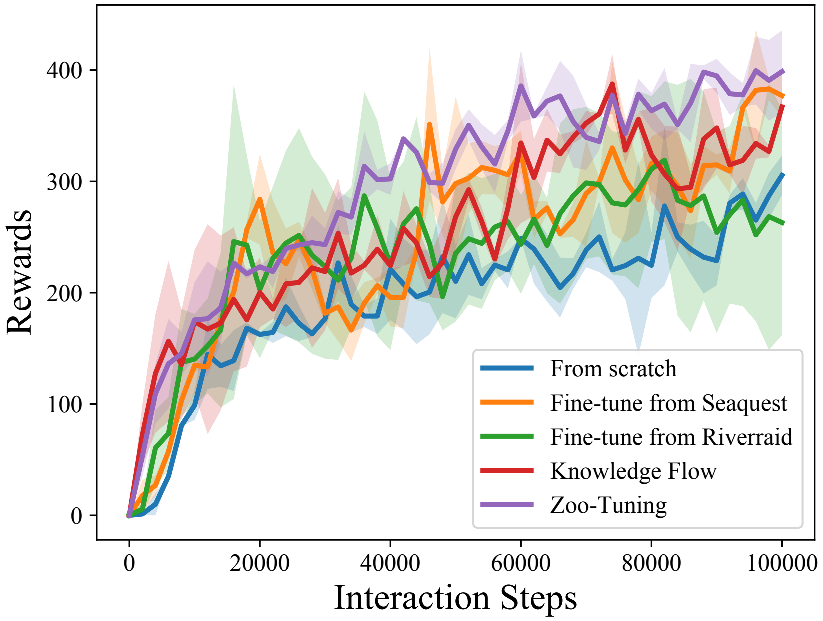
Model	General		Fine-grained			Specialized			Train		Inference	
	CIFAR-100	COCO-70	Aircraft	Cars	Indoors	DMLab	EuroSAT	Avg. Acc	GFLOPs	Params	GFLOPs	Params
ImageNet sup.	81.18	81.97	84.63	89.38	73.69	74.57	98.43	83.41	4.12	23.71M	4.12	23.71M
MOCO pt.	75.31	75.66	83.44	85.38	70.98	75.06	98.82	80.66	4.12	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.

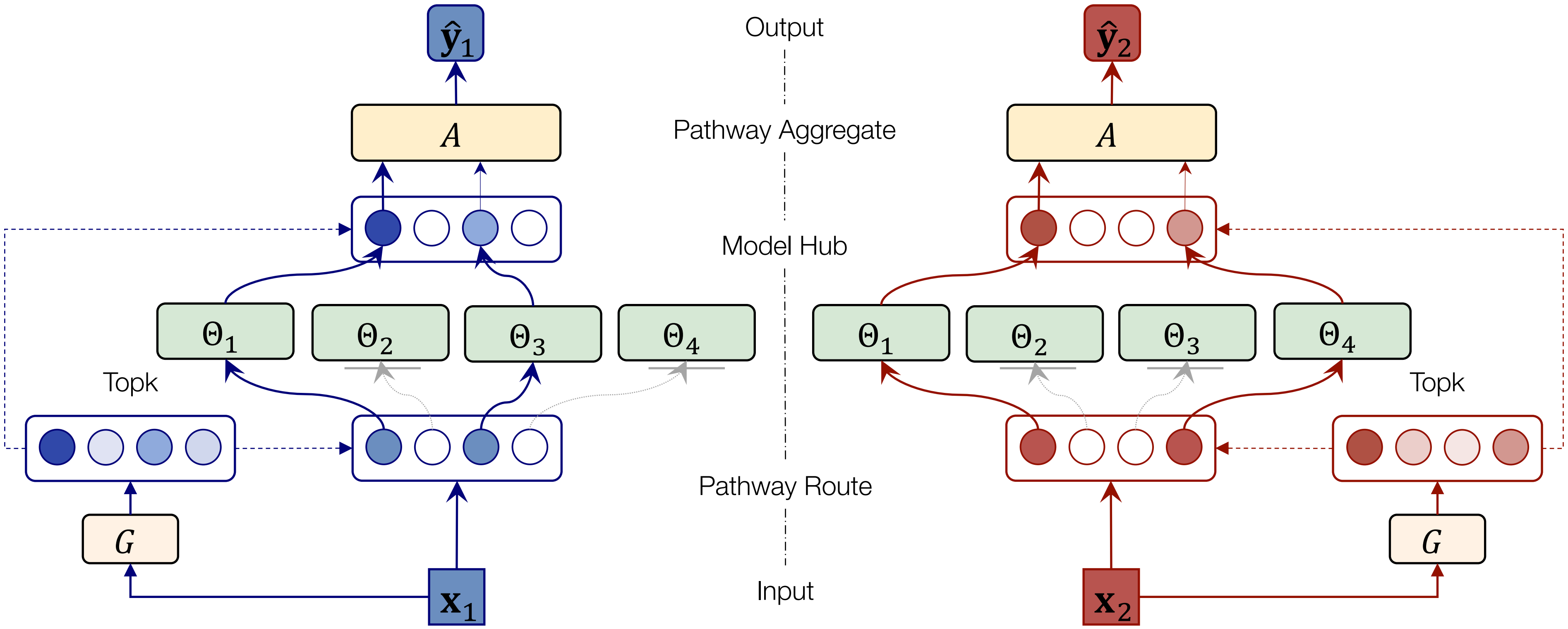






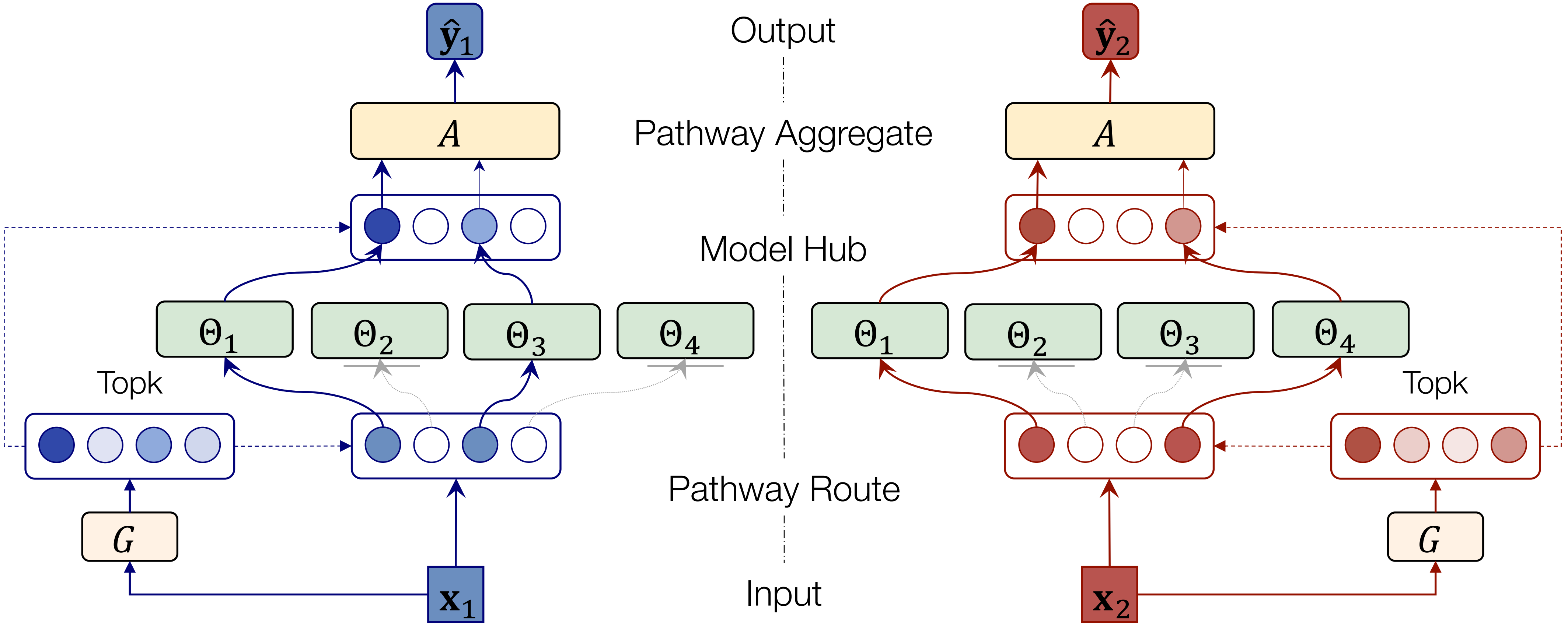
Heterogeneous Model Hub

Design data-dependent pathways throughout the Model Hub.



Hub-Pathway

- **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.



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

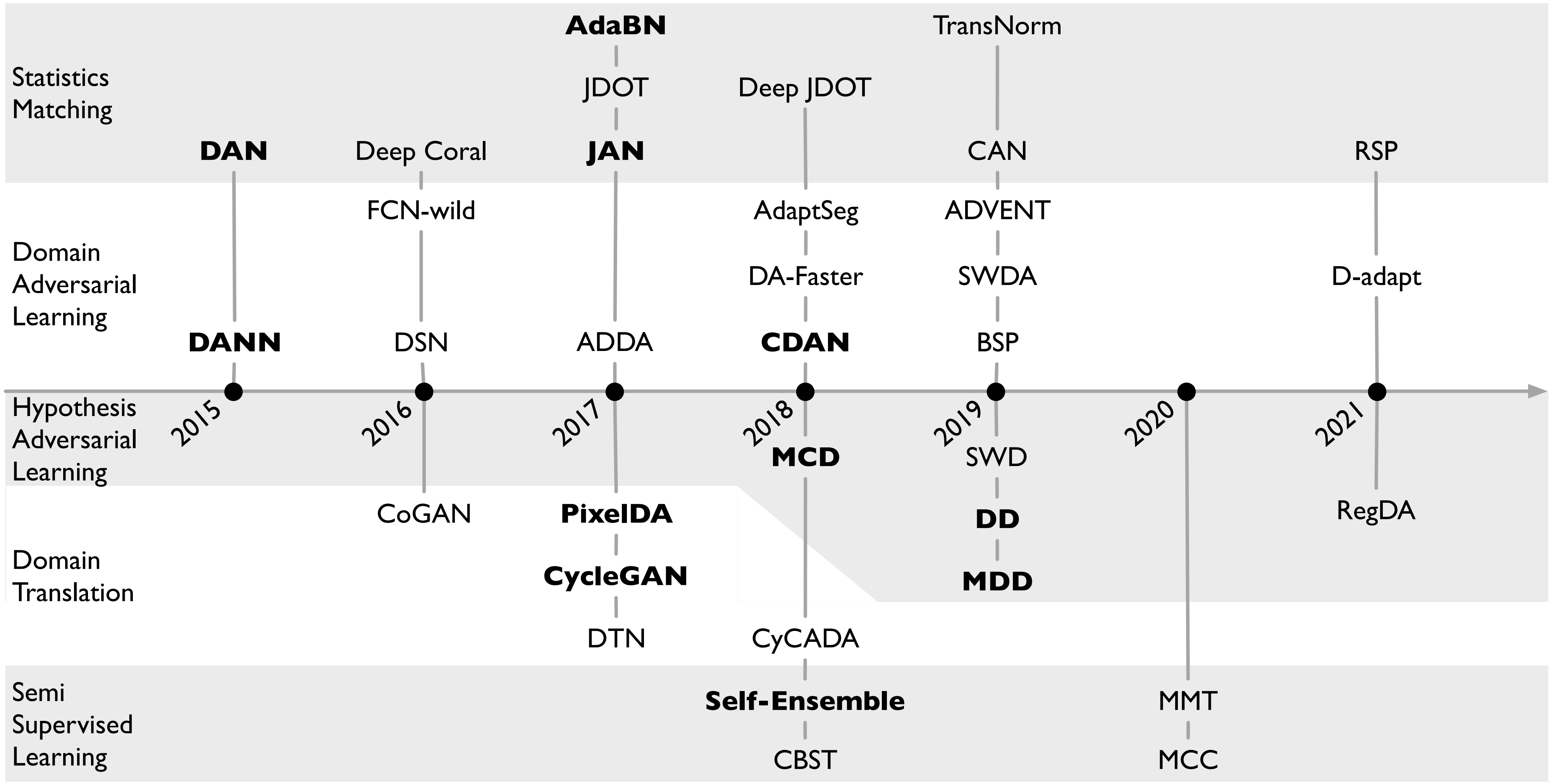
Adaptive Pathways

Remarks on Task Adaptation

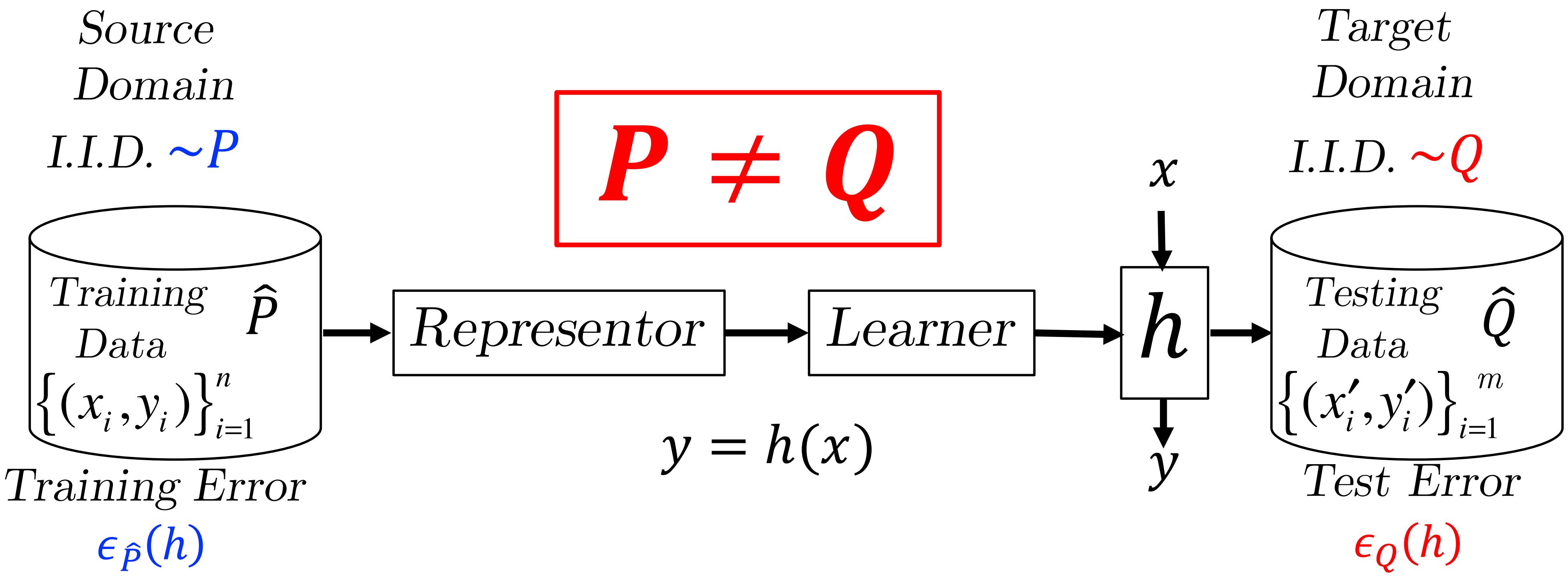
	Adaptation Accuracy[1]	Data Efficiency[2]	Parameter Efficiency[3]	Modality Scalability[4]	Task Scalability[5]
Feature Transfer	★	★★	★★★	★★★	★★★
Vanilla Fine-tuning	★★★	★	★	★★★	★★★
Domain Adaptive Tuning	★★★	★★	★	★★	★★★
Regularization Tuning	★★★	★★	★	★★★	★
Residual Tuning	★★	★★	★★	★★	★★
Parameter Difference Tuning	★★	★★	★★	★★★	★★★
Metric Learning	★	★★★	★★★	★★★	★
Prompt Learning	★★	★★★	★★★	★	★

[1] Accuracy when there are large-scale labeled data in downstream tasks.
[2] Accuracy when there are only small-scale labeled data in downstream tasks.
[3] Whether parameters can be controlled when the number of downstream tasks increases.
[4] Whether pre-trained model can be adapted to various modalities, such as text, graph.
[5] Whether pre-trained model can be adapted to different downstream tasks, such as detection.

Domain Adaptation



Domain Adaptation

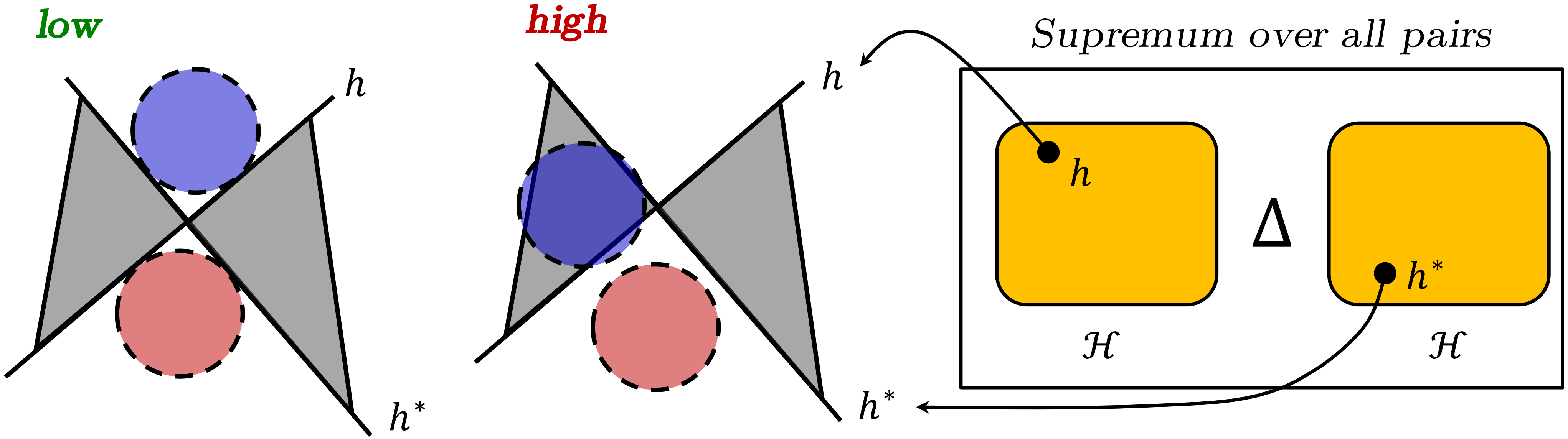


- How to measure the discrepancy between P and Q?
- Can we control target error $\epsilon_Q(h)$?

HΔH-Divergence

- HΔH-Divergence

$$d_{\mathcal{H}\Delta\mathcal{H}}(P,Q) = \sup_{h,h' \in \mathcal{H}} \left| \epsilon_Q(h,h') - \epsilon_P(h,h') \right|$$



Ben-David et al. A Theory of Learning from Different Domains. **Machine Learning** 2010.

HΔH-Divergence

- HΔH-Divergence

$$d_{\mathcal{H}\Delta\mathcal{H}}(P, Q) = \sup_{h,h' \in \mathcal{H}} \left| \epsilon_Q(h, h') - \epsilon_P(h, h') \right|$$

- Theorem (Generalization Bound with HΔH-Divergence)
- Denote by d the VC-dimension of hypothesis space $\mathcal{H}$ and ideal joint error $\epsilon_{ideal} = \epsilon_P(h^*) + \epsilon_Q(h^*)$. We have

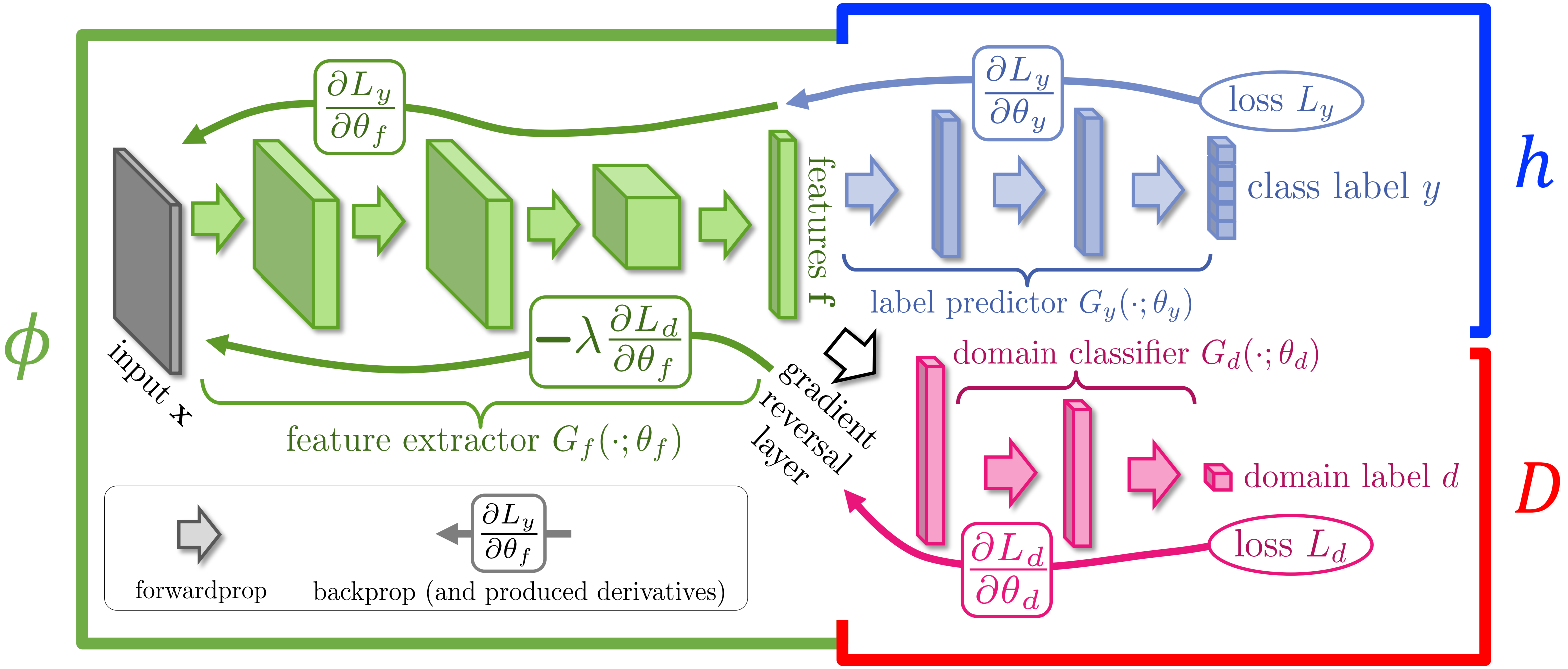
$$\epsilon_Q(h) \leq \epsilon_{\hat{P}}(h) + d_{\mathcal{H}\Delta\mathcal{H}}(\hat{P}, \hat{Q}) + \epsilon_{ideal} + O\left(\sqrt{\frac{d \log n}{n}} + \sqrt{\frac{d \log m}{m}}\right)$$

Ben-David et al. A Theory of Learning from Different Domains. **Machine Learning** 2010.

Domain Adversarial Learning

- Learning representation ϕ to minimize $d_{\mathcal{H}\Delta\mathcal{H}}(\phi(P), \phi(Q))$:

$$\min_{\phi,h} \left\{ \underbrace{\mathbb{E}_{(x,y)\sim P} L(h(\phi(x)), y)}_{\text{Supervised Learning on source}} + \max_{D} \underbrace{\left(\mathbb{E}_P L(D(\phi(x)), 1) + \mathbb{E}_Q L(D(\phi(x)), 0) \right)}_{\text{Minimize Upper bound of } d_{\mathcal{H}\Delta\mathcal{H}}} \right\}$$

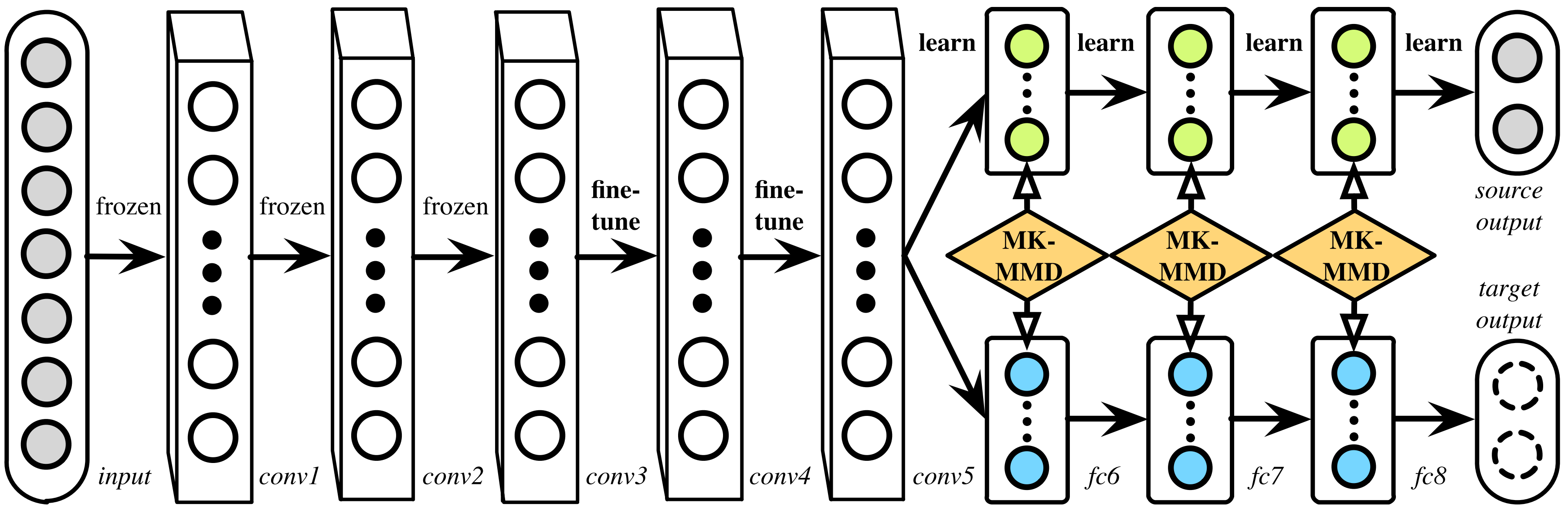


Ganin et al. Domain Adversarial Training of Neural Networks. **JMLR** 2016.

Domain Adversarial Learning

- Learning representation ϕ to minimize $d_{\text{MMD}}(\phi(P), \phi(Q))$:

$$\min_{\phi,h} \left\{ \underbrace{\mathbb{E}_{(x,y)\sim P} L\big(h(\phi(x)), y\big)}_{\text{Supervised Learning on source}} + \underbrace{\lambda \max_{k\in\mathcal{K}} \left\| \mathbb{E}_P[\phi(\boldsymbol{x}^s)] - \mathbb{E}_Q[\phi(\boldsymbol{x}^t)] \right\|_{\mathcal{K}}^2}_{\text{Minimize Upper bound of } d_{\mathcal{H}\Delta\mathcal{H}}} \right\}$$



Long et al. Learning Transferable Features with Deep Adaptation Networks. ICML 2015.

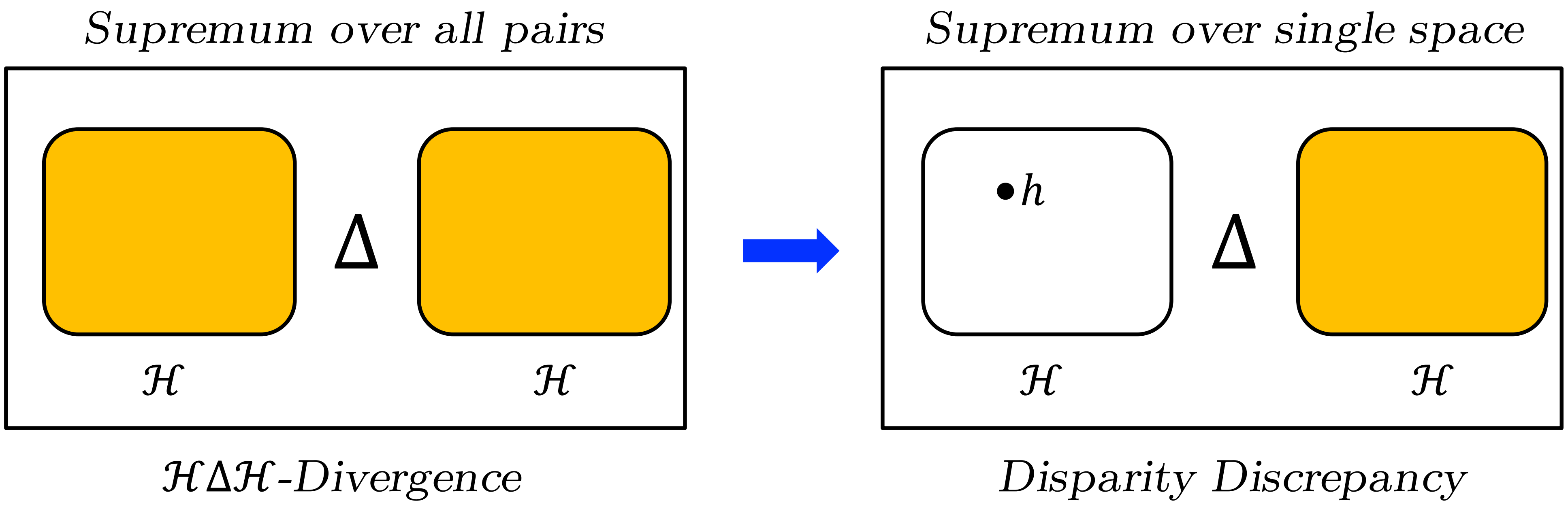
Theory vs. Practice

- Binary Classification vs. Multiclass Classification
- Discrete Classifier vs. Classifier with Scoring Function
- $H \Delta H$ is excessively large that is hard to estimate and optimize

Disparity Discrepancy

- Disparity Discrepancy

$$d_{h,\mathcal{H}}(P,Q) = \sup_{h' \in \mathcal{H}} \left(\epsilon_Q(h,h') - \epsilon_P(h,h')\right)$$



Disparity Discrepancy

- Disparity Discrepancy

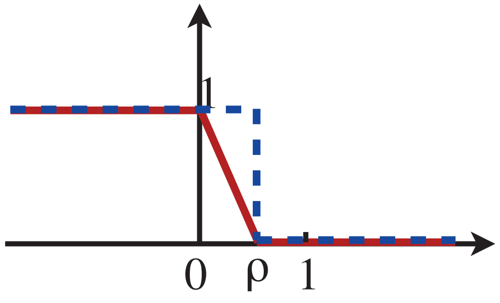
$$d_{h,\mathcal{H}}(P,Q) = \sup_{h'\in\mathcal{H}} \left(\epsilon_Q(h,h') - \epsilon_P(h,h')\right)$$

- Theorem (Generalization Bound with Disparity Discrepancy)
- For any $\delta > 0$ and binary classifier $h \in \mathcal{H}$, with probability $1 - 3\delta$, we have

$$\epsilon_Q(h) \le \epsilon_{\hat{P}}(h) + d_{h,\mathcal{H}}\left(\hat{P},\hat{Q}\right) + \epsilon_{ideal} + 2\mathfrak{R}_{n,P}(\mathcal{H})$$
$$+2\mathfrak{R}_{n,P}(\mathcal{H}\Delta\mathcal{H}) + 2\mathfrak{R}_{m,Q}(\mathcal{H}\Delta\mathcal{H}) + 2\sqrt{\frac{\log\frac{2}{\delta}}{2n}} + \sqrt{\frac{\log\frac{2}{\delta}}{2m}}$$

Yuchen Zhang & Mingsheng Long et al. Bridging Theory and Algorithm for Domain Adaptation. ICML 2019.

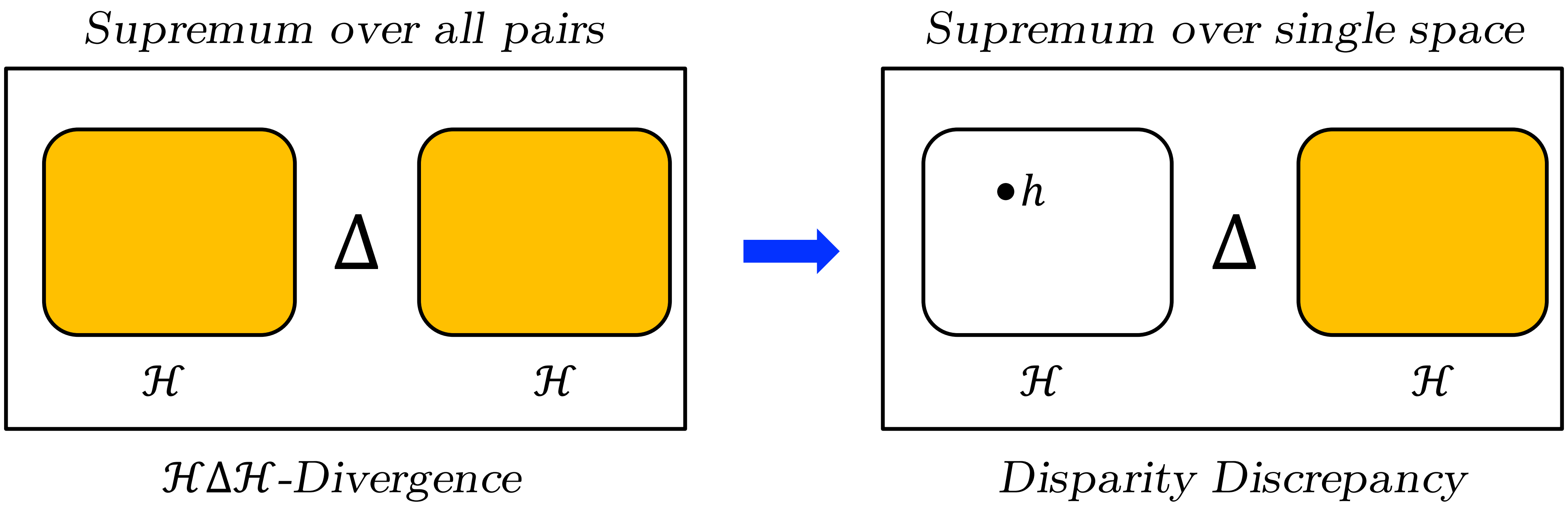
Margin Disparity Discrepancy



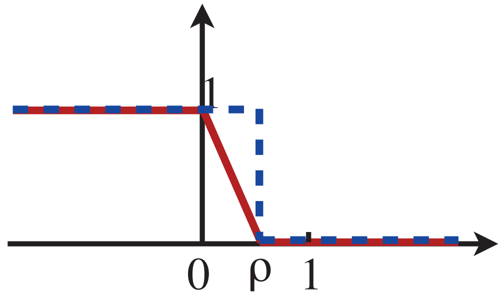
Margin Loss

- Margin Disparity Discrepancy

$$d_{f,\mathcal{F}}^{(\rho)}(P,Q) = \sup_{f' \in \mathcal{F}} \left(\epsilon_Q^{(\rho)}(f', f) - \epsilon_P^{(\rho)}(f', f) \right)$$



Margin Disparity Discrepancy



Margin Loss

- Margin Disparity Discrepancy

$$d_{f,\mathcal{F}}(P,Q) = \sup_{f'\in\mathcal{F}} \left(\epsilon_Q^{(\rho)}(f',f) - \epsilon_P^{(\rho)}(f',f)\right)$$

- Theorem (Generalization Bound with Margin Disparity Discrepancy)
- For any $\delta > 0$ and scoring classifier $f \in \mathcal{F}$, with probability $1 - 3\delta$, we have

$$\epsilon_Q(f) \le \epsilon_{\hat{P}}^{(\rho)}(f) + d_{h,\mathcal{H}}^{(\rho)}(\hat{P},\hat{Q}) + \epsilon_{ideal} + \frac{2k^2}{\rho}\mathfrak{R}_{n,P}(\Pi_1\mathcal{F})$$

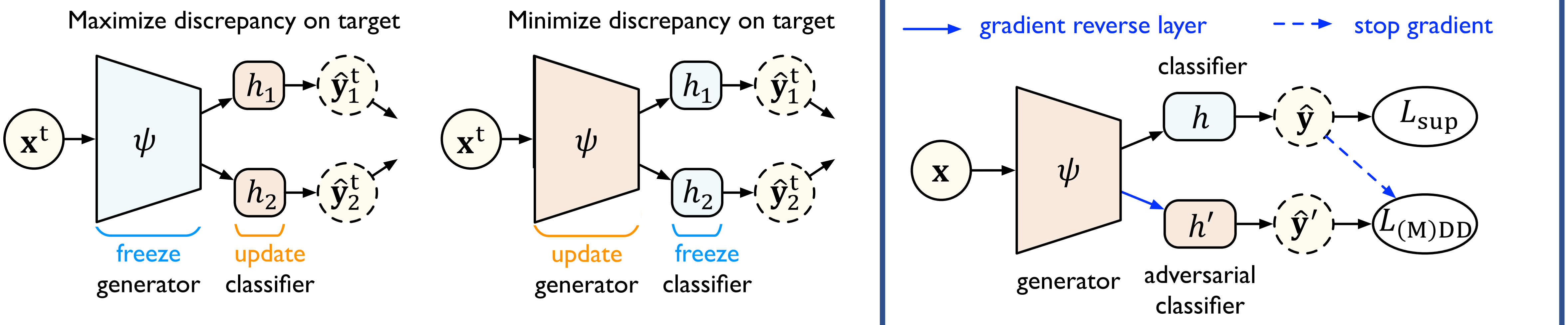
$$+\frac{k}{\rho}\mathfrak{R}_{n,P}(\Pi_{\mathcal{H}}\mathcal{F}) + \frac{k}{\rho}\mathfrak{R}_{m,Q}(\Pi_{\mathcal{H}}\mathcal{F}) + 2\sqrt{\frac{\log\frac{2}{\delta}}{2n}} + \sqrt{\frac{\log\frac{2}{\delta}}{2m}}$$

Yuchen Zhang & Mingsheng Long et al. Bridging Theory and Algorithm for Domain Adaptation. ICML 2019.

Hypothesis Adversarial Learning

- Margin Disparity Discrepancy

$$\min_{\psi,f} \max_{f'} \epsilon_{\hat{P}}^{(\rho)}(f) + \left(\epsilon_{\hat{Q}}^{(\rho)}(f',f) - \epsilon_{\hat{P}}^{(\rho)}(f',f)\right)$$



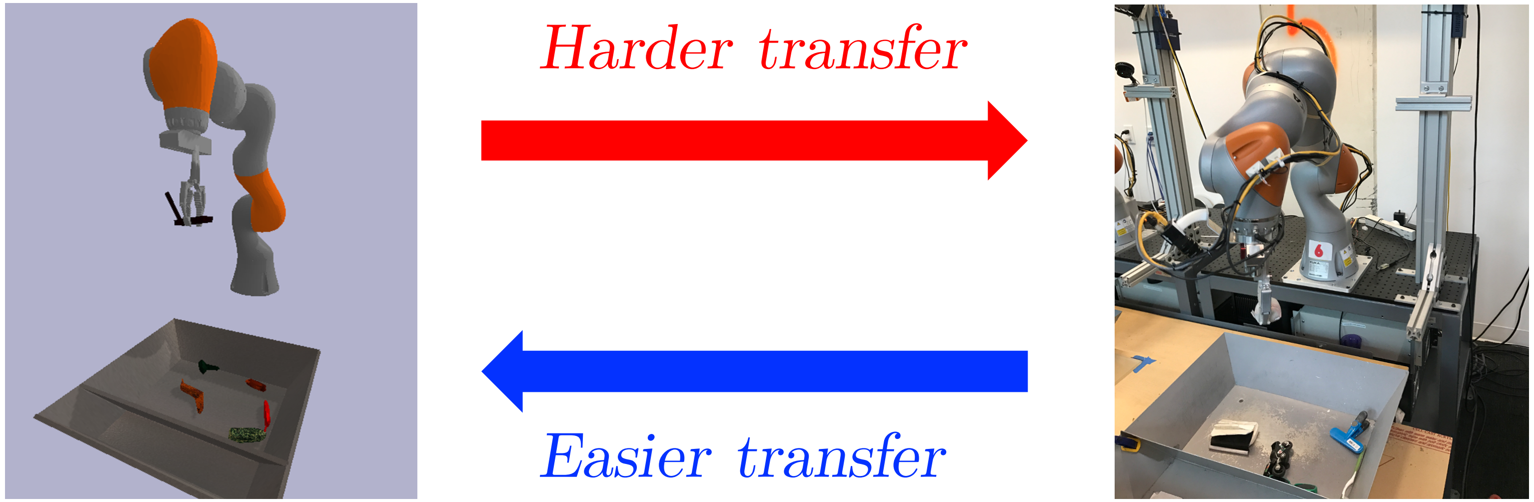
(a) MCD (Step B)

(a) MCD (Step C)

(c) MDD / DD

Yuchen Zhang & Mingsheng Long et al. Bridging Theory and Algorithm for Domain Adaptation. ICML 2019.

Theory vs. Practice



Simulation Real

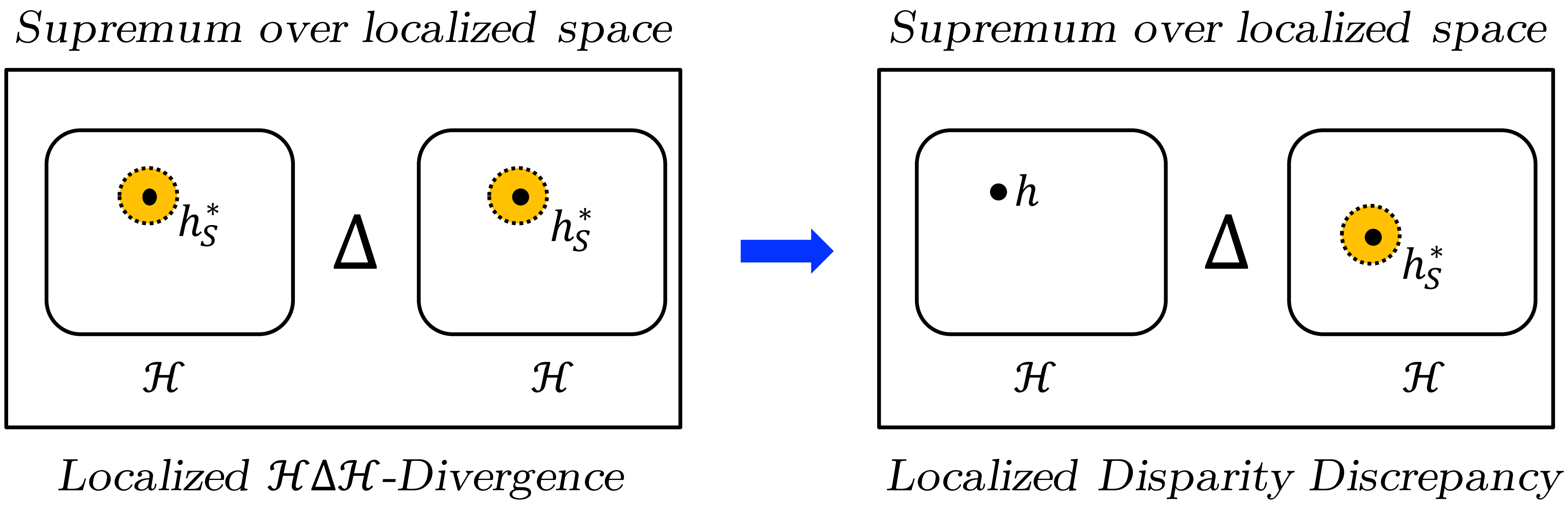
- A common observation is that difficulty of transfer is asymmetric.
- Previous bounds will remain unchanged after switching P and Q.
- Previous discrepancies are supremum over the whole hypothesis space.

Localized Disparity Discrepancy

Pre-train on source

- Margin Disparity Discrepancy

$$d_{h,\mathcal{H}_r}(P,Q) = \sup_{h' \in \mathcal{H}_r := \{h \in \mathcal{H} \mid \mathbb{E}_P L(h(x),y) \le r\}} \left(\epsilon_Q(h,h') - \epsilon_P(h,h')\right)$$



Localized Disparity Discrepancy

Pre-train on source

- Localized Disparity Discrepancy

$$d_{h,\mathcal{H}_r}(P,Q) = \sup_{h' \in \mathcal{H}_r := \{h \in \mathcal{H} \mid \mathbb{E}_P L(h(x),y) \le r\}} \left(\epsilon_Q(h,h') - \epsilon_P(h,h')\right)$$

- Theorem (Generalization Bound with Localized Disparity Discrepancy)
- For any $\delta > 0$ and binary classifier $h \in \mathcal{H}$, with probability $1-\delta$, we have

$$\epsilon_Q(h) \le \epsilon_{\hat{P}}(f) + d_{h,\mathcal{H}_r}(\hat{P},\hat{Q}) + \epsilon_{ideal} + O\left(\frac{d\log n}{n} + \frac{d\log m}{m}\right)$$

$$+O\left(\sqrt{\frac{(\epsilon_{\hat{P}}(h) + r)d\log n}{n}} + \sqrt{\frac{\left(\epsilon_{\hat{P}}(h) + d_{h,\mathcal{H}_r}(\hat{P},\hat{Q}) + r\right)d\log m}{m}}\right)$$

Yuchen Zhang & Mingsheng Long et al. Localized Discrepancies for Domain Adaptation. arXiv 2021.

Remarks on Domain Adaptation

	Adaptation Accuracy[2]	Data Efficiency[2]	Modality Scalability[3]	Task Scalability[4]	Theory Guarantee[5]
Statistics Matching	★	★★★	★★★	★★	★★★
Domain Adversarial Learning	★★	★★	★★★	★★	★★★
Hypothesis Adversarial Learning	★★★	★★	★★★	★★	★★★
Domain Translation	★★	★	★	★★★	★
Semi-Supervised Learning	★★	★★	★★	★	★

[1] Accuracy when there are large-scale data in source and target domains.
[2] Accuracy when there are only small-scale data in source and target domains.
[3] Whether the model can be adapted to various modalities, such as text, time series.
[4] Whether the model can be adapted to different tasks, such as regression, detection.
[5] Whether the generalization error of target domain can be theoretically bounded in adaptation.

Transfer Learning Library

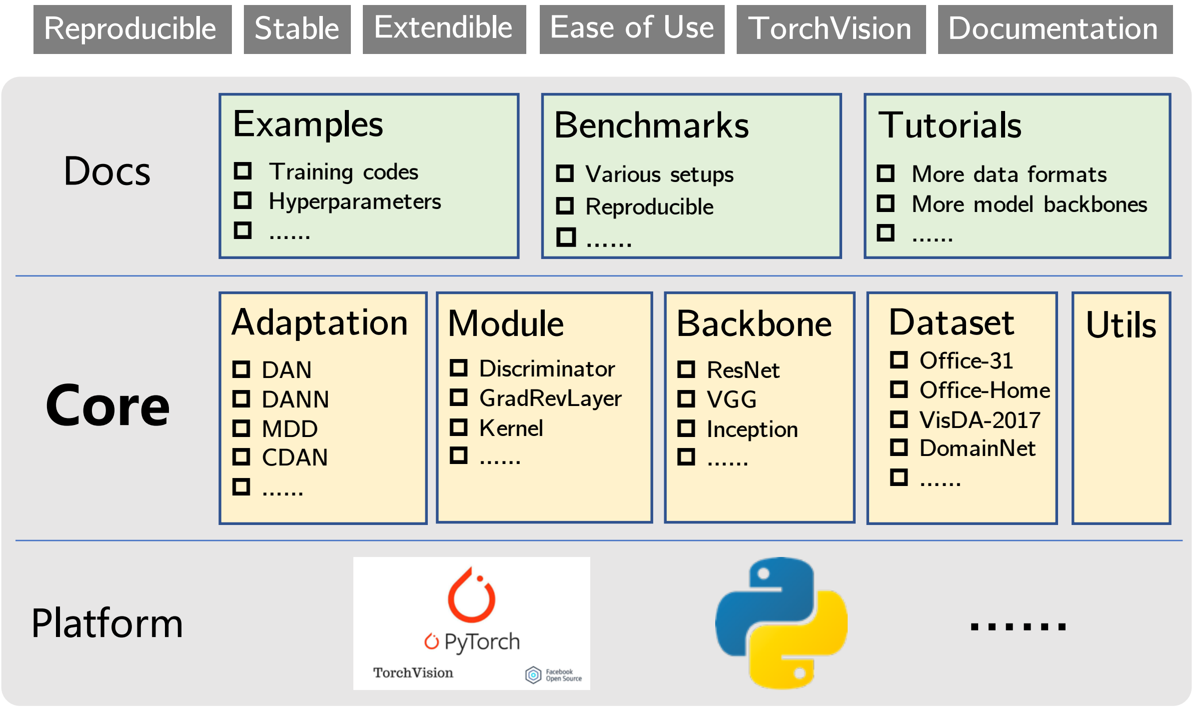
thuml / **Transfer-Learning-Library** Public

Transfer Learning Library for Domain Adaptation, Task Adaptation, and Domain Generalization

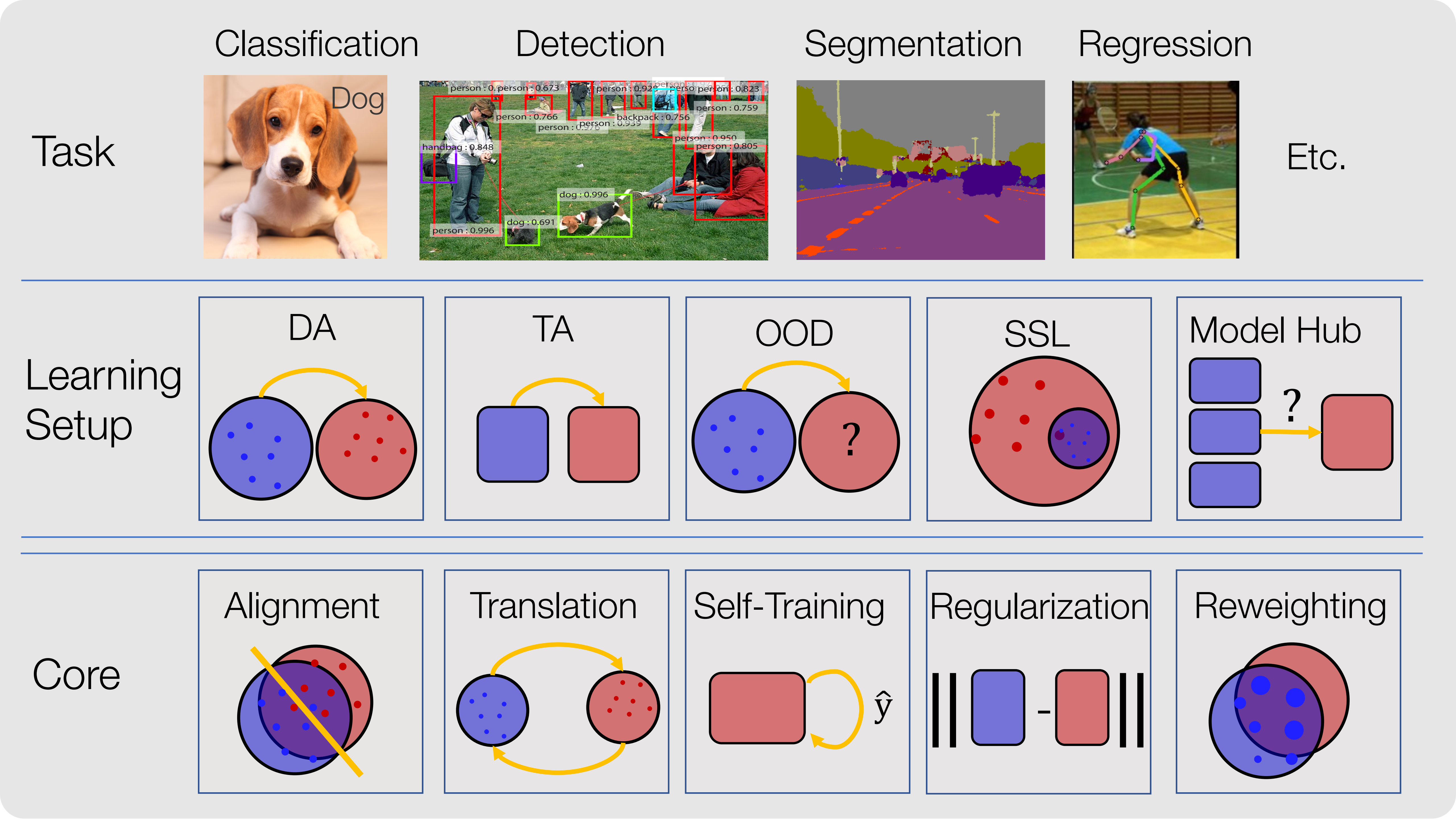
transfer.thuml.ai

MIT license

2k stars 394 forks



https://github.com/thuml/Transfer-Learning-Library

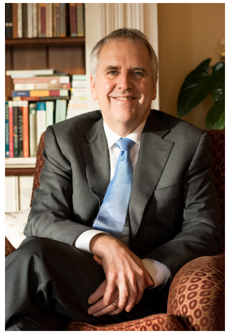


Task
Classification
Detection
Segmentation
Regression
Dog
Etc.
Learning Setup
DA
TA
OOD
SSL
Model Hub
?
Core
Alignment
Translation
Self-Training
Regularization
Reweighting
ŷ

Thank You!

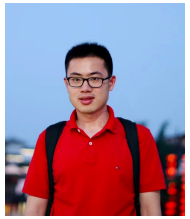
Mingsheng Long
(龙明盛)
Tsinghua University
mingsheng@tsinghua.edu.cn

Jianmin Wang
(王建民)
Tsinghua University
jimwang@tsinghua.edu.cn



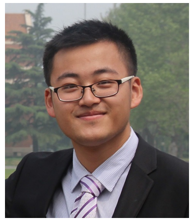
Michael I. Jordan
(迈克尔・欧文・乔丹)
UC Berkeley
jordan@cs.berkeley.edu

Yuchen Zhang
(张育宸)

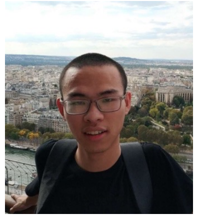


Zhangjie Cao
(曹张杰)

Han Zhu
(朱晗)



Yue Cao
(曹越)



Kaichao You
(游凯超)

Junguang Jiang
(江俊广)

Ximei Wang
(王希梅)

Xinyang Chen
(陈新阳)

Yang Shu
(树扬)

清華大學 Tsinghua University

大数据系统软件国家工程研究中心

THUML