IJCAI-ECAI 2026 Tutorial

Principles, Structures, and Adaptive Agents:

Scaling AI under Limited Supervision

Module A: Principles — Learning with Limited Labels

Yaqing Wang

Beijing Institute of Mathematical Sciences and Applications (BIMSA)

https://wangyaqing.github.io/

Roadmap

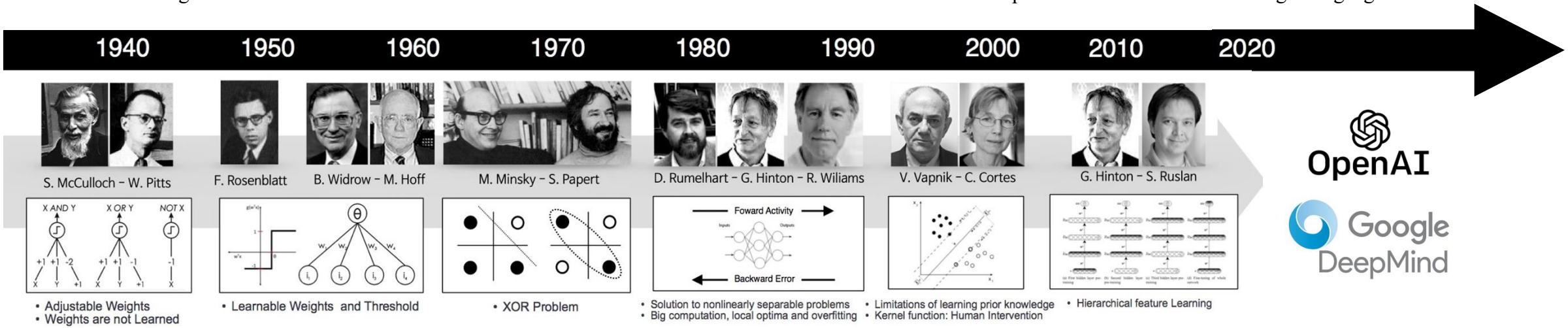
- Principles of Few-Shot Learning
- Adaptation Mechanisms
  - Optimization-based
  - Metric-based
  - Amortization-based
  - In-context learning
- Choosing an Adaptation Mechanism
- Practice: From Applications to Few-Shot Problems

Roadmap

- ☐ Principles of Few-Shot Learning
- ☐ Adaptation Mechanisms
  - ☐ Optimization-based
  - ☐ Metric-based
  - ☐ Amortization-based
  - ☐ In-context learning
- ☐ Choosing an Adaptation Mechanism
- ☐ Practice: From Applications to Few-Shot Problems

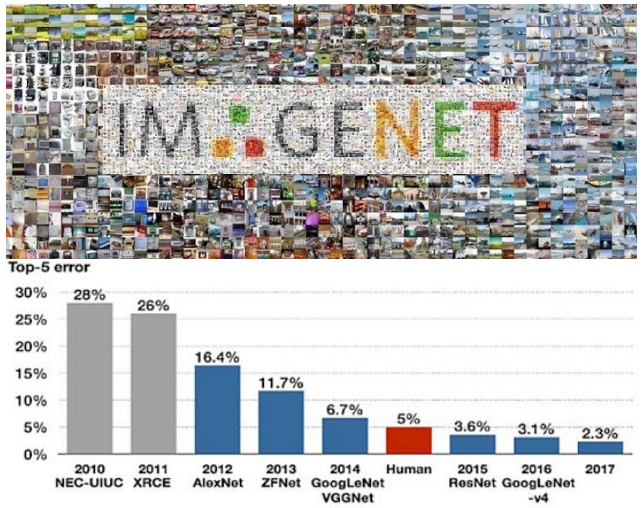
Machine Learning Landmarks

Hand-designed Features Shallow Models Deep Models Large Language Models

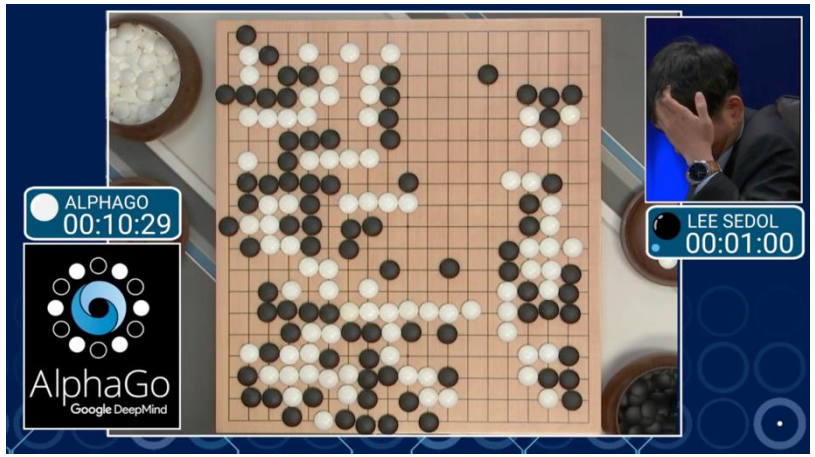


OpenAI

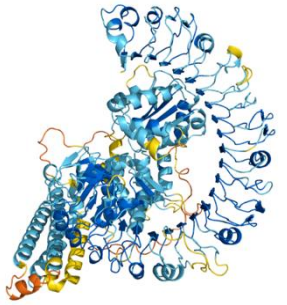
Google DeepMind



ImageNet 2012



AlphaGo 2016



AlphaFold

AlphaFold 2020

ChatGPT 2022

The Success of Machine Learning

Large, diverse data ⟶ Broad generalization

(+ large models)

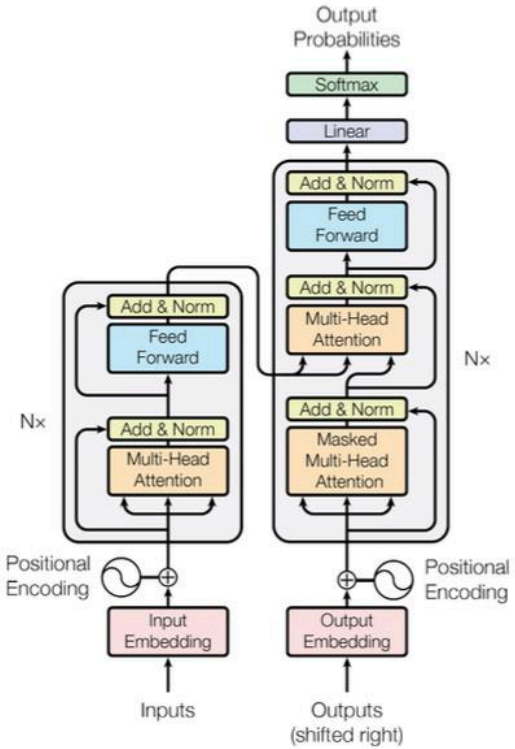
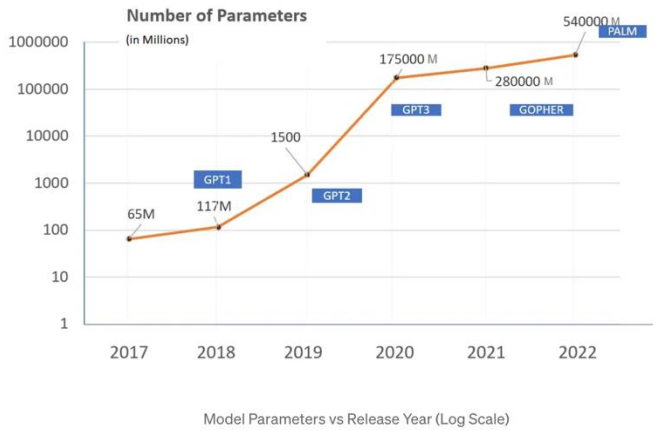
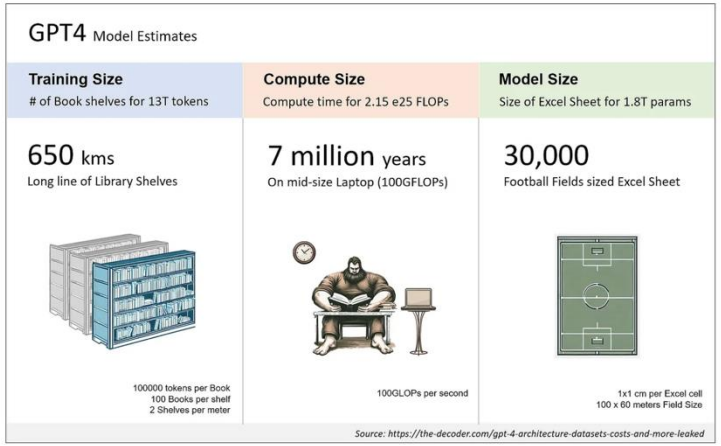


Figure 1: The Transformer - model architecture.



Model Parameters vs Release Year (Log Scale)

Training Tokens vs Release Year (Log Scale)

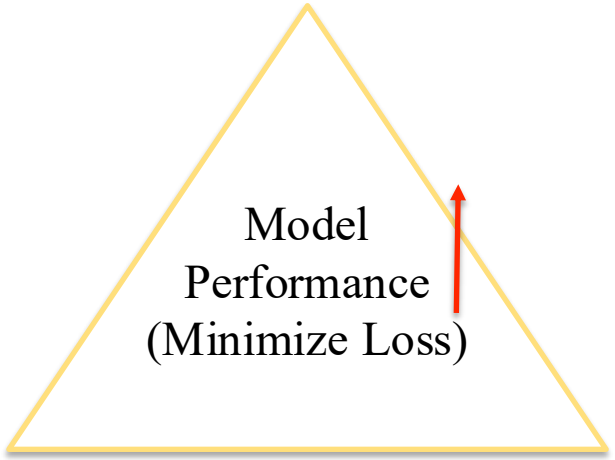


GPT4 Model Estimates

Under the paradigm of supervised learning.

Scaling Law

Compute Scaling
(GPUs, training time, other costs)

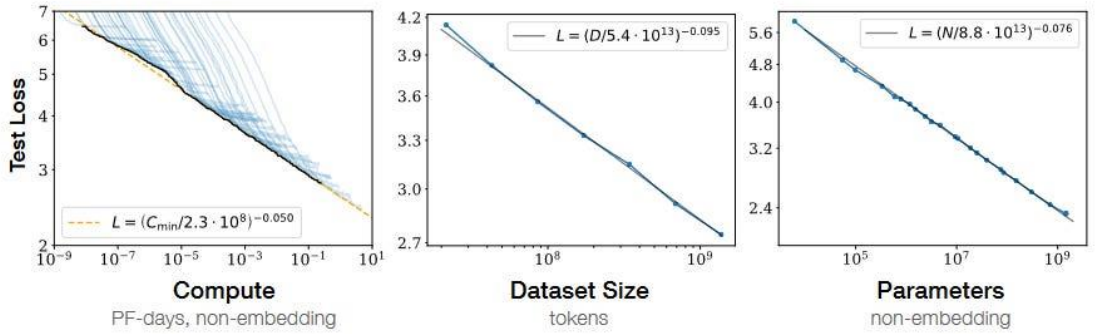


Data Scaling
(Number of tokens)

Model Size Scaling
(Number of parameters)

What is scaling law?

Scaling laws in machine learning refer to empirical relationships that describe how the performance of a model (such as a neural network) improves as a function of various factors like the amount of training data, model size (number of parameters), and computational resources.



When It Fails…

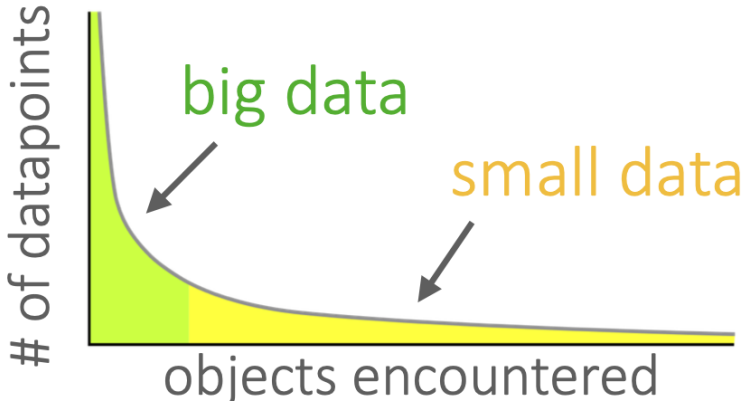
What if you do not have a large dataset?

medical imaging, robotics, translation for rare languages, personalized education and recommendations…

What if you want a general-purpose AI system in the real world?

Need to continuously adapt and learn on the job. Learning each thing from scratch will not cut it.

What if your data has a long tail?



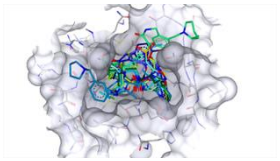
These settings break the supervised learning paradigm.

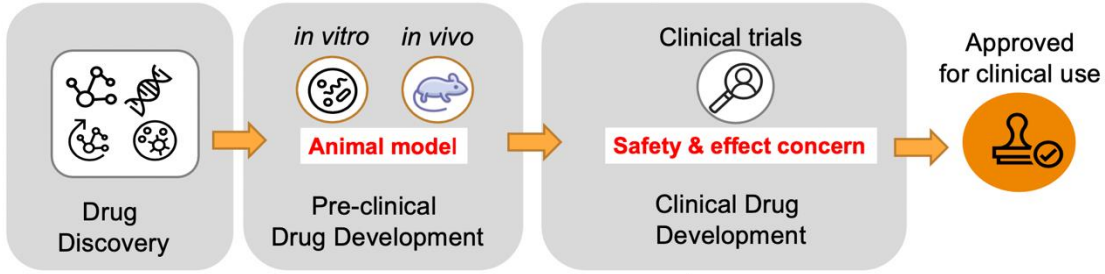
Scientific Scarcity

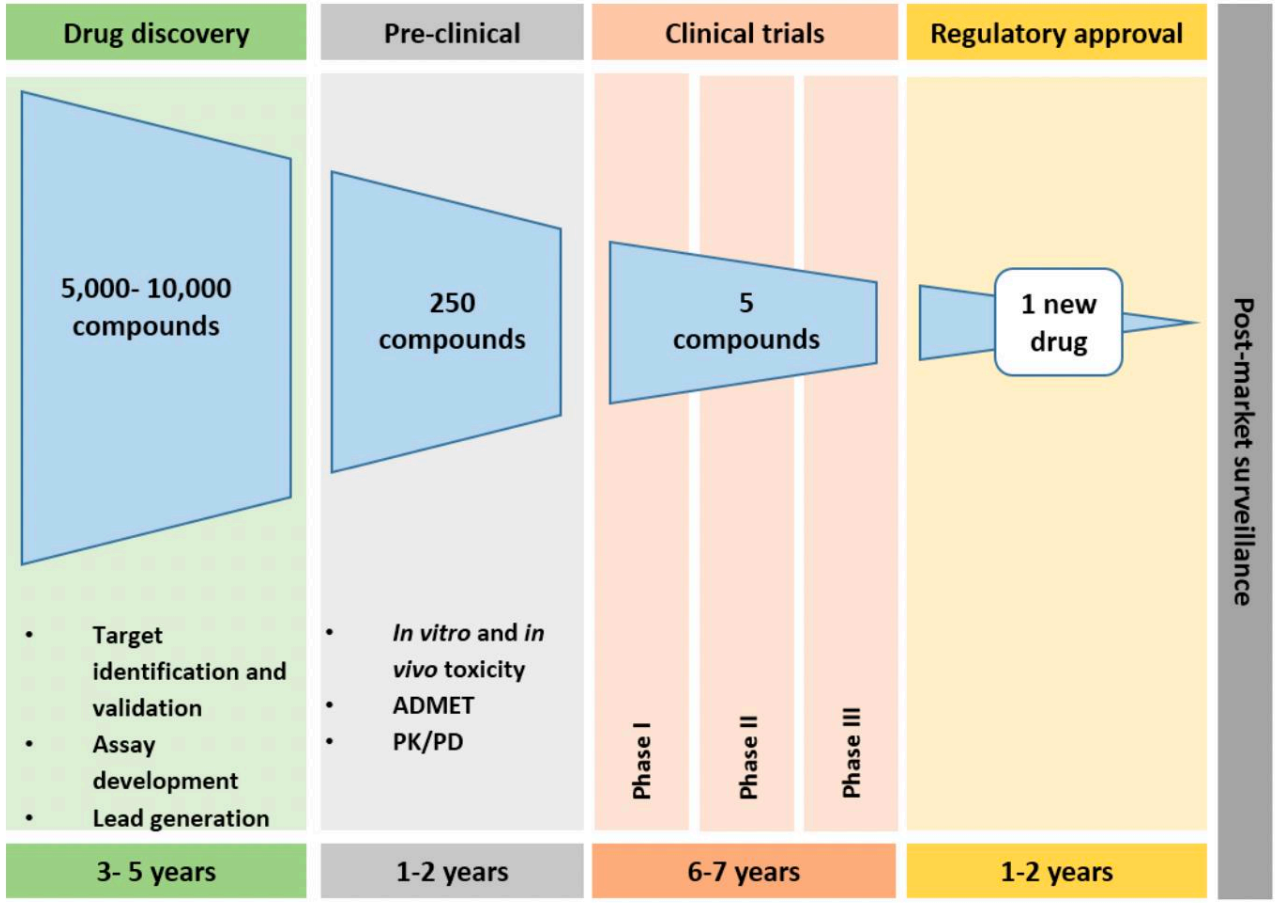
Example: Drug Discovery

Lengthy process, expensive and high failure rate：
Total cost: $2.6 billion
Total time: ≥ 10 years
Clinical success rate: ~12%
Scarcity of labeled samples

Large chemical space to be explored
Drug-like small molecules：$10^{33} - 10^{60}$
Existing molecular databases to linearly screen：$\sim 10^{6}$
Hard to enumerate



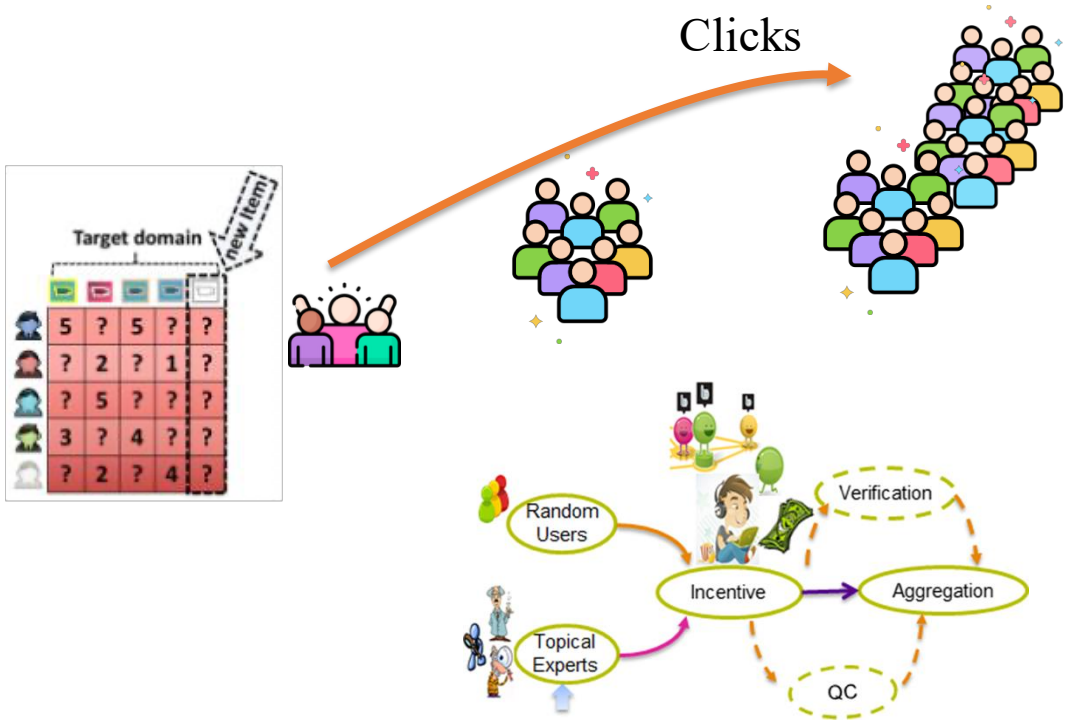




Drug Development [H. Matthews et al., Proteomes 2016]

Efficient Data Utilization

Example: Recommendation for Emerging Items



Abundant new items appear each day

- cold-start phase: have no user interaction,
- warm-up phase: accumulate a few initial clicks
- common phase: get sufficient clicks

High-quality annotations are hard to obtain!

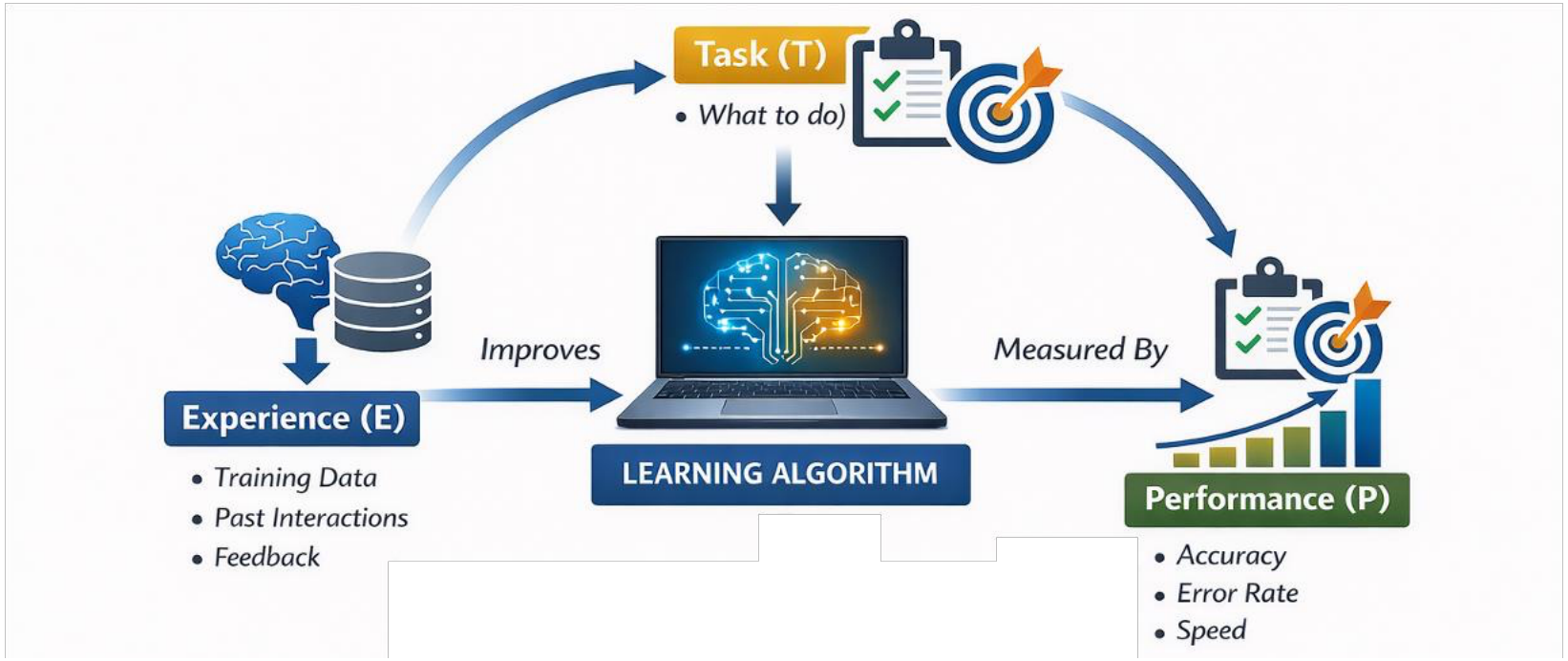
⇒ Require efficient incremental learning

Few-Shot Learning (FSL)

In Few-shot Learning (FSL), E is limited!

➢ Machine Learning [M. T. Mitchell. 1997]

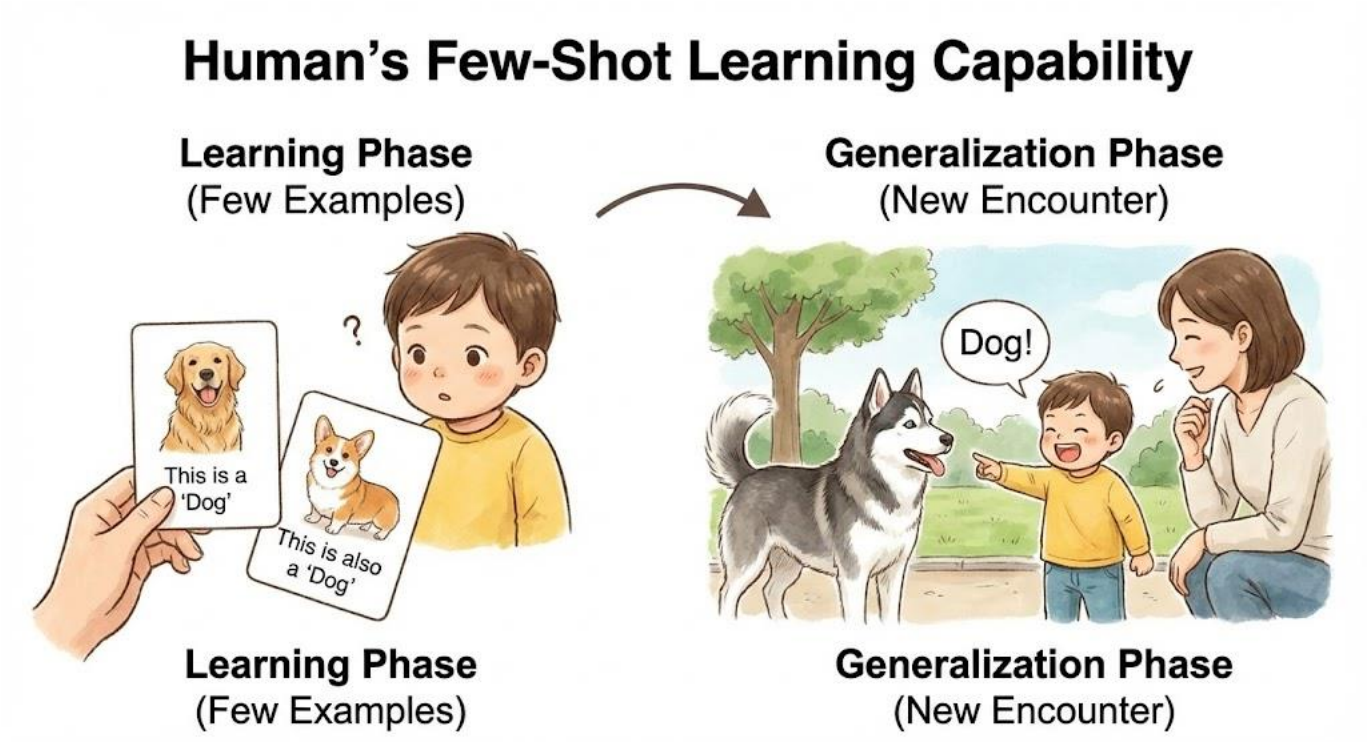
A computer program is said to learn from experience E with respect to some classes of task T and performance measure P if its performance can improve with E on T measured by P.



Generalizing from a Few Examples

Human Learning Efficiency: Humans are natural Few-Shot Learners.

A child can grasp a new concept (e.g., recognizing a "dog") after seeing just one or two examples, without needing thousands of training images.



Human Intelligence is Supported by Prior Knowledge

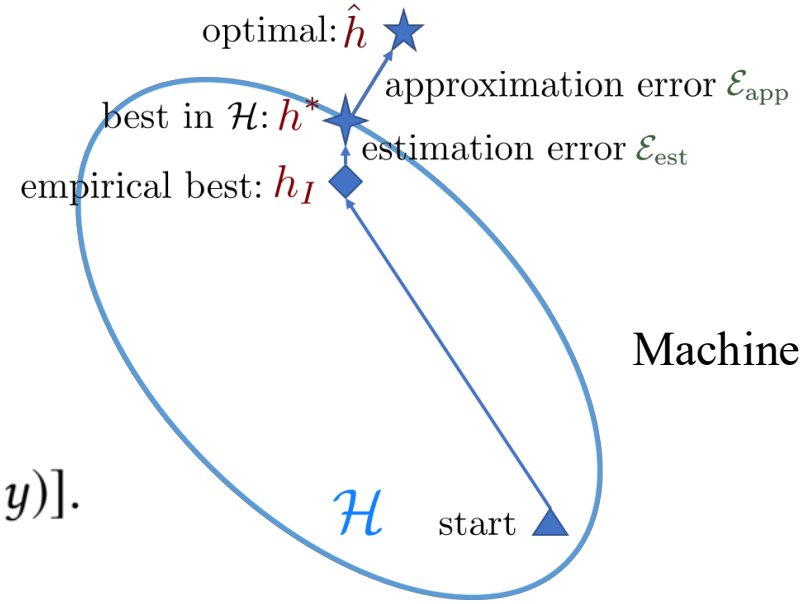
humanity's inherent prior knowledge

societal prior knowledge

personal prior knowledge

Vision: To advance machine learning technology and bridge the gap between AI and human capabilities with the help of prior knowledge

Core Issue of FSL



Machine learning

➢ Expected Risk

Unknown Data distribution

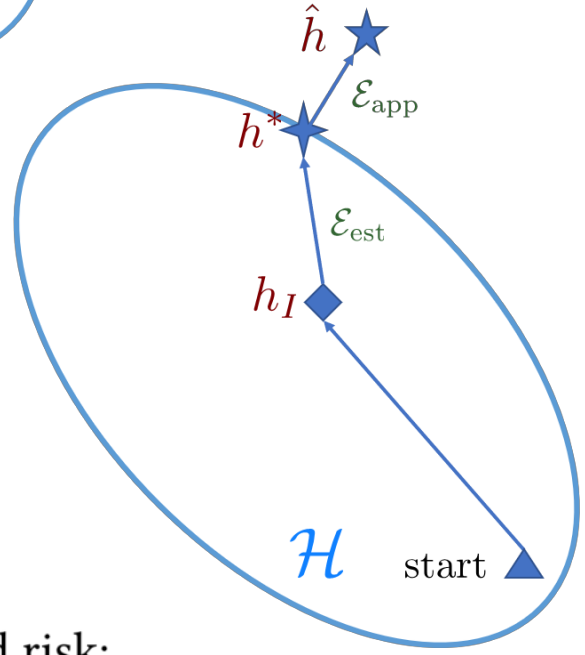
$$R(h) = \int \ell(h(x), y)\, dp(x, y) = \mathbb{E}[\ell(h(x), y)].$$

hypothesis

➢ Empirical Risk

$$R_I(h) = \frac{1}{I} \sum_{i=1}^{I} \ell(h(x_i), y_i)$$

Unreliable Empirical Risk Minimizer



FSL

➢ Total Error

$$\mathbb{E}[R(h_I) - R(\hat{h})] = \mathbb{E}[R(h^*) - R(\hat{h})] + \mathbb{E}[R(h_I) - R(h^*)]$$

- $\hat{h} = \arg\min_h R(h)$ be the function that minimizes the expected risk;
- $h^* = \arg\min_{h \in \mathcal{H}} R(h)$ be the function in $\mathcal{H}$ that minimizes the expected risk;
- $h_I = \arg\min_{h \in \mathcal{H}} R_I(h)$ be the function in $\mathcal{H}$ that minimizes the empirical risk.

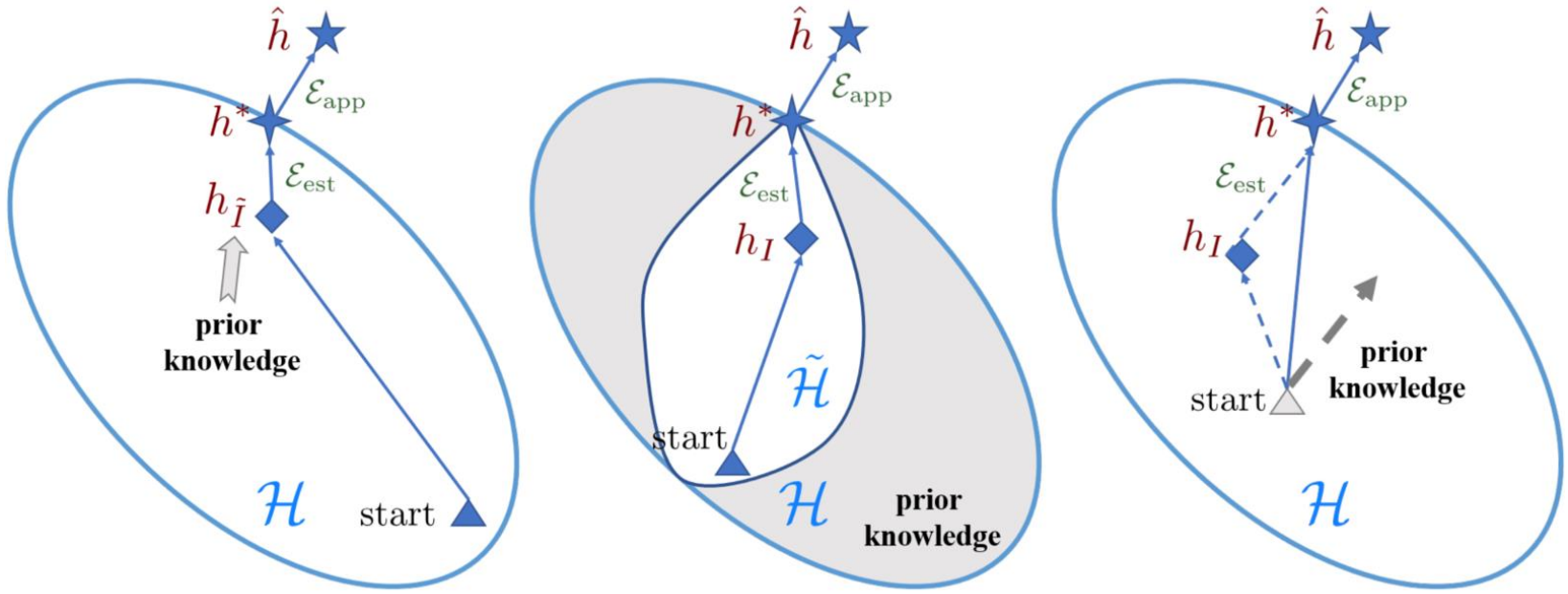
"Generalizing from a Few Examples: A Survey on Few-Shot Learning." ACM Computing Surveys (CSUR), 2020.

Prior Knowledge Helps FSL

➢Total Error

$$\mathbb{E}[R(h_I) - R(\hat{h})] = \underbrace{\mathbb{E}[R(h^*) - R(\hat{h})]}_{\mathcal{E}_{\text{app}}(\mathcal{H})} + \underbrace{\mathbb{E}[R(h_I) - R(h^*)]}_{\mathcal{E}_{\text{est}}(\mathcal{H}, I)}$$

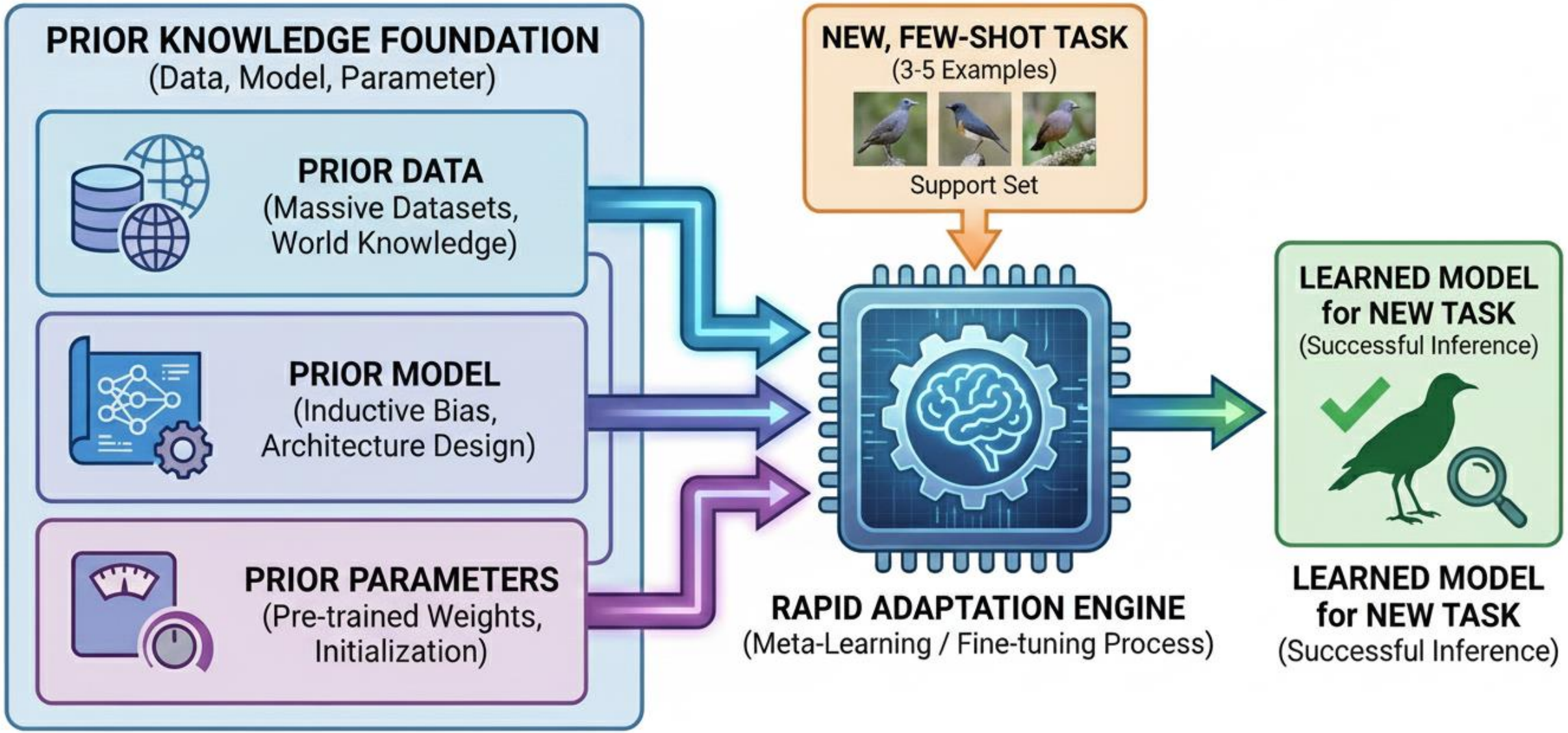
- Data: Increase the number of training samples using prior knowledge
- Model: Constrain the complexity of H using prior knowledge
- Algorithm: Alter search strategy in H by prior knowledge



(a) Data. (b) Model. (c) Algorithm.

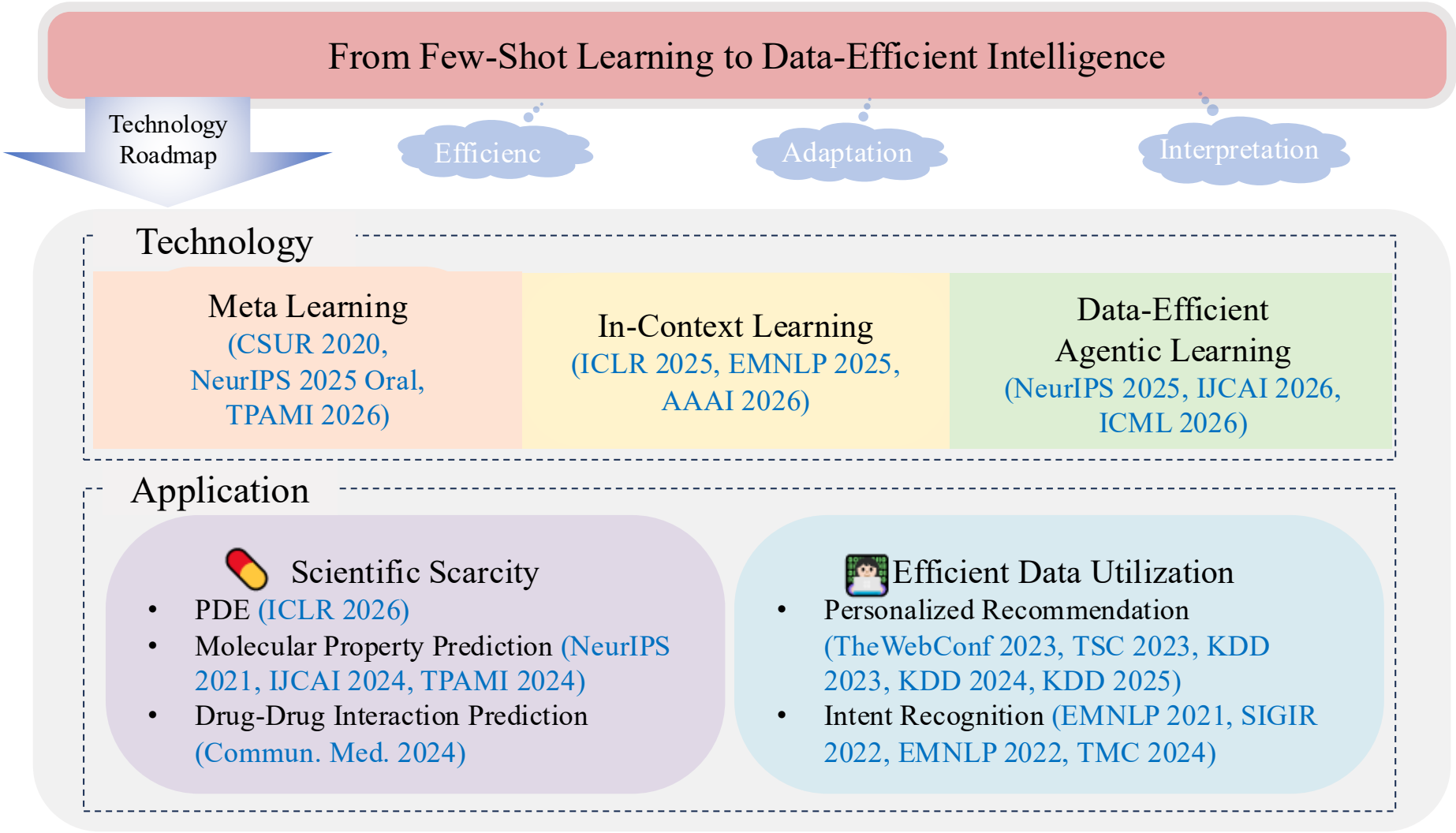
"Generalizing from a Few Examples: A Survey on Few-Shot Learning." ACM Computing Surveys (CSUR), 2020.

Few-Shot Learning with Prior Knowledge



Ack: Gemini

My Research Landscape



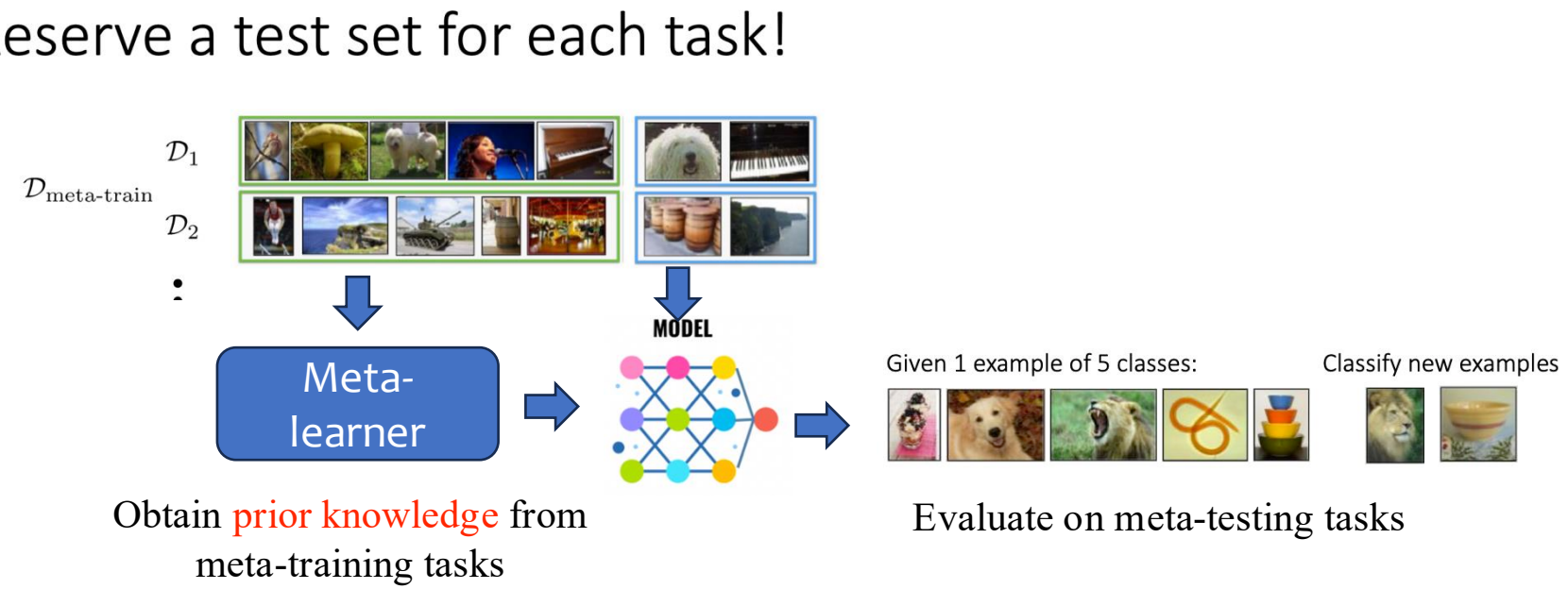
Roadmap

- ✓ Principles of Few-Shot Learning
- ☐ Adaptation Mechanisms
  - ☐ Optimization-based
  - ☐ Metric-based
  - ☐ Amortization-based
  - ☐ In-context learning
- ☐ Choosing an Adaptation Mechanism
- ☐ Practice: From Applications to Few-Shot Problems

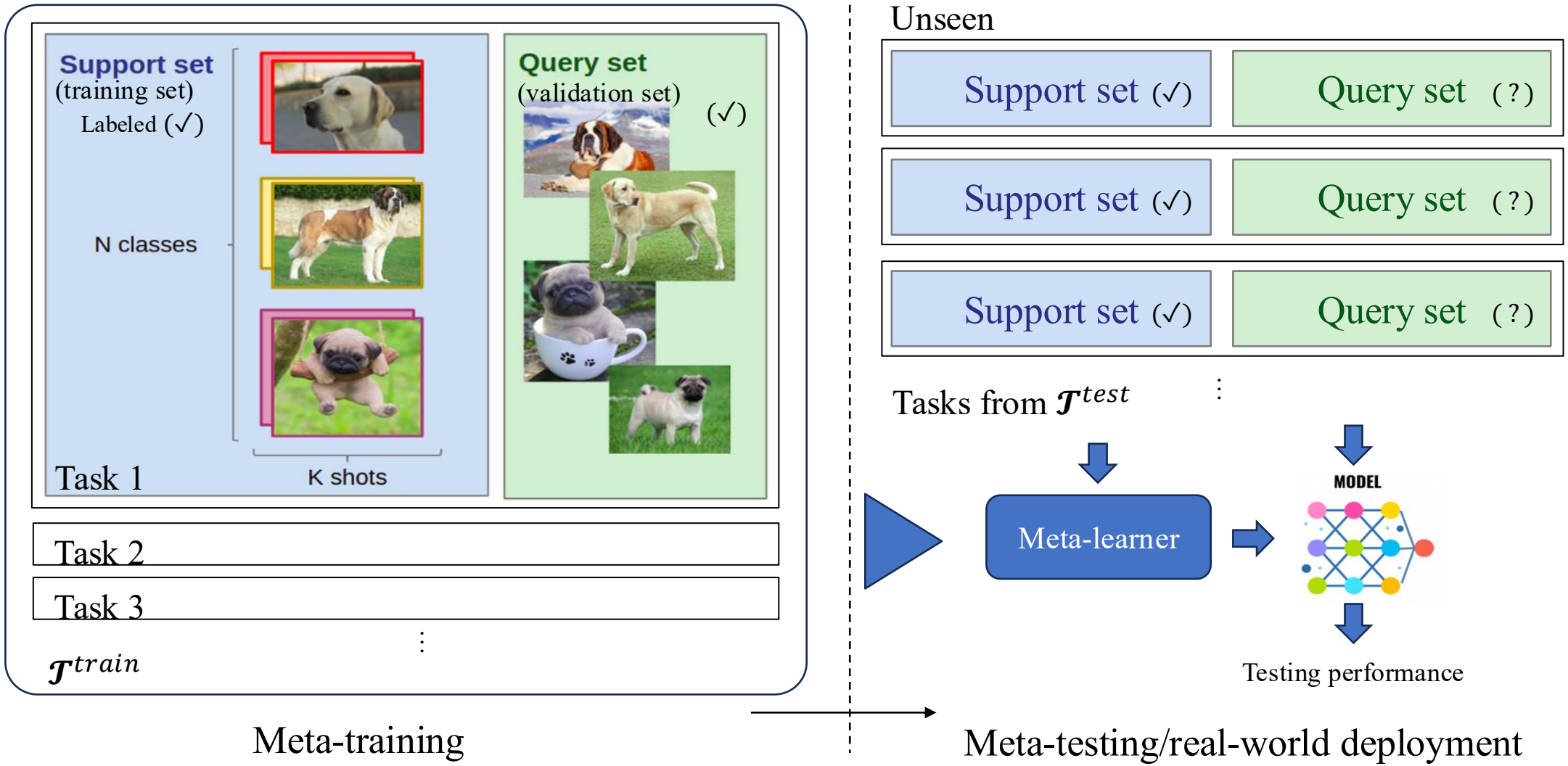
Meta-Learning

How to generalize to new tasks with only a limited number of examples?

-Based on a simple machine learning principle: test and train conditions must match

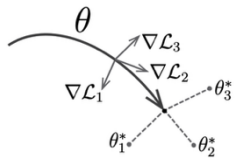


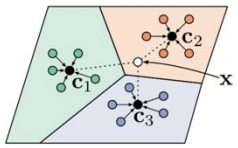
Overview of the Meta-Train/Test Pipeline

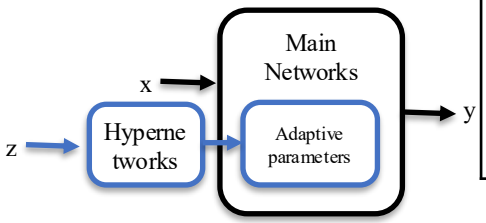


Mainstream Meta-Learning Algorithms

Learning (prior knowledge) to learn (adapt to each task).
Different kinds of prior knowledge and adaptation:







	Learning	To Learn	My Works
Optimization	Optimizer (start point, strategy…)	Fine-tune the model parameters	NeurIPS 2021, TPAMI 2024
Metric	Transferable representation and similarity metric	Predict based on distance/similarity on the space	SIGIR 2022, NeurIPS 2025 (Oral, Top 0.4%)
Amortization	Black-box task encoder	Generate specific weights	WWW 2023, IJCAI 2024, KDD 2024, TMC 2024

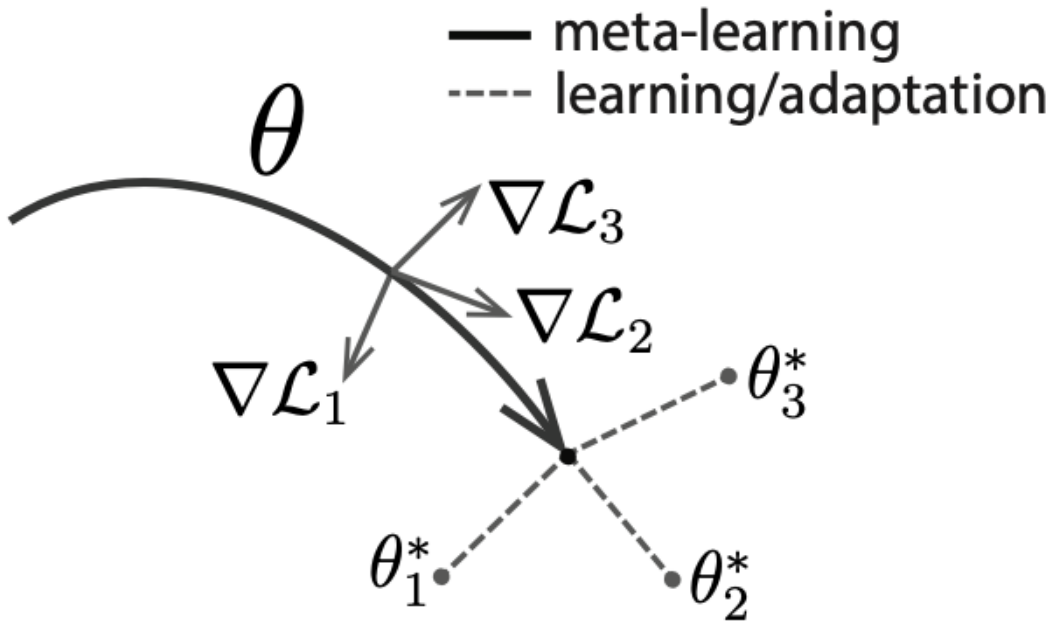
Roadmap

- ✓ Principles of Few-Shot Learning
- ☐ Adaptation Mechanisms
  - ☐ Optimization-based
  - ☐ Metric-based
  - ☐ Amortization-based
  - ☐ In-context learning
- ☐ Choosing an Adaptation Mechanism
- ☐ Practice: From Applications to Few-Shot Problems

Optimization-Based Meta-Learning

Learning optimizer to fine-tune

Meta-learn the start-point:



Optimization-Based—MAML

Algorithm:

Step 1: Adapt the shared model initialization to each meta-training task.

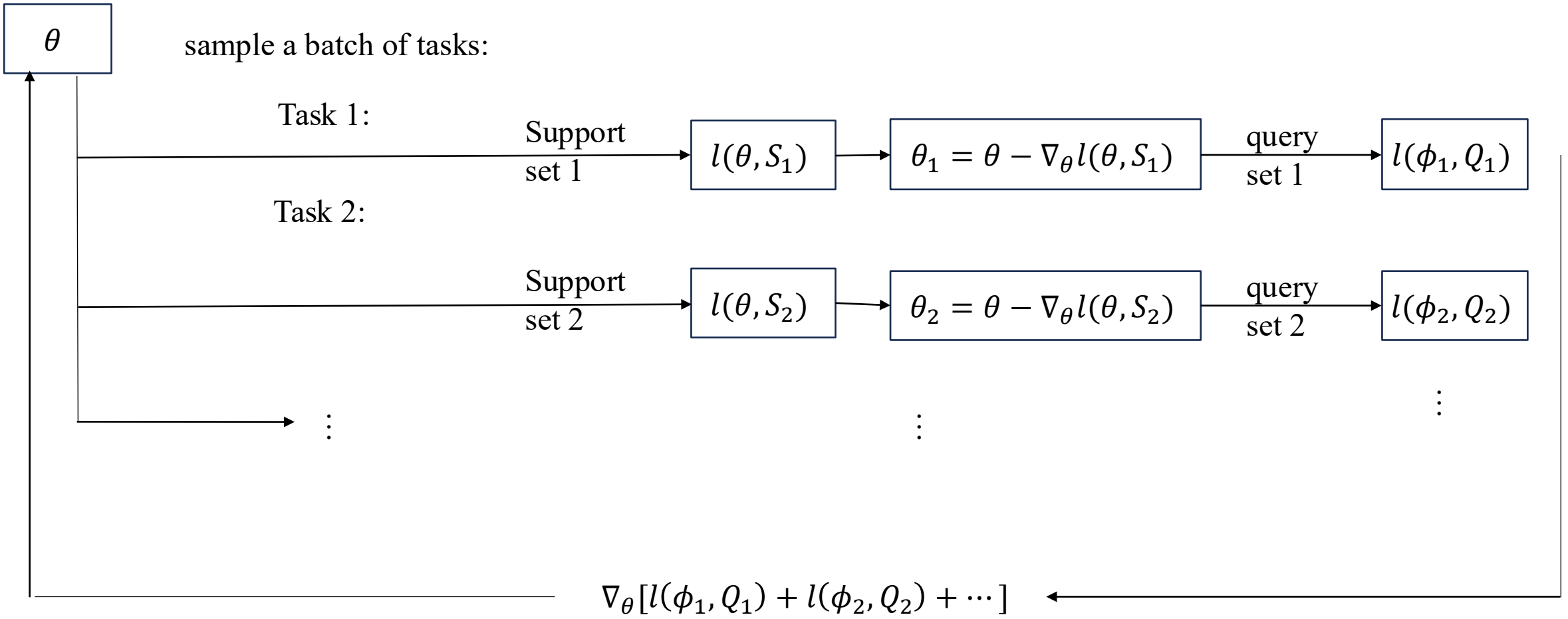
Step 2: Update the shared initialization using the query loss after adaptation.

Algorithm 2 MAML for Few-Shot Supervised Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters
1: randomly initialize θ
2: **while** not done **do**
3: Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
4: **for all** $\mathcal{T}_i$ **do**
Step 1
5: Sample K datapoints $\mathcal{D} = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$
6: Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ using $\mathcal{D}$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation (2) or (3)
7: Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
Step 2
8: Sample datapoints $\mathcal{D}_i' = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$ for the meta-update
9: **end for**
10: Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$ using each $\mathcal{D}_i'$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation 2 or 3
11: **end while**

Optimization-Based—MAML

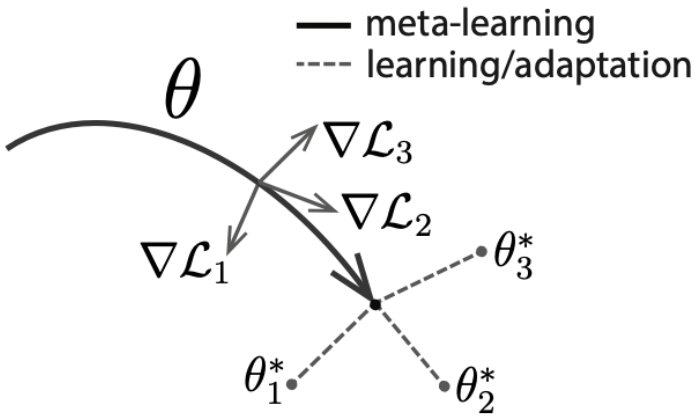
For episode in range (max_episodes):



Optimization-Based Pipeline

1. Learn from many related tasks.
2. Learn a shared model initialization.
3. Use the support set to update the model.
4. Apply the adapted model to the query examples.

Adaptation is performed through gradient updates.



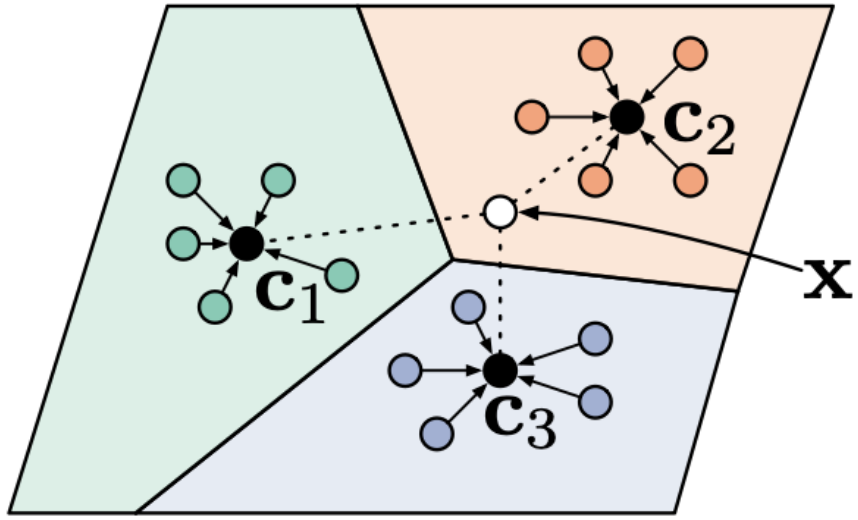
Roadmap

- ✓ Principles of Few-Shot Learning
- ☐ Adaptation Mechanisms
  - ✓ Optimization-based
  - ☐ Metric-based
  - ☐ Amortization-based
  - ☐ In-context learning
- ☐ Choosing an Adaptation Mechanism
- ☐ Practice: From Applications to Few-Shot Problems

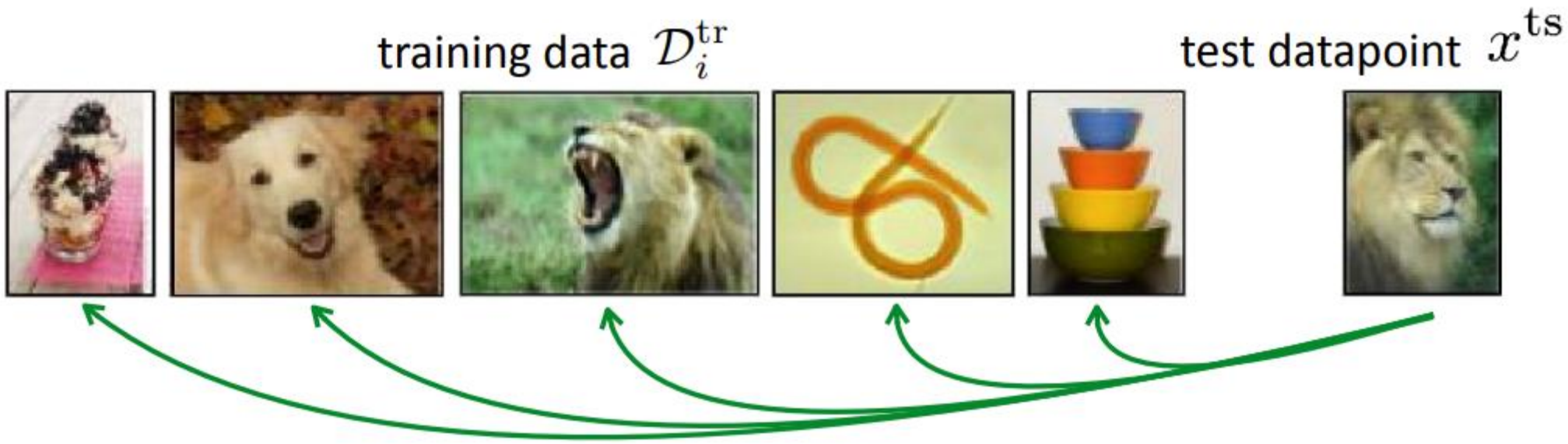
Metric-Based Meta-Learning

learn a transferable representation and similarity metric

- Meta-learn the encoder
- Compare the query embedding with the labeled embeddings

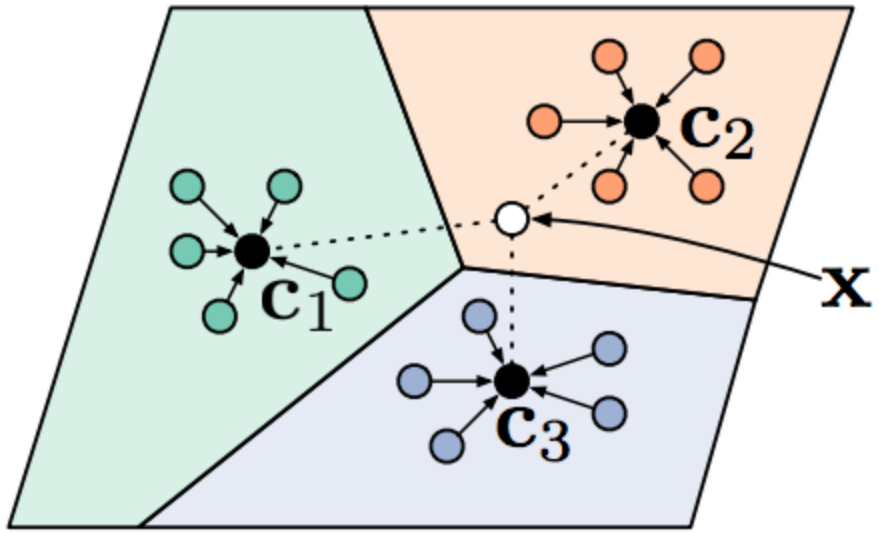


Metric-Based Algorithm



Compare test image with training images
In what space do you compare? With what distance metric?

Metric-Based—Prototypical Network



(a) Few-shot

$$\mathbf{c}_k = \frac{1}{|\mathcal{D}_i^{\mathrm{tr}}|} \sum_{(x,y)\in\mathcal{D}_i^{\mathrm{tr}}} f_\theta(x)$$

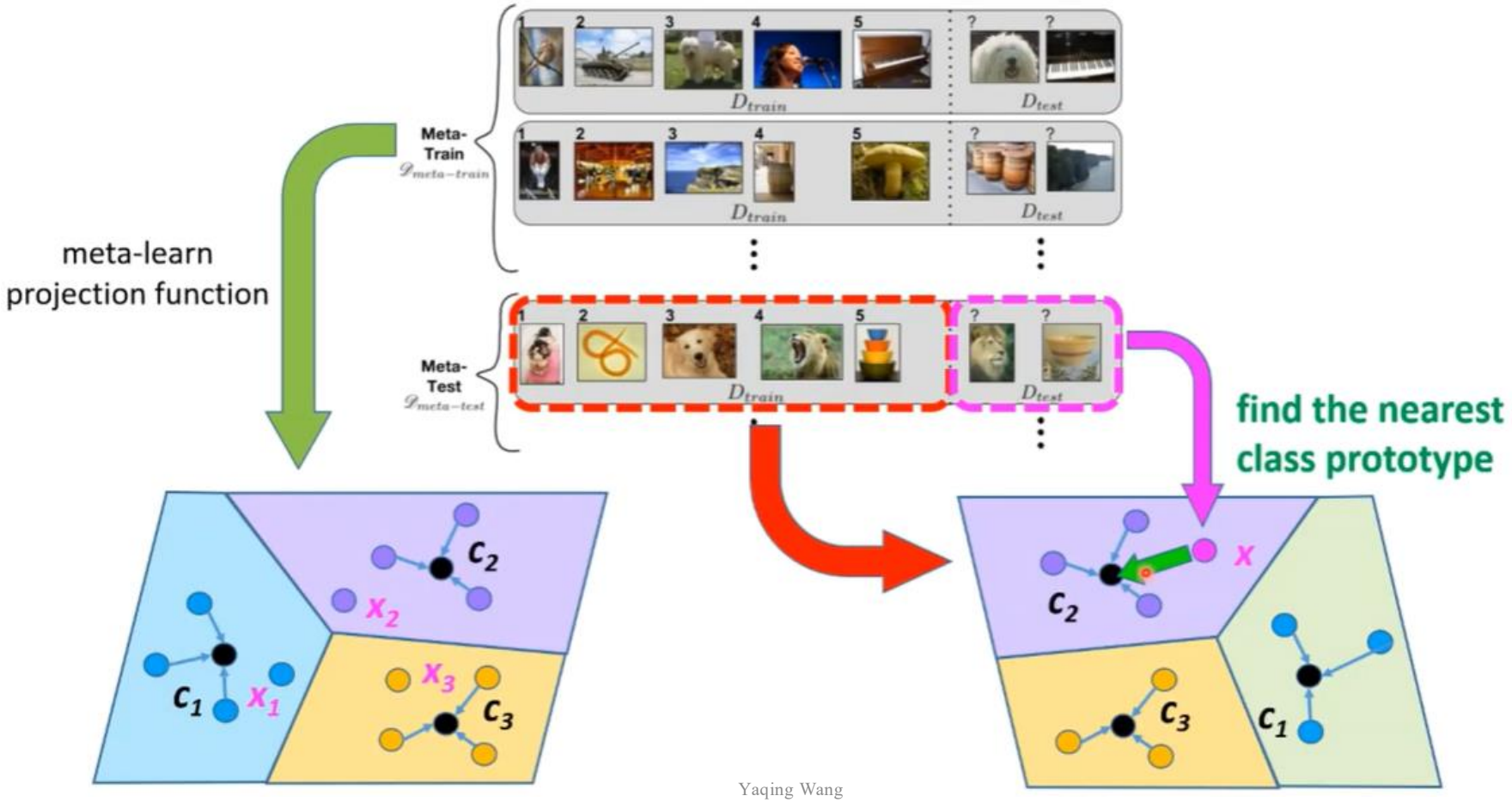
$$p_\theta(y=k|x) = \frac{\exp(-d(f_\theta(x), \mathbf{c}_k))}{\sum_{k'} \exp(-d(f_\theta(x), \mathbf{c}_{k'}))}$$

d: Euclidean, or cosine distance

Snell J, Swersky K, Zemel R. Prototypical networks for few-shot learning[J]. Advances in neural information processing systems, 2017, 30.

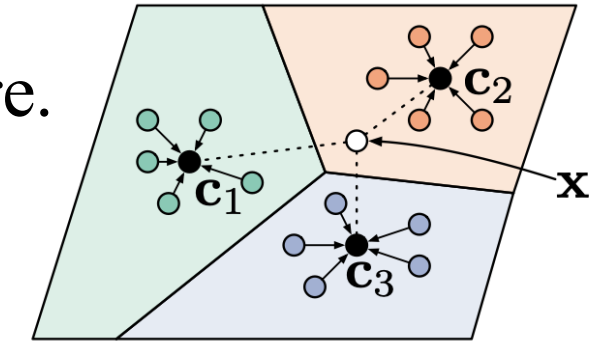
Metric-Based—Prototypical Network

Testing Prototypical Network



Metric-Based Pipeline

1. Learn from many related tasks.
2. Learn a shared representation and similarity measure.
3. Use the support set to represent the new task.
4. Make predictions according to learned similarities.



Adaptation is performed by constructing a task-specific classifier.

Roadmap

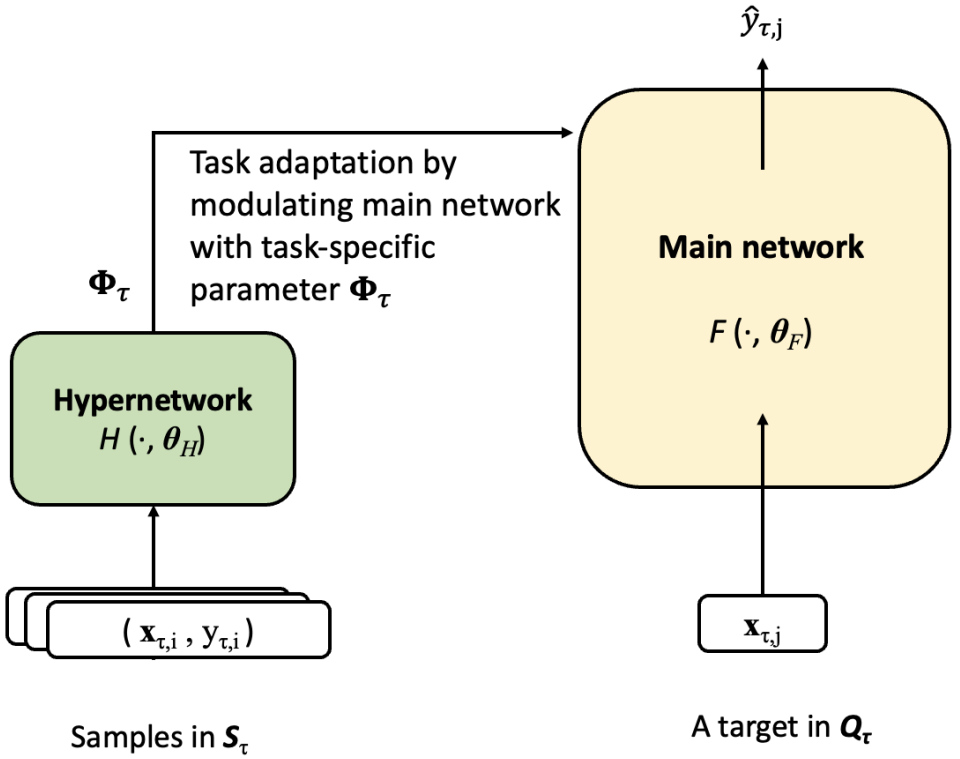
- ✓ Principles of Few-Shot Learning
- ☐ Adaptation Mechanisms
  - ✓ Optimization-based
  - ✓ Metric-based
  - ☐ Amortization-based
  - ☐ In-context learning
- ☐ Choosing an Adaptation Mechanism
- ☐ Practice: From Applications to Few-Shot Problems

Amortization-Based Meta-Learning

Meta-learn a task-conditioning network that generates a small set of task-specific parameters.

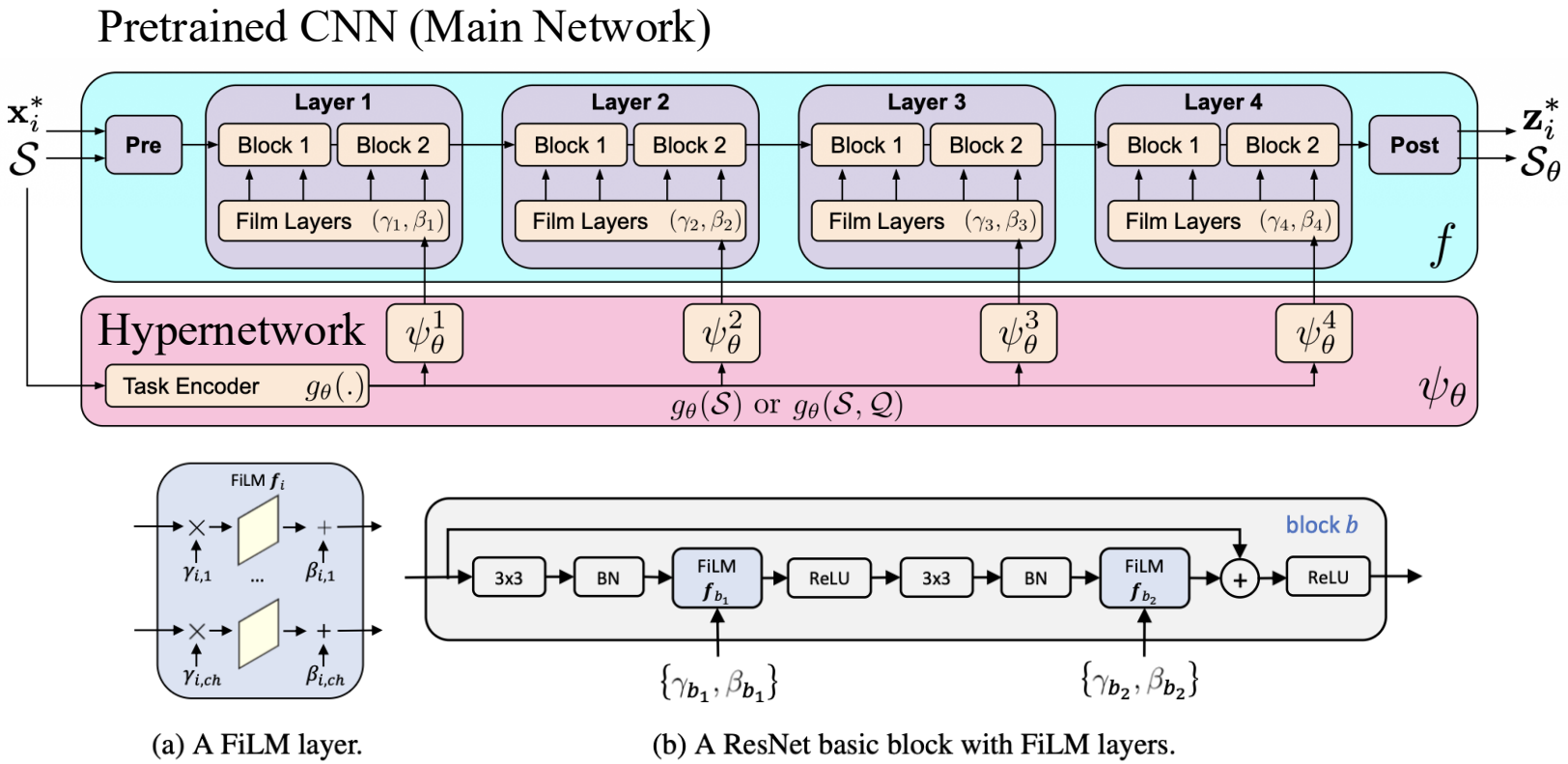
The Architecture

- Hypernetwork: a set encoder
- Main-network (backbone): predicto
- Modulation function



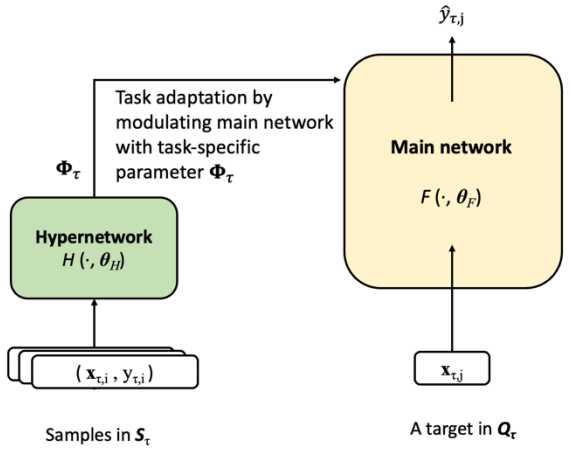
Amortization-Based—(Simple) CNAPS

- Generate task-specific modulation parameters for the main network
- Test-time adaptation becomes a forward pass



Requeima J, Gordon J, Bronskill J, et al. Fast and flexible multi-task classification using conditional neural adaptive processes[J]. Advances in neural information processing systems, 2019, 32.
Bateni P, Barber J, Goyal R, et al. Beyond simple meta-learning: Multi-purpose models for multi-domain, active and continual few-shot learning[J]. arXiv preprint arXiv:2201.05151, 2022.

Amortization-Based Pipeline



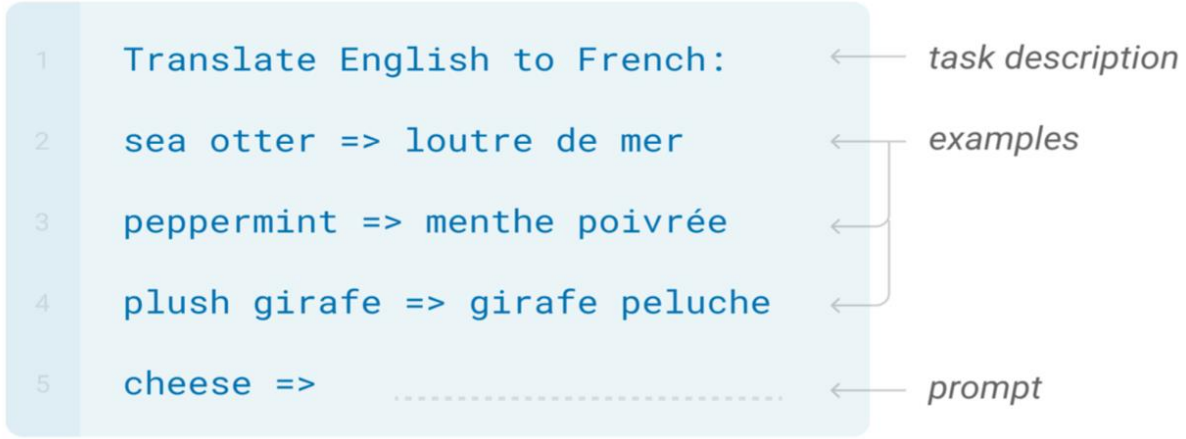
1. Learn from many related tasks.
2. Learn a shared model and a task-conditioning mechanism.
3. Use the support set to generate a small set of task-specific parameters.
4. Use these parameters to adapt the shared model for the query examples.

Adaptation is produced directly, without iterative gradient updates.

Roadmap

- ✓ Principles of Few-Shot Learning
- ☐ Adaptation Mechanisms
  - ✓ Optimization-based
  - ✓ Metric-based
  - ✓ Amortization-based
  - ☐ In-context learning
- ☐ Choosing an Adaptation Mechanism
- ☐ Practice: From Applications to Few-Shot Problems

In-Context Learning (ICL)



- ✓ Larger language models showed increasingly strong task adaptation from the context alone.

Language models are few-shot learners

T Brown, B Mann, N Ryder… - Advances in neural …, 2020 - proceedings.neurips.cc

We demonstrate that scaling up language models greatly improves task-agnostic, few-shot performance, sometimes even becoming competitive with prior state-of-the-art fine-tuning approaches. Specifically, we train GPT-3, an autoregressive language model with 175 billion parameters, 10x more than any previous non-sparse language model, and test its performance in the few-shot setting. For all tasks, GPT-3 is applied without any gradient updates or fine-tuning, with tasks and few-shot demonstrations specified purely via text …

☆ 保存 99 引用 被引用次数：79539 相关文章 所有 37 个版本 »

Language Models are Few-Shot Learners

Tom B. Brown* **Benjamin Mann*** **Nick Ryder*** **Melanie Subbiah***

Jared Kaplan† **Prafulla Dhariwal** **Arvind Neelakantan** **Pranav Shyam**

Girish Sastry **Amanda Askell** **Sandhini Agarwal** **Ariel Herbert-Voss**

Gretchen Krueger **Tom Henighan** **Rewon Child** **Aditya Ramesh**

Daniel M. Ziegler **Jeffrey Wu** **Clemens Winter**

Christopher Hesse **Mark Chen** **Eric Sigler** **Mateusz Litwin** **Scott Gray**

Benjamin Chess **Jack Clark** **Christopher Berner**

Sam McCandlish **Alec Radford** **Ilya Sutskever** **Dario Amodei**

Abstract

We demonstrate that scaling up language models greatly improves task-agnostic, few-shot performance, sometimes even becoming competitive with prior state-of-the-art fine-tuning approaches. Specifically, we train GPT-3, an autoregressive language model with 175 billion parameters, 10x more than any previous non-sparse language model, and test its performance in the few-shot setting. For all tasks, GPT-3 is applied without any gradient updates or fine-tuning, with tasks and few-shot demonstrations specified purely via text interaction with the model. GPT-3 achieves strong performance on many NLP datasets, including translation, question-answering, and cloze tasks. We also identify some datasets where GPT-3's few-shot learning still struggles, as well as some datasets where GPT-3 faces methodological issues related to training on large web corpora.

1 Introduction

NLP has shifted from learning task-specific representations and designing task-specific architectures to using task-agnostic pre-training and task-agnostic architectures. This shift has led to substantial progress on many challenging NLP tasks such as reading comprehension, question answering, textual entailment, among others. Even though the architecture and initial representations are now task-agnostic, a final task-specific step remains: fine-tuning on a large dataset of examples to adapt a task agnostic model to perform a desired task.

Recent work [RWC+19] suggested this final step may not be necessary. [RWC+19] demonstrated that a single pretrained language model can be zero-shot transferred to perform standard NLP tasks

*Equal contribution
†Johns Hopkins University, OpenAI

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

A Task Can Be Specified by Instructions and Examples

- Zero-shot： instruction only
- One-shot： instruction + one example
- Few-shot： instruction + several examples

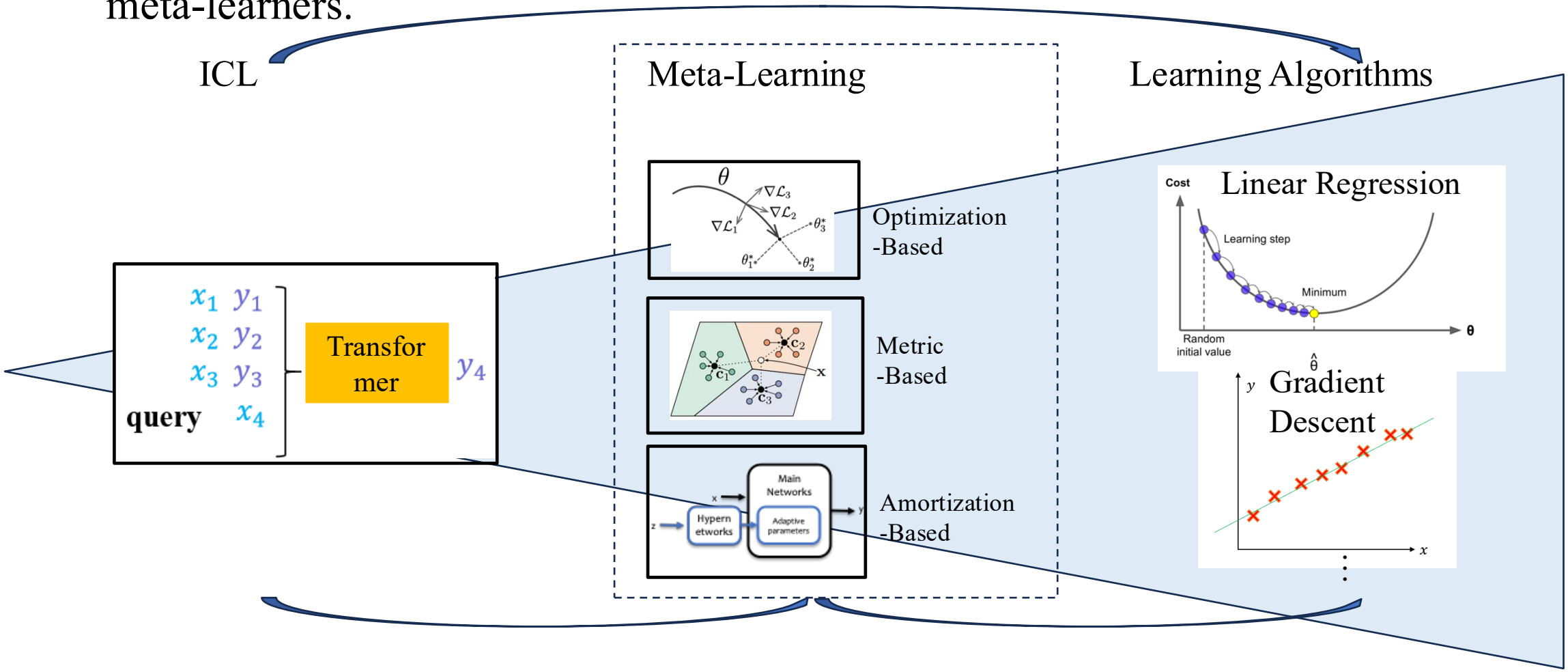
Context specifies the task through instructions, demonstrations, or both.

ICL and Meta-Learning Share an Adaptation View

The key difference is where adaptation happens: explicitly through a learned mechanism, or implicitly within the forward pass.

Why ICL Models Are Good Few-Shot Learners?

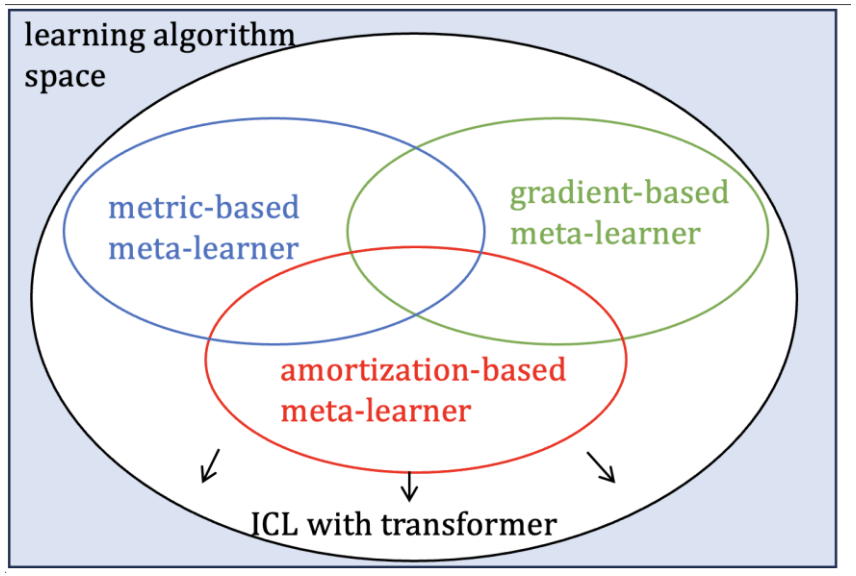
We study ICL by comparing it with optimization-, metric-, and amortization-based meta-learners.

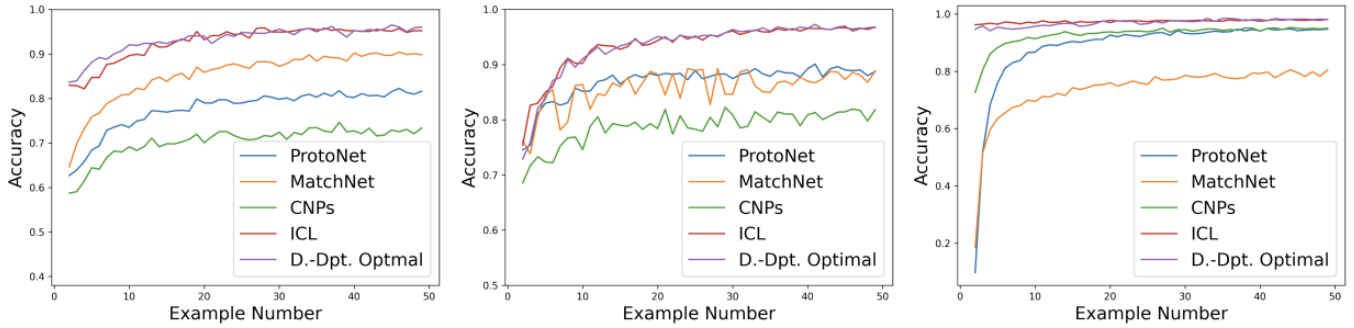


"Why In-Context Learning Models are Good Few-Shot Learners?." in International Conference on Learning Representations (ICLR), 2025.

ICL with Transformers as Meta-Learners

With suitable parameters, a Transformer can represent gradient-based, metric-based, and amortized adaptation.





(a) Pair-wise metric-based tasks. (b) Class-prototype metric-based. (c) Amortization-based.

- ✓ When trained on one task family, ICL recovers the corresponding learning algorithm.
- ✓ When trained on mixed task families, ICL learns to select a data-dependent adaptation rule.

ICL Pipeline

1. Learn from diverse data and tasks during pretraining.
2. Learn a shared model that can infer a task from context.
3. Provide task instructions and/or demonstrations as context.
4. Use the context to make predictions for new queries without updating model parameters.

Adaptation is performed through context-conditioned computation, without parameter updates.

Roadmap

- ✓ Principles of Few-Shot Learning
- ✓ Adaptation Mechanisms
  - ✓ Optimization-based
  - ✓ Metric-based
  - ✓ Amortization-based
  - ✓ In-context learning
- ☐ Choosing an Adaptation Mechanism
- ☐ Practice: From Applications to Few-Shot Problems

Four Ways to Adapt from Limited Evidence

Method	What the evidence changes	Test-time mechanism
Optimization-based Meta-Learning	Model parameters	Gradient updates
Metric-based Meta-Learning	Task-specific classifier	Similarity-based construction
Amortization-based Meta-Learning	A small set of task-specific parameters	Task-conditioning network
In-context learning	Internal states and predictions	Context-conditioned inference

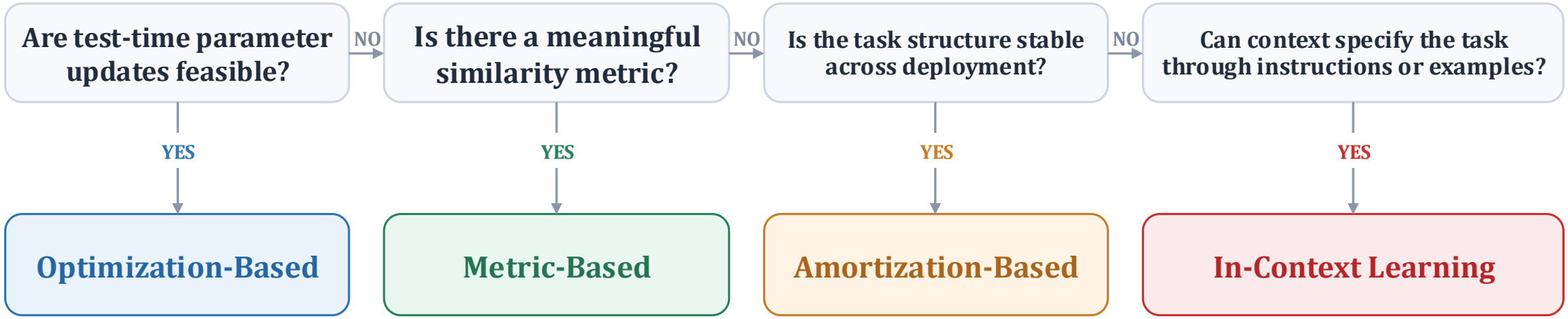
The key distinction is how limited evidence is converted into task adaptation.

No Adaptation Mechanism Dominates Everywhere

Method	Main Strength	Main limitation
Optimization-based Meta-Learning	Flexible task-specific adaptation	Slow and sensitive to optimization
Metric-based Meta-Learning	Fast and simple	Depends on a meaningful representation and metric
Amortization-based Meta-Learning	Fast, structured task conditioning	Limited by the meta-training task distribution
In-context learning	Flexible task specification without parameter updates	Sensitive to context selection, order and formatting

Flexibility, speed and robustness cannot usually be maximized at the same time.

Choosing an Adaptation Mechanism



- If none applies: reformulate the task interface or use a hybrid mechanism.
- Large distribution shift? Prefer explicit adaptation or a hybrid mechanism.

Roadmap

- ✓ Principles of Few-Shot Learning
- ✓ Adaptation Mechanisms
  - ✓ Optimization-based
  - ✓ Metric-based
  - ✓ Amortization-based
  - ✓ In-context learning
- ✓ Choosing an Adaptation Mechanism
- ☐ Practice: From Applications to Few-Shot Problems

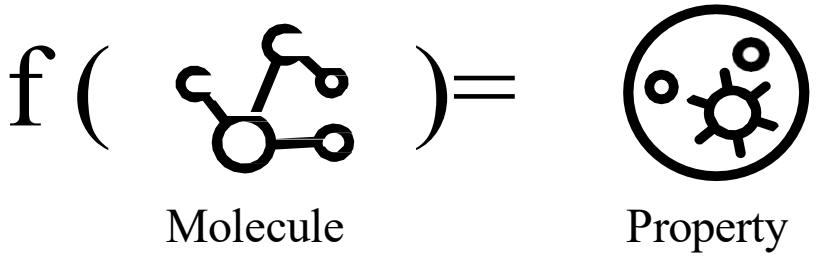
Practices: Modeling Applications as Few-Shot Learning

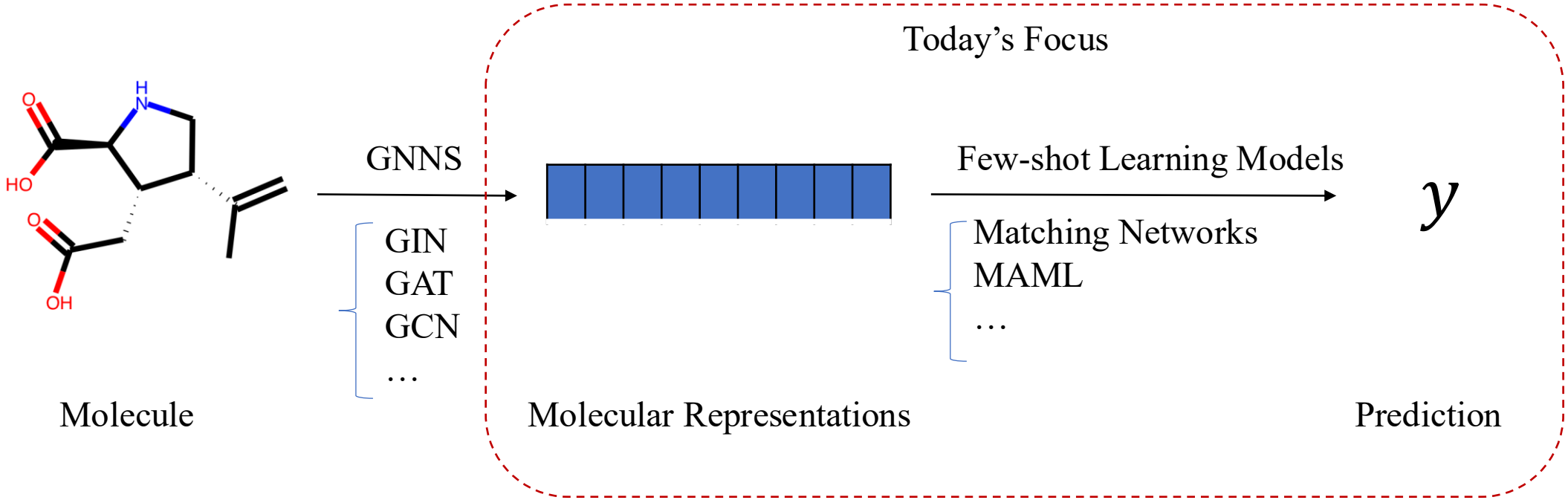
A General Modeling Recipe

1. Identify Label Scarcity
2. Define the Task
3. Construct Support and Query Sets
4. Extract the Task-Specific Insight
5. Design and Validate the Adaptation Strategy

We illustrate this recipe through two application cases

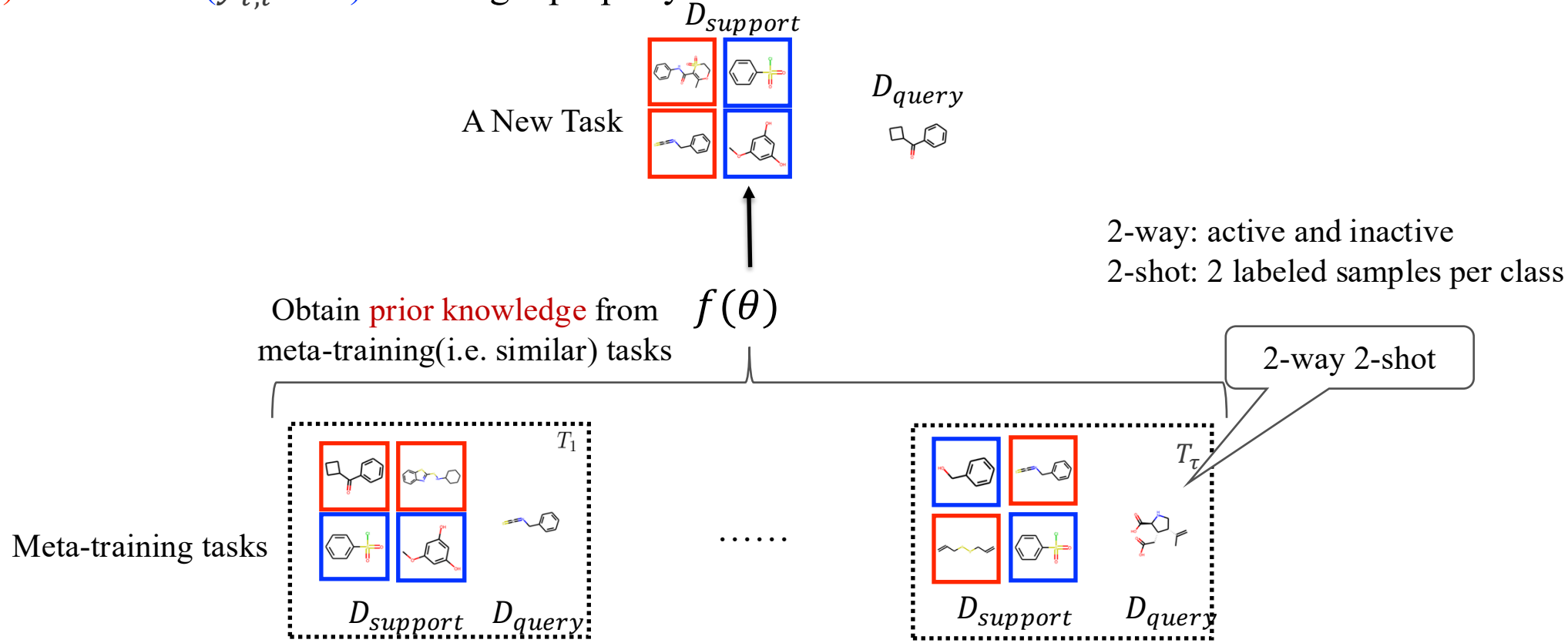
Molecular Property Prediction (MPP)



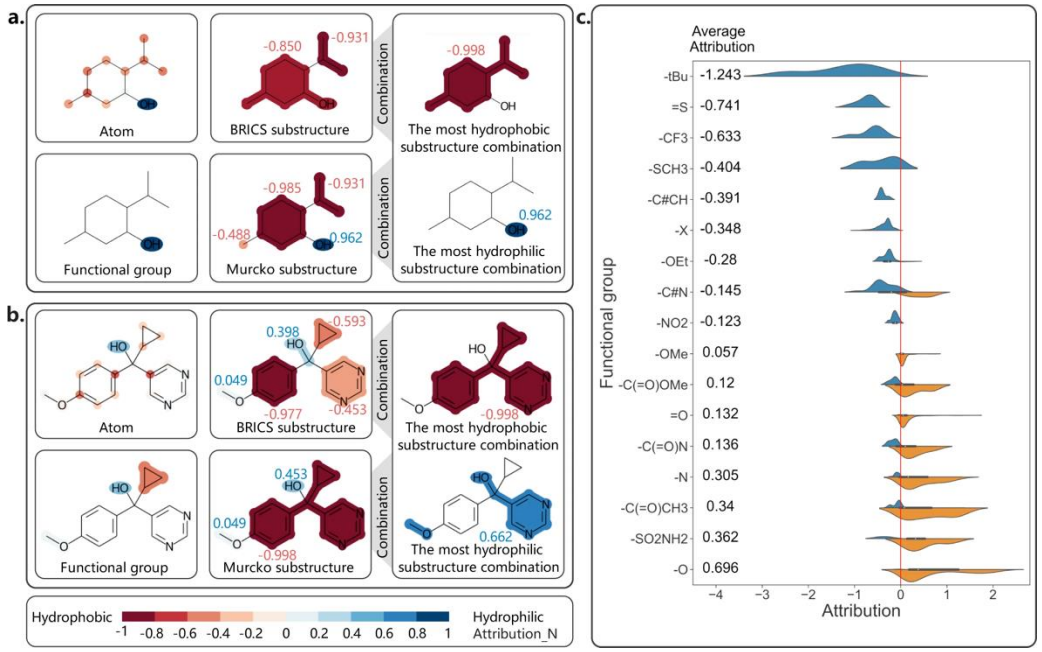


MPP as a Few-Shot Learning Problem

- Learning a predictor from a bunch of property prediction tasks and generalize to predict **new properties with a few labeled** molecules
- Each task corresponds to an experimental assay testing on whether each molecule $x_{\tau,i}$ is active ($y_{\tau,i} = 1$) or inactive ($y_{\tau,i} = 0$) on a target property

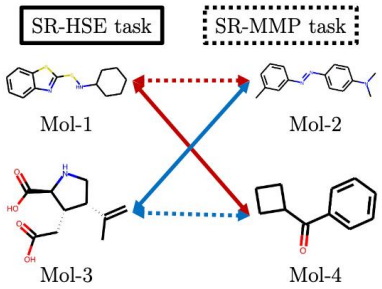


Our Insight



https://www.nature.com/articles/s41467-023-38192-3/figures/1

- Different molecular structures correspond to different properties



Molecules		Label	
ID	SMILES	SR-HSE	SR-MMP
Mol-1	c1ccc2sc(SNC3CCCCC3)nc2c1	1	1
Mol-2	Cc1cccc(/N=N/c2ccc(N(C)C)cc2)c1	0	1
Mol-3	C=C(C)[C@H]1CN[C@H](C(=O)O)[C@H]1CC(=O)O	0	0
Mol-4	O=C(c1ccccc1) C1CCC1	1	0

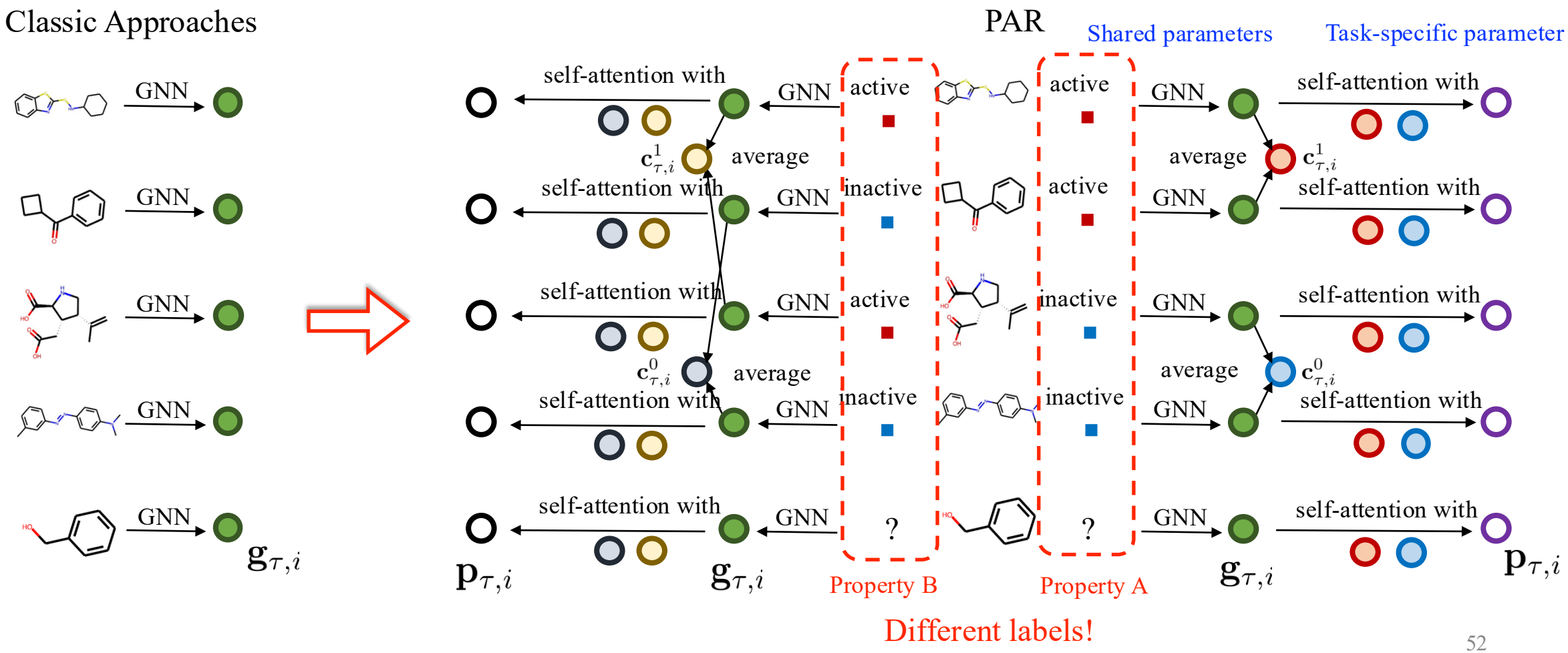
Figure 1: Examples of relation graphs for the same molecules coexisting in two tasks of Tox21. Red (blue) edges mean the connected molecules are both active (inactive) on the target property.

- Relationships between molecules vary with target properties

"Property-Aware Relation Networks for Few-Shot Molecular Property Prediction", in Neural Information Processing Systems (NeurIPS), 2021
"Property-Aware Relation Networks for Few-Shot Molecular Property Prediction", in IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2024.

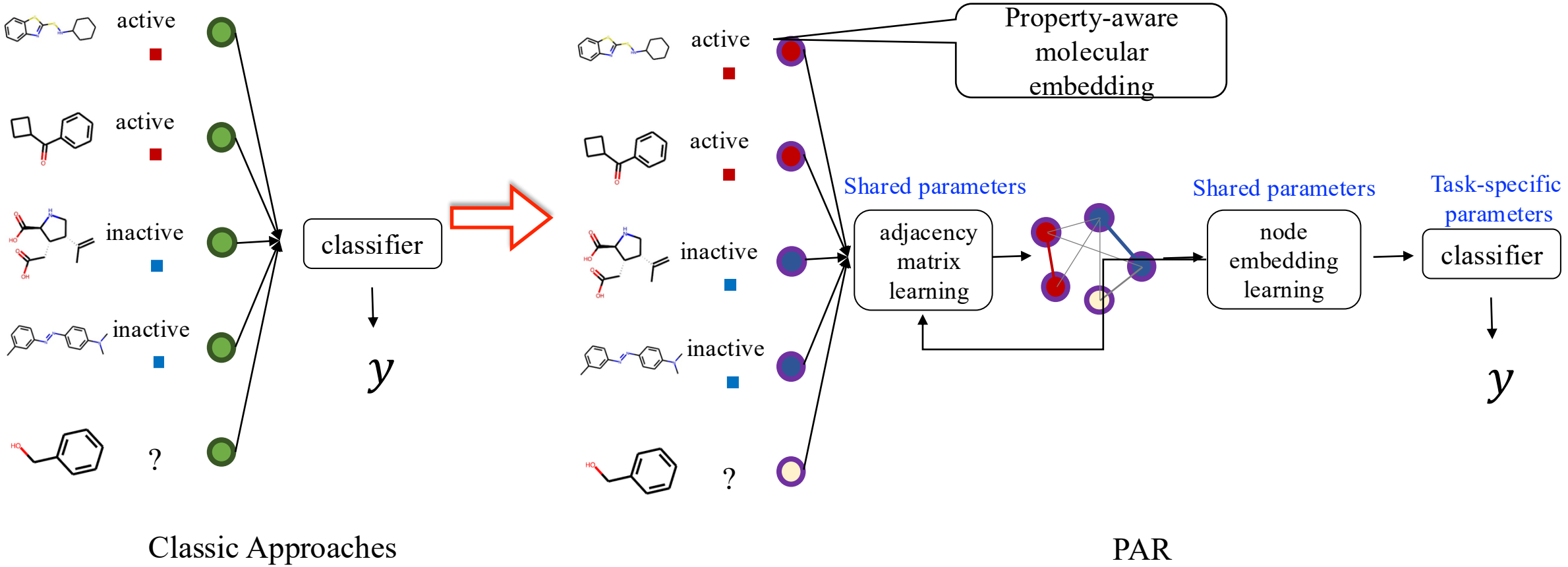
PAR

- The same molecule shows different properties in different MPP tasks
- The general molecular embeddings are transformed to be property-aware



PAR

- Molecular relationship changes across target properties

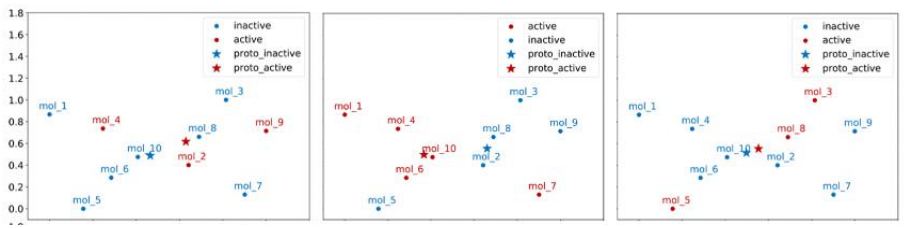


Results

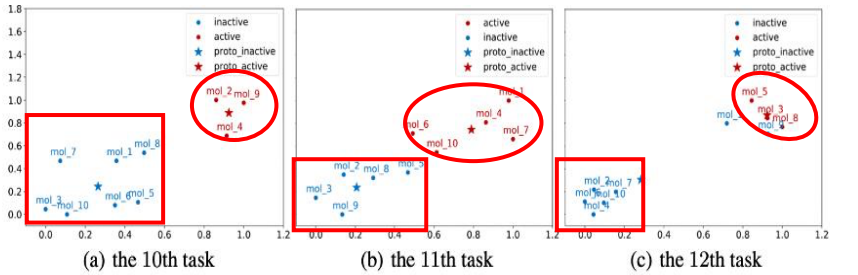
Property-aware adaptation consistently improves performance, especially in the lowest-shot settings.

Table 2: ROC-AUC scores on benchmark molecular property prediction datasets. The best results (according to the pairwise t-test with 95% confidence) are highlighted in gray. Methods which use pretrained graph-based molecular encoder are marked in green.

	Method	Tox21		SIDER		MUV		ToxCast	
		10-shot	1-shot	10-shot	1-shot	10-shot	1-shot	10-shot	1-shot
Trained from Scratch	Siamese	$80.40_{(0.35)}$	$65.00_{(1.58)}$	$71.10_{(4.32)}$	$51.43_{(3.31)}$	$59.96_{(5.13)}$	$50.00_{(0.17)}$	-	-
	ProtoNet	$74.98_{(0.32)}$	$65.58_{(1.72)}$	$64.54_{(0.89)}$	$57.50_{(2.34)}$	$65.88_{(4.11)}$	$58.31_{(3.18)}$	$63.70_{(1.26)}$	$56.36_{(1.54)}$
	MAML	$80.21_{(0.24)}$	$75.74_{(0.48)}$	$70.43_{(0.76)}$	$67.81_{(1.12)}$	$63.90_{(2.28)}$	$60.51_{(3.12)}$	$66.79_{(0.85)}$	$65.97_{(5.04)}$
	TPN	$76.05_{(0.24)}$	$60.16_{(1.18)}$	$67.84_{(0.95)}$	$62.90_{(1.38)}$	$65.22_{(5.82)}$	$50.00_{(0.51)}$	$62.74_{(1.45)}$	$50.01_{(0.05)}$
	EGNN	$81.21_{(0.16)}$	$79.44_{(0.22)}$	$72.87_{(0.73)}$	$70.79_{(0.95)}$	$65.20_{(2.08)}$	$62.18_{(1.76)}$	$63.65_{(1.57)}$	$61.02_{(1.94)}$
	IterRefLSTM	$81.10_{(0.17)}$	$80.97_{(0.10)}$	$69.63_{(0.31)}$	$71.73_{(0.14)}$	$49.56_{(5.12)}$	$48.54_{(3.12)}$	-	-
	PAR	$82.06_{(0.12)}$	$80.46_{(0.13)}$	$74.68_{(0.31)}$	$71.87_{(0.48)}$	$66.48_{(2.12)}$	$64.12_{(1.18)}$	$69.72_{(1.63)}$	$67.28_{(2.90)}$
Pretrained	Pre-GNN	$82.14_{(0.08)}$	$81.68_{(0.09)}$	$73.96_{(0.08)}$	$73.24_{(0.12)}$	$67.14_{(1.58)}$	$64.51_{(1.45)}$	$73.68_{(0.74)}$	$72.90_{(0.84)}$
	Meta-MGNN	$82.97_{(0.10)}$	$82.13_{(0.13)}$	$75.43_{(0.21)}$	$73.36_{(0.32)}$	$68.99_{(1.84)}$	$65.54_{(2.13)}$	-	-
	Pre-PAR	$84.93_{(0.11)}$	$83.01_{(0.09)}$	$78.08_{(0.16)}$	$74.46_{(0.29)}$	$69.96_{(1.37)}$	$66.94_{(1.12)}$	$75.12_{(0.84)}$	$73.63_{(1.00)}$



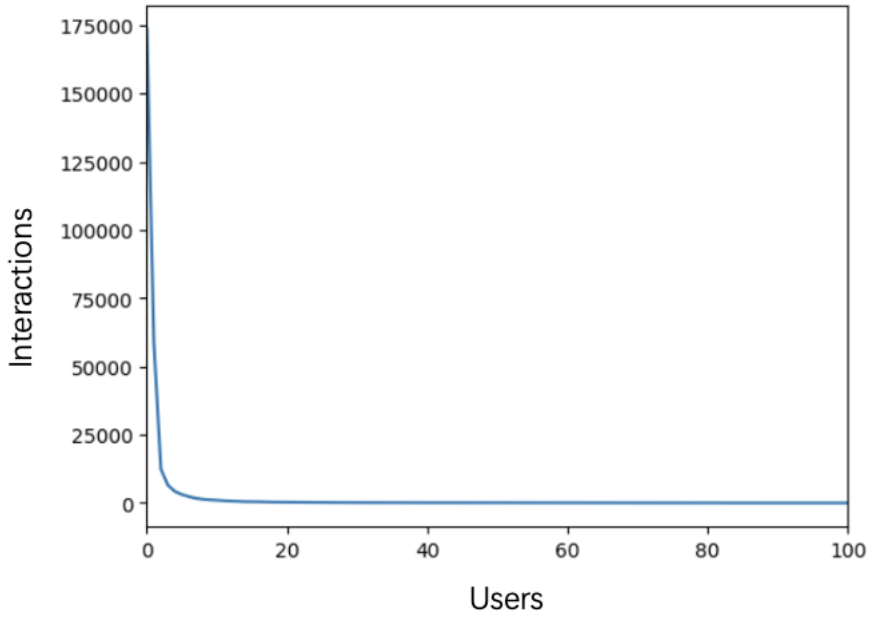
General molecular embeddings



Property-aware molecular embeddings

User Cold-Start Recommendation

- New users continuously enter recommendation systems with only a few observed interactions.
- User activity follows a long-tailed distribution, and most users have limited interaction histories.



Distribution of user-interactions in BookCrossing

Cold-Start Recommendation as a Few-Shot Learning Problem

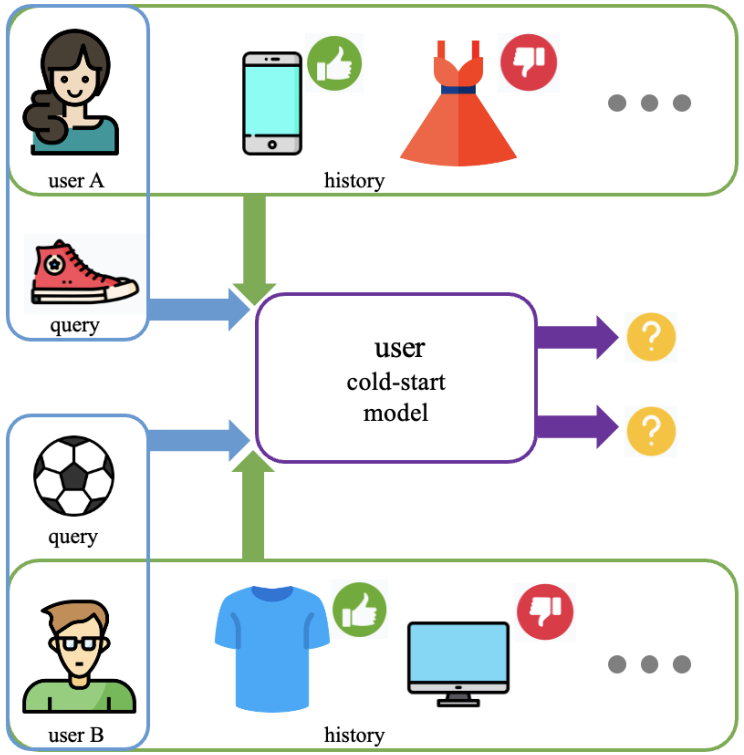
Each task T_i corresponds to a user u_i, with

- a support set $S_i = \{(v_j, y_{i,j})\}_{j=1}^{N}$ containing existing interaction histories
- a query set $Q_i = \{(v_j, y_{i,j})\}_{j=1}^{M}$ containing interactions to predict.

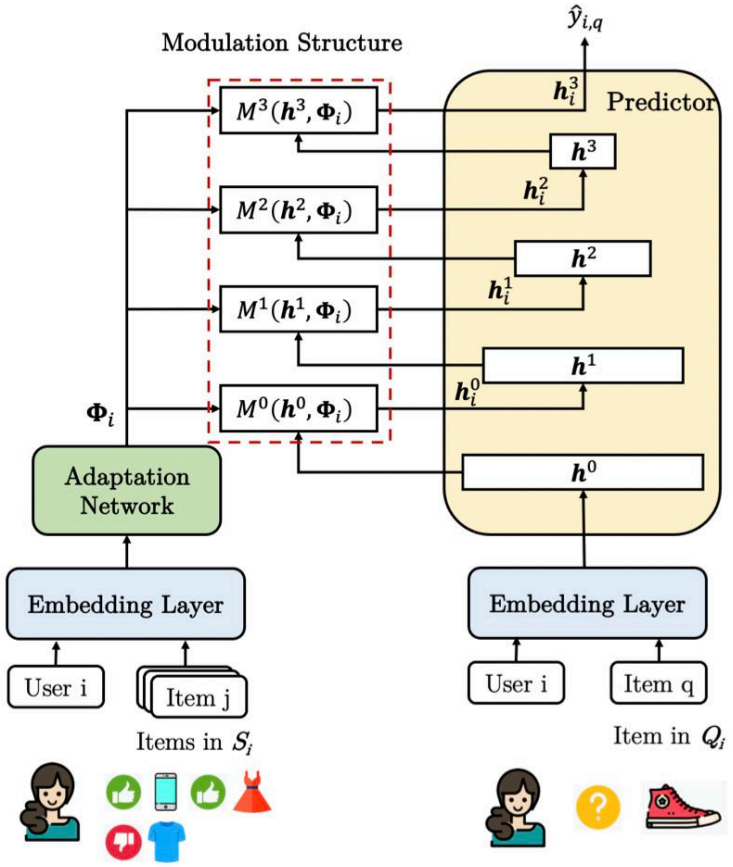
S_i and Q_i are disjoint, and N is small for cold-start users.

Here,

- u_i represents content features of a user (categorical or numerical, e.g., occupation, age),
- v_j represents content features of an item.



ColdNAS



- The embedding layer E with parameter θ_E embeds the categorical features from users and items into dense vectors, i.e., $(\boldsymbol{u}_i, \boldsymbol{v}_j) = E(u_i, v_j; \boldsymbol{\theta}_E)$.
- The adaptation network A with parameter $\boldsymbol{\theta}_A$ takes the support set $\mathcal{S}_i$ for a specific user u_i as input and generates user-specific adaptive parameters, i.e.,

$$\boldsymbol{\Phi}_i = \{\boldsymbol{\phi}_i^k\}_{k=1}^{C} = A(\mathcal{S}_i; \boldsymbol{\theta}_A), \tag{1}$$

  where C is the number of adaptive parameter groups for certain modulation structure.
- The predictor P with parameter $\boldsymbol{\theta}_P$ takes user-specific parameters $\boldsymbol{\phi}_i$ and $v_q \in \mathcal{Q}_i$ from the query set as input, and generate predictions by

$$\hat{y}_{i,j} = P((\boldsymbol{u}_i, \boldsymbol{v}_q), \boldsymbol{\Phi}_i; \boldsymbol{\theta}_P). \tag{2}$$

"ColdNAS: Search to Modulate for User Cold-Start Recommendation", in The Web Conference, 2023.

"Searching to Modulate for Cold-Start Recommendation", in IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2026.

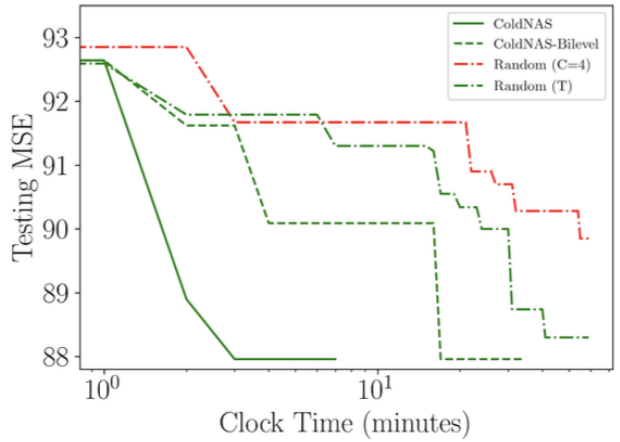
Results

User-specific modulation improves both predictive performance and search efficiency.

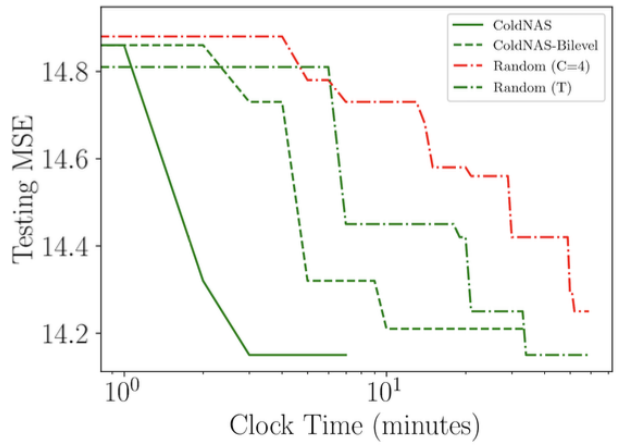
Table 2: Test performance (%) obtained on benchmark datasets. The best results are highlighted in bold and the second-best in italic. For MSE and MAE, smaller value is better. For $nDCG_3$ and $nDCG_5$, larger value is better.

Dataset	Metric	DropoutNet	MeLU	MetaCS	MetaHIN	MAMO	TaNP	ColdNAS-Fixed	ColdNAS
MovieLens	MSE	$100.90_{(0.70)}$	$95.02_{(0.03)}$	$95.05_{(0.04)}$	$91.89_{(0.06)}$	$90.20_{(0.22)}$	$\underline{89.11}_{(0.18)}$	$91.05_{(0.13)}$	$\mathbf{87.96}_{(0.12)}$
	MAE	$85.71_{(0.48)}$	$77.38_{(0.25)}$	$77.42_{(0.26)}$	$75.79_{(0.27)}$	$75.34_{(0.26)}$	$\underline{74.78}_{(0.14)}$	$75.65_{(0.30)}$	$\mathbf{74.29}_{(0.20)}$
	$nDCG_3$	$69.21_{(0.76)}$	$74.43_{(0.59)}$	$74.46_{(0.78)}$	$74.69_{(0.32)}$	$74.95_{(0.13)}$	$\underline{75.60}_{(0.07)}$	$75.11_{(0.09)}$	$\mathbf{76.16}_{(0.03)}$
	$nDCG_5$	$68.43_{(0.48)}$	$73.52_{(0.41)}$	$73.45_{(0.56)}$	$73.63_{(0.22)}$	$73.84_{(0.16)}$	$\underline{74.29}_{(0.12)}$	$73.89_{(0.12)}$	$\mathbf{74.74}_{(0.09)}$
BookCrossing	MSE	$15.38_{(0.23)}$	$15.15_{(0.02)}$	$15.20_{(0.08)}$	$14.76_{(0.07)}$	$14.82_{(0.05)}$	$14.75_{(0.05)}$	$\underline{14.44}_{(0.16)}$	$\mathbf{14.15}_{(0.08)}$
	MAE	$3.75_{(0.01)}$	$3.68_{(0.01)}$	$3.66_{(0.01)}$	$3.50_{(0.01)}$	$3.51_{(0.02)}$	$\underline{3.48}_{(0.01)}$	$3.49_{(0.02)}$	$\mathbf{3.40}_{(0.01)}$
	$nDCG_3$	$77.66_{(0.18)}$	$\underline{77.69}_{(0.15)}$	$77.68_{(0.12)}$	$77.66_{(0.19)}$	$77.68_{(0.09)}$	$77.48_{(0.06)}$	$77.65_{(0.09)}$	$\mathbf{77.83}_{(0.01)}$
	$nDCG_5$	$80.87_{(0.15)}$	$81.10_{(0.15)}$	$80.97_{(0.09)}$	$80.95_{(0.04)}$	$81.01_{(0.05)}$	$\underline{81.16}_{(0.21)}$	$81.12_{(0.06)}$	$\mathbf{81.32}_{(0.10)}$
Last.fm	MSE	$21.91_{(0.38)}$	$21.69_{(0.34)}$	$21.68_{(0.12)}$	$\underline{21.43}_{(0.23)}$	$21.64_{(0.10)}$	$21.58_{(0.20)}$	$21.62_{(0.16)}$	$\mathbf{20.91}_{(0.05)}$
	MAE	$43.02_{(0.52)}$	$42.28_{(1.21)}$	$42.28_{(0.76)}$	$\underline{42.07}_{(0.49)}$	$42.30_{(0.28)}$	$42.15_{(0.56)}$	$42.32_{(0.34)}$	$\mathbf{41.78}_{(0.24)}$
	$nDCG_3$	$75.13_{(0.48)}$	$80.15_{(2.09)}$	$80.81_{(0.97)}$	$\underline{82.01}_{(0.56)}$	$80.73_{(0.80)}$	$81.03_{(0.36)}$	$80.77_{(0.32)}$	$\mathbf{82.80}_{(0.69)}$
	$nDCG_5$	$69.03_{(0.31)}$	$75.03_{(0.68)}$	$75.01_{(0.64)}$	$\underline{75.98}_{(0.33)}$	$75.45_{(0.29)}$	$\underline{75.98}_{(0.41)}$	$75.48_{(0.21)}$	$\mathbf{76.77}_{(0.10)}$

	M^0	M^1	M^2	M^3
MovieLens	$\min(\max(\boldsymbol{h}^0, \boldsymbol{\phi}_i^{0,1}), \boldsymbol{\phi}_i^{0,2}) + \boldsymbol{\phi}_i^{0,3}$	$\boldsymbol{h}^1 + \boldsymbol{\phi}_i^{1,1}$	$\boldsymbol{h}^2$	$\boldsymbol{h}^3$
BookCrossing	$\min(\boldsymbol{h}^0, \boldsymbol{\phi}_i^{0,1})$	$\boldsymbol{h}^1 + \boldsymbol{\phi}_i^{1,1}$	$\boldsymbol{h}^2 \odot \boldsymbol{\phi}_i^{2,1} + \boldsymbol{\phi}_i^{2,2}$	$\boldsymbol{h}^3$
Last.fm	$\boldsymbol{h}^0 + \boldsymbol{\phi}_i^{0,1}$	$\boldsymbol{h}^1 + \boldsymbol{\phi}_i^{1,1}$	$\max(\boldsymbol{h}^2, \boldsymbol{\phi}_i^{2,1}) + \boldsymbol{\phi}_i^{2,2}$	$\boldsymbol{h}^3$
ColdNAS-Fixed	$\boldsymbol{h}^0 \odot \boldsymbol{\phi}_i^{0,1} + \boldsymbol{\phi}_i^{0,2}$	$\boldsymbol{h}^1 \odot \boldsymbol{\phi}_i^{1,1} + \boldsymbol{\phi}_i^{1,2}$	$\boldsymbol{h}^2 \odot \boldsymbol{\phi}_i^{2,1} + \boldsymbol{\phi}_i^{2,2}$	$\boldsymbol{h}^3 \odot \boldsymbol{\phi}_i^{3,1} + \boldsymbol{\phi}_i^{3,2}$



(a) MovieLens.



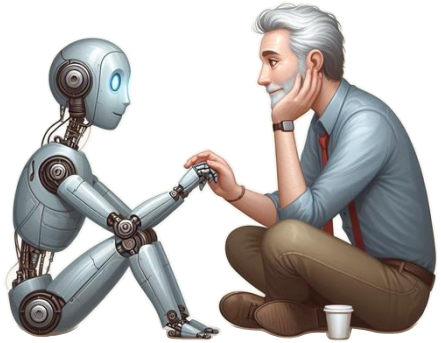
(b) BookCrossing.

Summary: Learning with Limited Labels

- ✓ Principles of Few-Shot Learning
- ✓ Adaptation Mechanisms
  - ✓ Optimization-based
  - ✓ Metric-based
  - ✓ Amortization-based
  - ✓ In-context learning
- ✓ Choosing an Adaptation Mechanism
- ✓ Practice: From Applications to Few-Shot Problems

The key is not merely choosing a method, but formulating the right task and designing the right adaptation strategy

Thanks! Questions?



Know More About Me at wangyaqing.github.io/

Contact Me at wangyaqing@bimsa.cn