SHANGHAI JIAO TONG UNIVERSITY

DoubleAdapt: A Meta-Learning Approach to Incremental Learning for Stock Trend Forecasting

Lifan Zhao, Shuming Kong, Yanyan Shen

Shanghai Jiao Tong University

Stock Trend Forecasting

Train a forecast model (*e.g.*, MLP or LSTM) for precise predictions of future stock trends

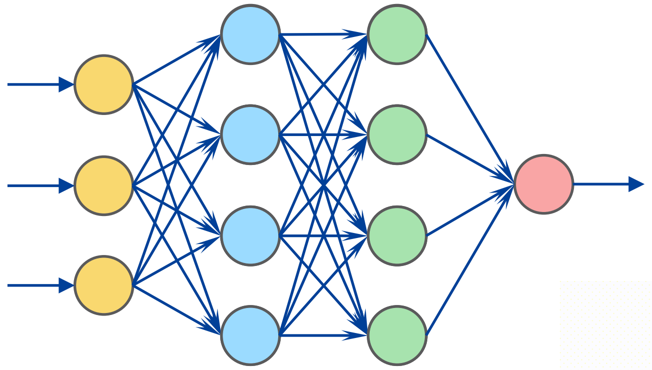
$F(\cdot\,;\theta)$

time series of stock prices

or

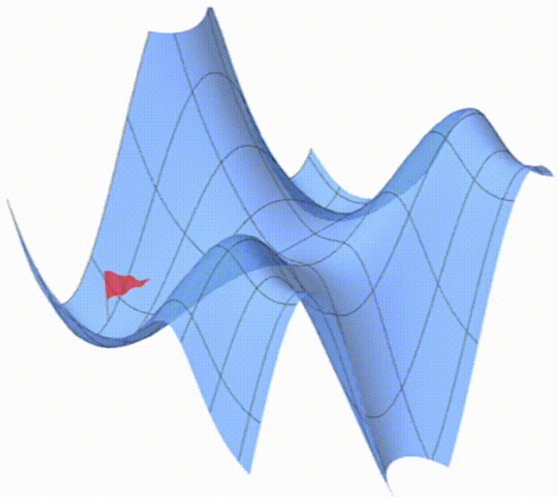
hand-crafted alpha signals

$\mathbf{x}$



y

stock price change rates
(*e.g.*, in the next day)



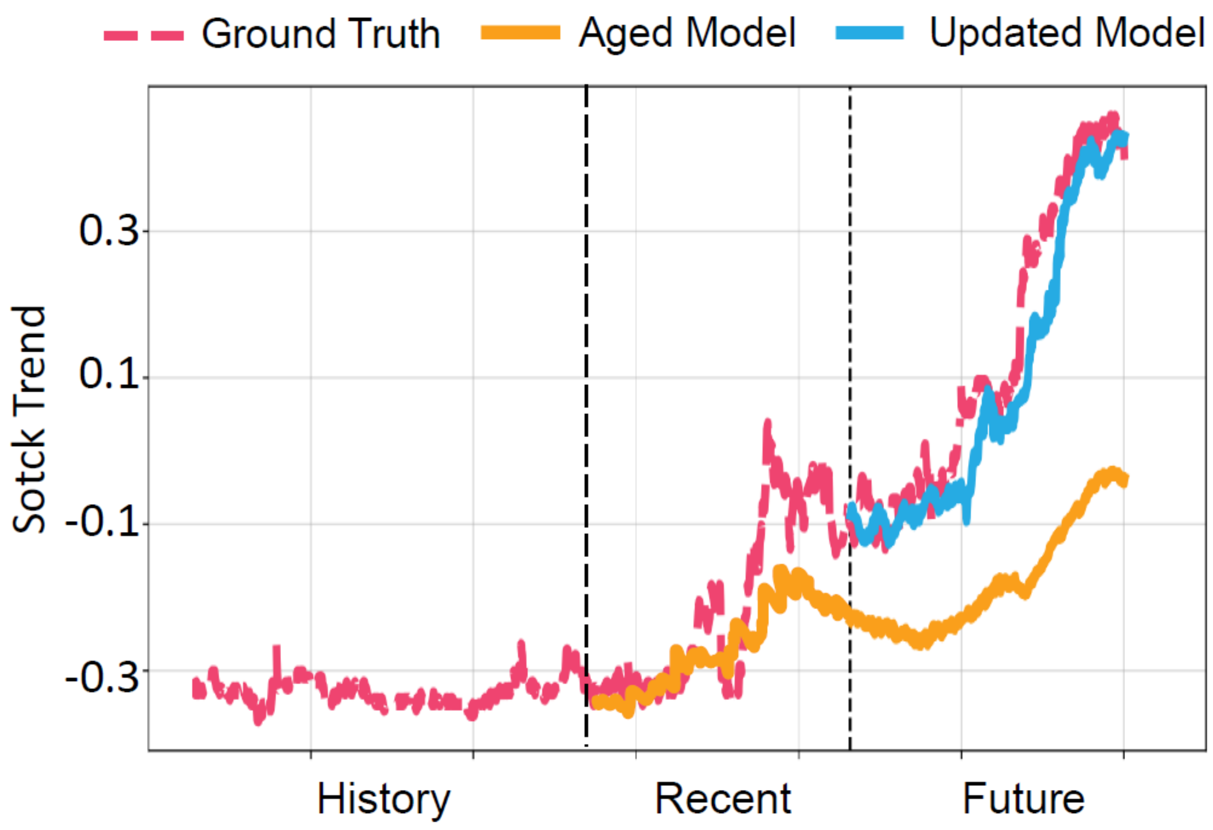
Optimization objective: $\min \sum_{(\mathbf{x},y)\in\mathcal{D}_{\text{train}}} (F(\mathbf{x};\theta) - y)^2$

Gradient descent: $\theta \leftarrow \theta - \alpha \frac{\partial (F(\mathbf{x};\theta)-y)^2}{\partial \theta}$

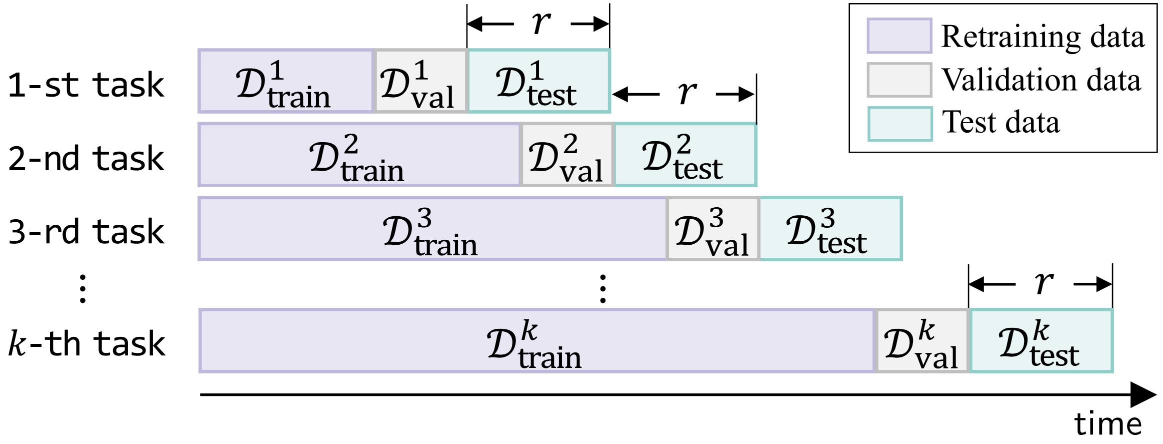
Update Model with Newly Incoming Data

New stock data continually arrive and reveal more underlying patterns.

To avoid the **model aging** issue and pursue higher accuracy, continually learning new emerging patterns is of vital importance!



RR (Rolling Retraining)



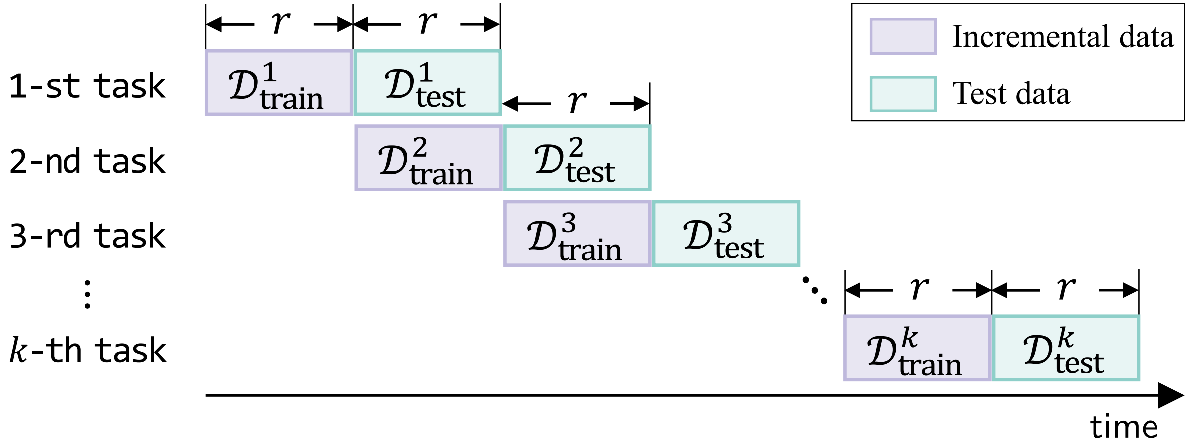
Pros

- Completeness of historical information.

Cons:

- Abundant recent samples (patterns) are filtered out for validation
- High time and space consumption.

IL (Incremental Learning)



Pros

- High training efficiency.
- Low space consumption.

Cons:

- Overfitting risk due to the limited size of incremental data and distribution shifts.

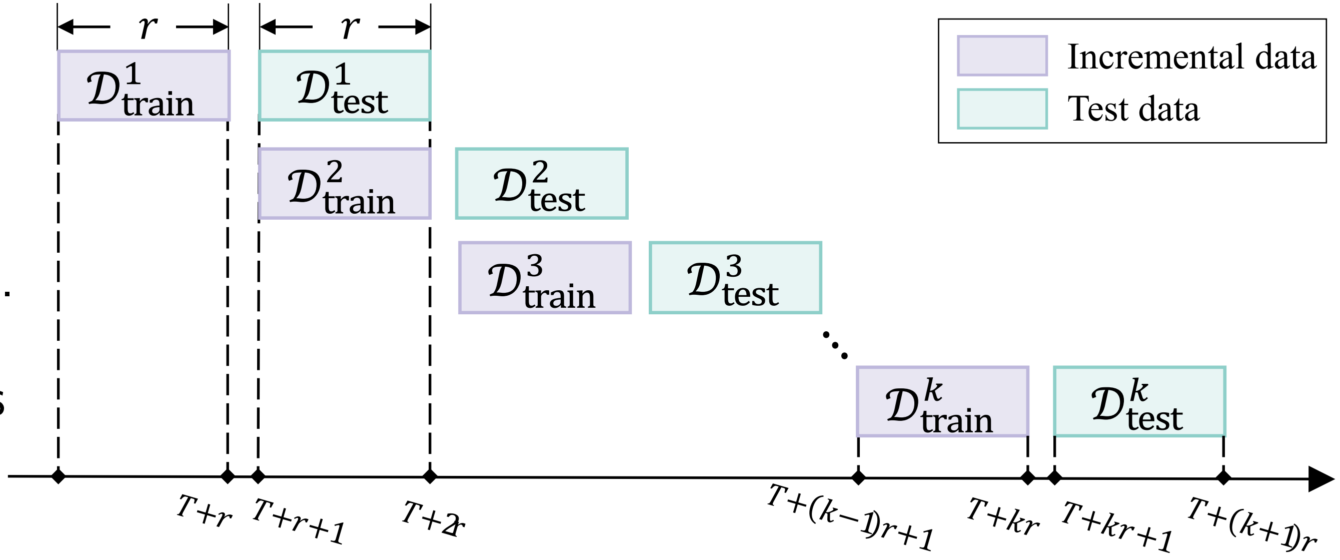
Notation and Definition

$\mathbf{X}^{(t)} \in \mathbb{R}^{S\times D}$: features of S stocks at date t.

$\mathbf{Y}^{(t)} \in \mathbb{R}^{S\times 1}$: labels of S stocks at date t.

θ^0: parameters pretrained on $\left\{\left(\mathbf{X}^{(t)}, \mathbf{Y}^{(t)}\right)\right\}_{t=1}^{T}$.

We launch an IL task every r days, where r is predetermined by practical applications.



One IL Task for Stock Trend Forecasting

For the k-th IL task at date $T+kr+1$, we fine-tune the parameters θ^{k-1} on incremental data $\mathcal{D}^k_{\text{train}} = \left\{\left(\mathbf{X}^{(t)}, \mathbf{Y}^{(t)}\right)\right\}_{t=T+(k-1)r+1}^{T+kr}$ and predict labels on test data $\mathcal{D}^k_{\text{test}} = \left\{\left(\mathbf{X}^{(t)}, \mathbf{Y}^{(t)}\right)\right\}_{t=T+kr+1}^{T+(k+1)r}$.

Output: updated parameters θ^k and predictions $\left\{\widehat{\mathbf{Y}}^{(t)}\right\}_{t=T+kr+1}^{T+(k+1)r}$.

IL Task evaluation: compute a loss function $\mathcal{L}_{\text{test}}$ on $\mathcal{D}^k_{\text{test}}$, *e.g.*, MSE.

Problem Statement

IL for Stock Trend Forecasting

Given a predefined r, IL for stock trend forecasting considers a sequence of IL tasks, i.e., $\mathcal{T} = \{(\mathcal{D}^1_{\text{train}}, \mathcal{D}^1_{\text{test}}), \cdots, (\mathcal{D}^k_{\text{train}}, \mathcal{D}^k_{\text{test}}), \cdots\}$.

In each task, we update the model parameters, perform online inference, and evaluate the performance based on the ground-truth labels of the test data.

Goal: achieve the best overall performance across all IL tasks, which can be evaluated by excess annualized returns or other ranking metrics of stock trend forecasting.

Typically, IL holds a strong assumption that a model which fits recent data can perform well on the following data under the same distribution.

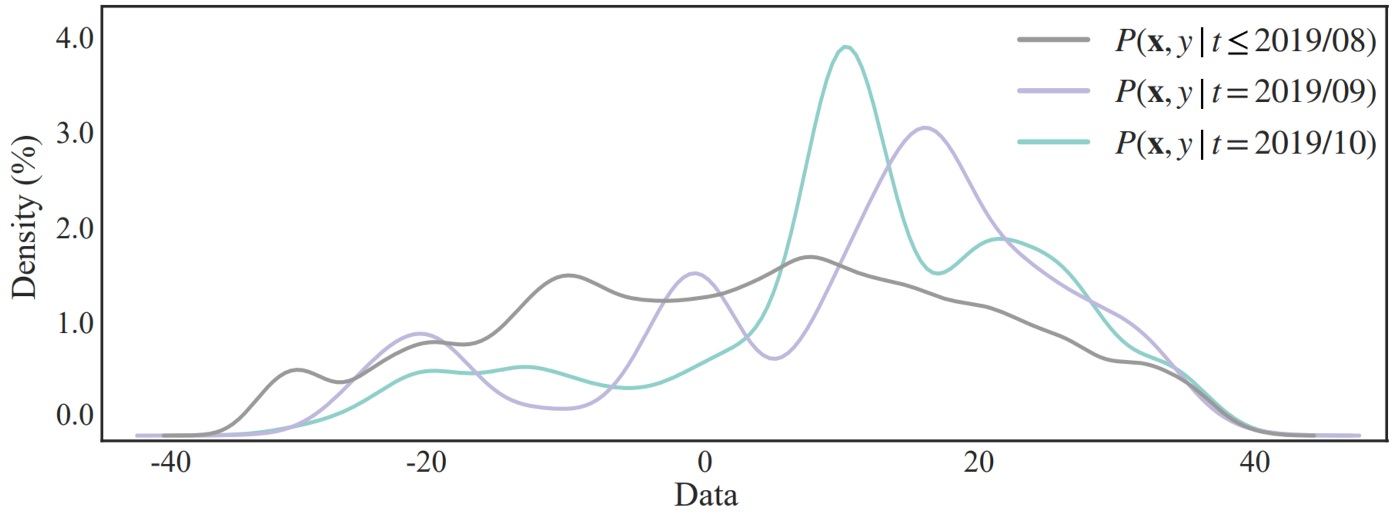
However, the stock market is dynamically evolving!

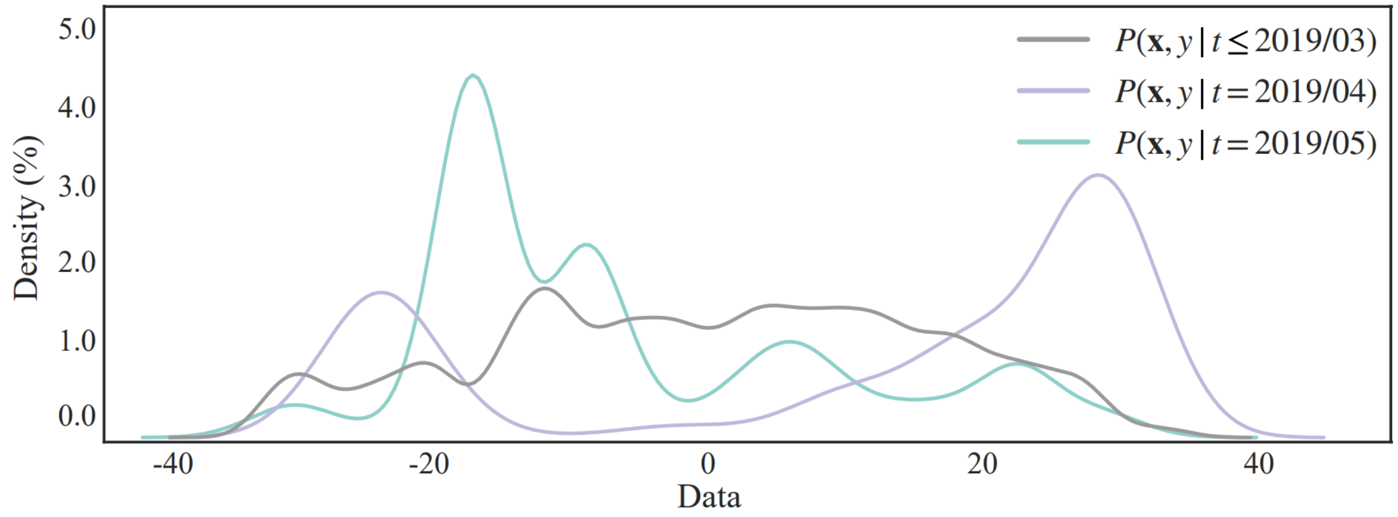
The Challenge of Distribution Shifts

$\mathcal{D}^k_{\text{train}}$ and $\mathcal{D}^k_{\text{test}}$ are likely to have two different joint distributions,

i.e., $\mathcal{P}^k_{\text{train}}(\mathbf{x}, y) \neq \mathcal{P}^k_{\text{test}}(\mathbf{x}, y)$.

Remark: typical IL cannot consistently benefit from incremental data and may even suffer from inappropriate updates.





How to Perform Incremental Learning Tasks Effectively?

Key Idea: Two-fold Adaptation

Updates stem from both the **incremental data** $\mathcal{D}_{\text{train}}^{k}$ and the **initial parameters** θ^{k-1}.

two-fold adaptation

Data Adaptation aims to close the gap between distributions of incremental data and test data.

- make distributions more stationary
- *e.g.*, resolve biased patterns

Model Adaptation focuses on learning a good initialization of parameters for each IL task.

- appropriately adapt to incremental data
- still retain a degree of robustness to distribution shifts

Data Adaptation

Some RR methods (e.g., DDG-DA) adopt data adaptation by resampling all the historical data (e.g., 500M samples).

Such a **coarse-grained** way is inapplicable to IL where the incremental data is of limited size (e.g., only 1K samples) and contains deficient samples to reveal future patterns.

We propose to adapt all features and labels in a **fine-grained** way.

Some shift patterns that repeatedly appear in the historical data are learnable.

- E.g., the stock trend shifts caused by overreacting to some bullish news happen from time to time.

We adapt both $\mathcal{D}^k_{\text{train}}$ and $\mathcal{D}^k_{\text{test}}$!

Data Adaptation

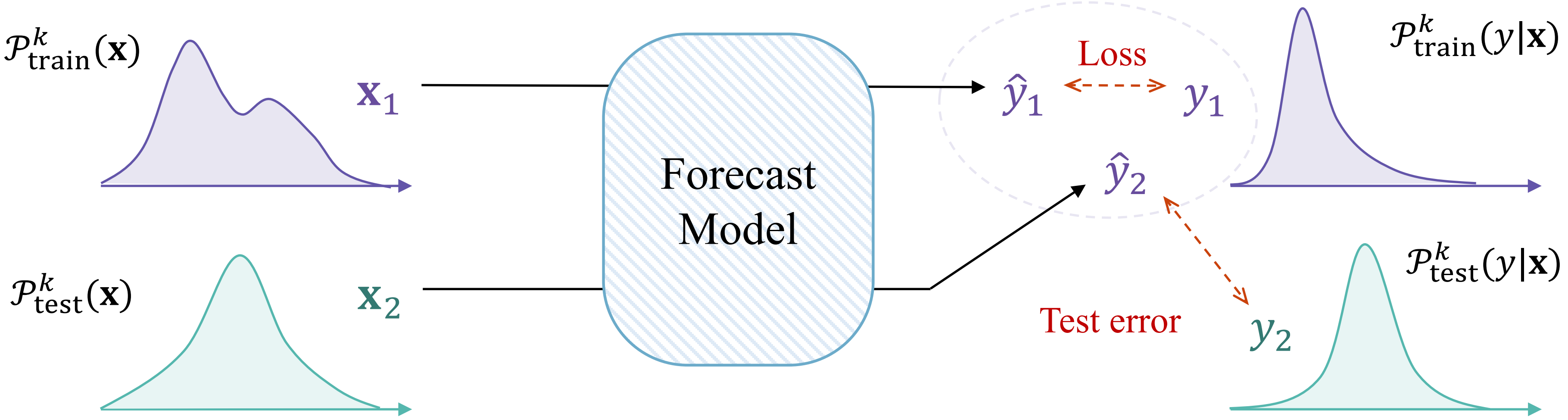
Distribution shifts $\mathcal{P}^k_{\text{train}}(\mathbf{x}, y) \neq \mathcal{P}^k_{\text{test}}(\mathbf{x}, y)$ can be zoomed into:

Covariate Shift

$$\mathcal{P}^k_{\text{train}}(\mathbf{x}) \neq \mathcal{P}^k_{\text{test}}(\mathbf{x})$$

Conditional distribution shift

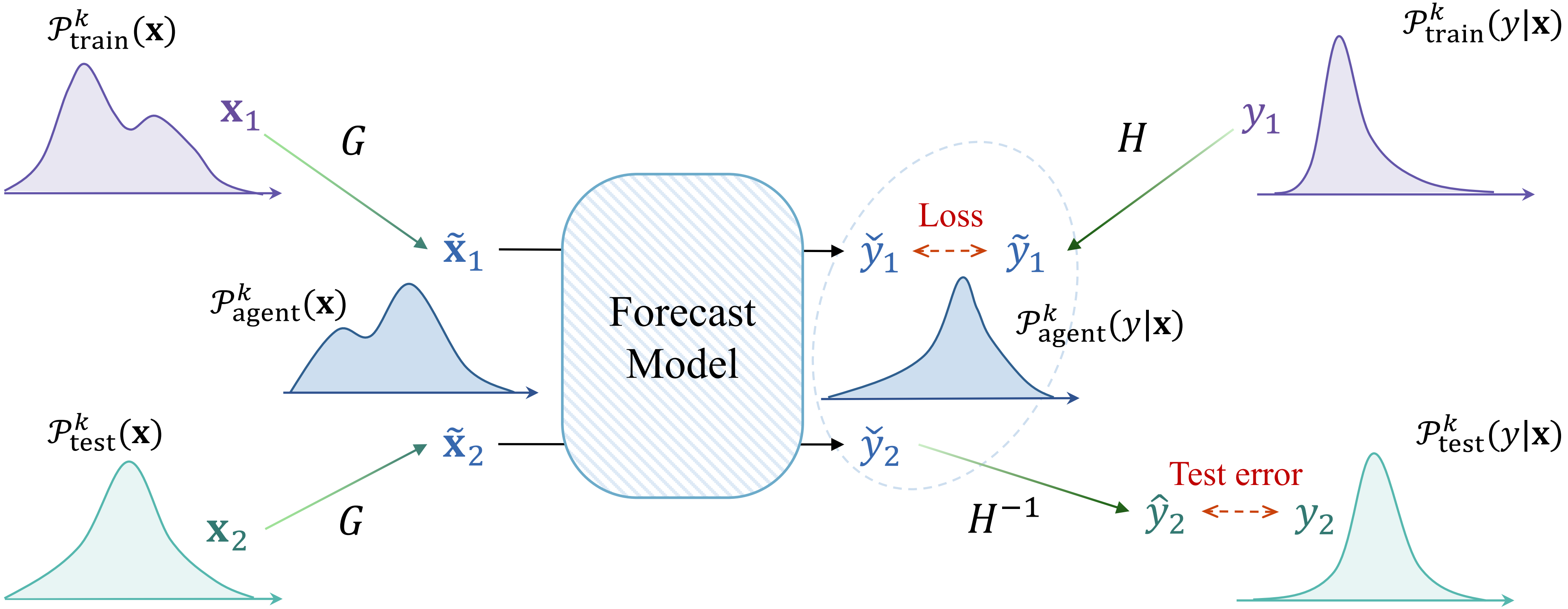
$$\mathcal{P}^k_{\text{train}}(y|\mathbf{x}) \neq \mathcal{P}^k_{\text{test}}(y|\mathbf{x})$$



Data Adaptation

data adapter needs to provide two mapping functions, G and H.

- G transforms the features of $\mathcal{D}^k_{\text{train}}$ and $\mathcal{D}^k_{\text{test}}$.
- H adapts the labels of $\mathcal{D}^k_{\text{train}}$, and H^{-1} restores the model outputs.



Model Adaptation

Conventional IL blindly inherits the parameters learned in the previous task (e.g., θ^{k-1}) as initial weights and updates it into θ^{k}.

The parameters may fall into a local optimum.

We use model adapter to guide parameter initialization.

What is a good initialization of parameters?

Robustness: preserve historical experience and retain generalization ability against distribution shifts.

Adaptiveness: quickly learn task-specific information without being trapped in past experiences.

How to Establish the Two-fold Adaptation?

Manual Design is Intractable!

Numerous factors should be considered, *e.g.*,

- forecast model
- dataset
- period
- degree of distribution shifts
- etc.

Tough to reach a sweet spot between robustness and adaptiveness!

- learn more from incremental data $\Longrightarrow$ overfitting
- learn less from incremental data $\Longrightarrow$ underfitting

A Bi-level Optimization Perspective

2023/08 2023/09 2023/10

$\mathcal{D}^k_{\text{train}}$ $\mathcal{D}^k_{\text{test}}$

Lower-level optimization objective:

minimize training loss on the incremental data by an optimal forecast model

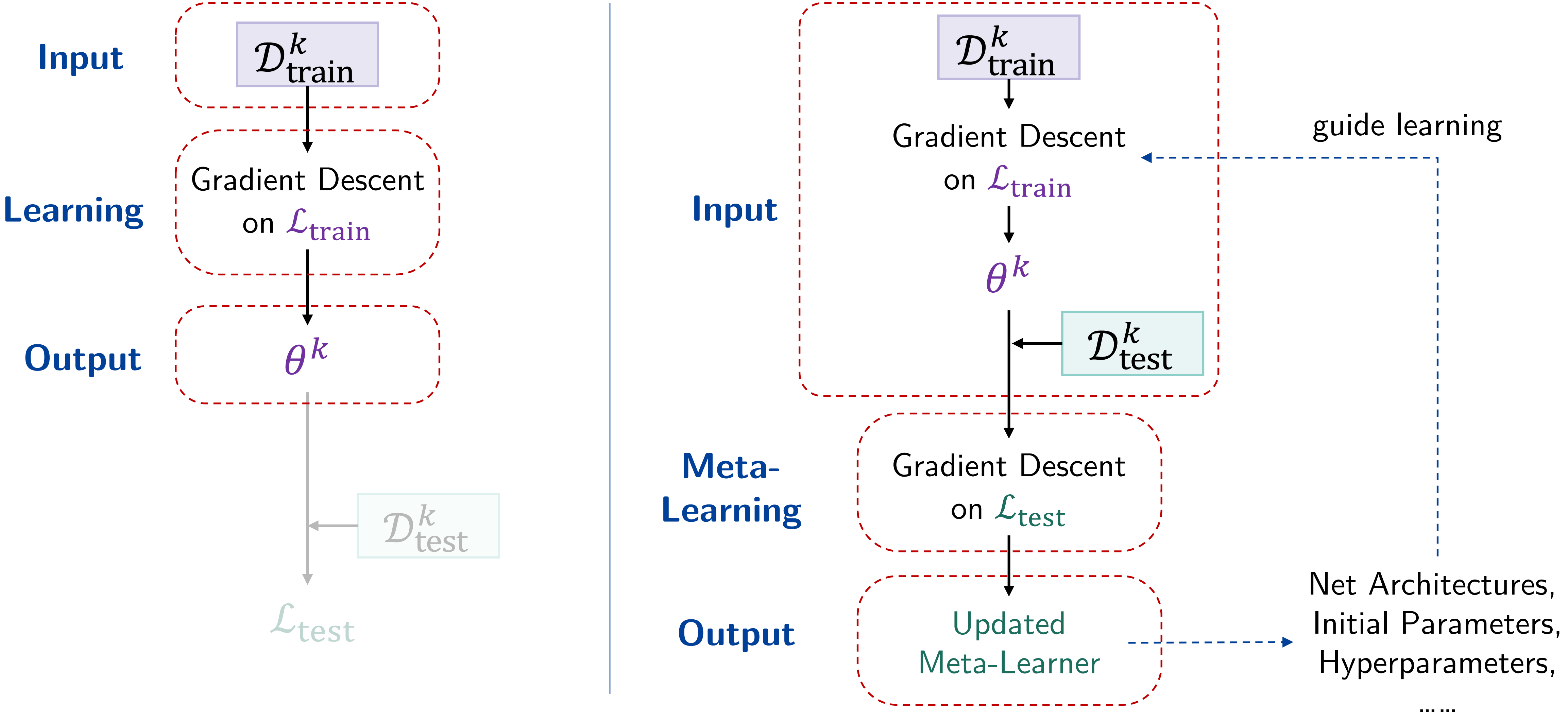
$$\min \sum_{(\mathbf{x},y)\in\mathcal{D}^k_{\text{train}}} \left(F\left(\mathbf{x};\theta^{k-1}\right)-y\right)^2$$

Upper-level optimization objective:

minimize test error on the future test data by optimal adaptations for incremental learning

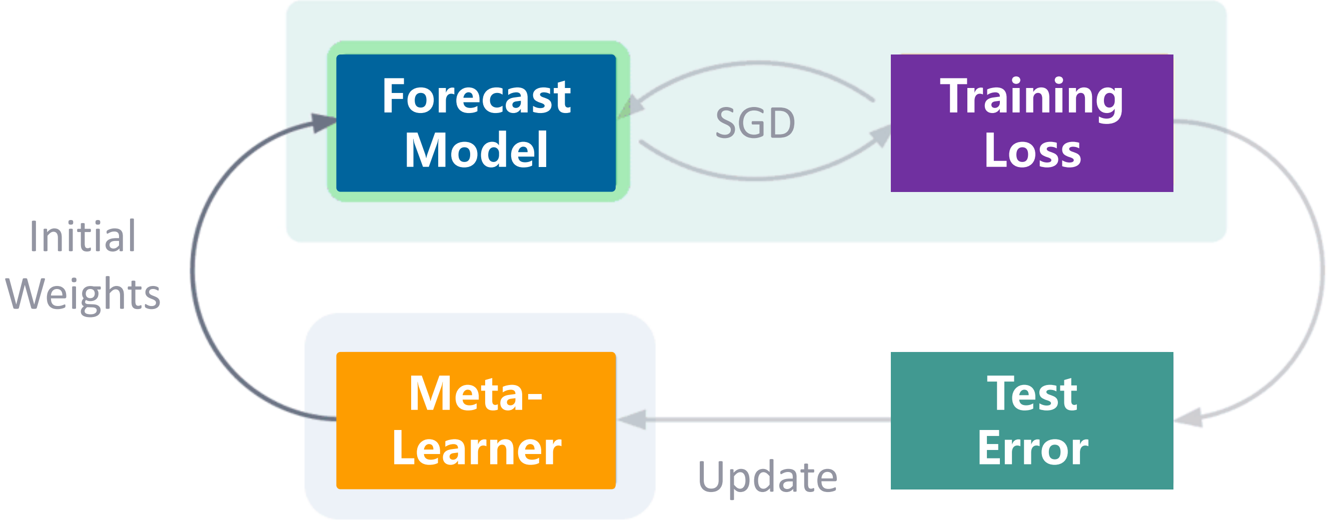
$$\min \sum_{(\mathbf{x},y)\in\mathcal{D}^k_{\text{test}}} \left(F\left(\mathbf{x};\theta^{k}\right)-y\right)^2$$

Meta-learning (Learning to Learn)



MAML: Learning a Good Initialization of Parameters [ICML'17]

$\theta^{k-1} = \phi^{k-1}$



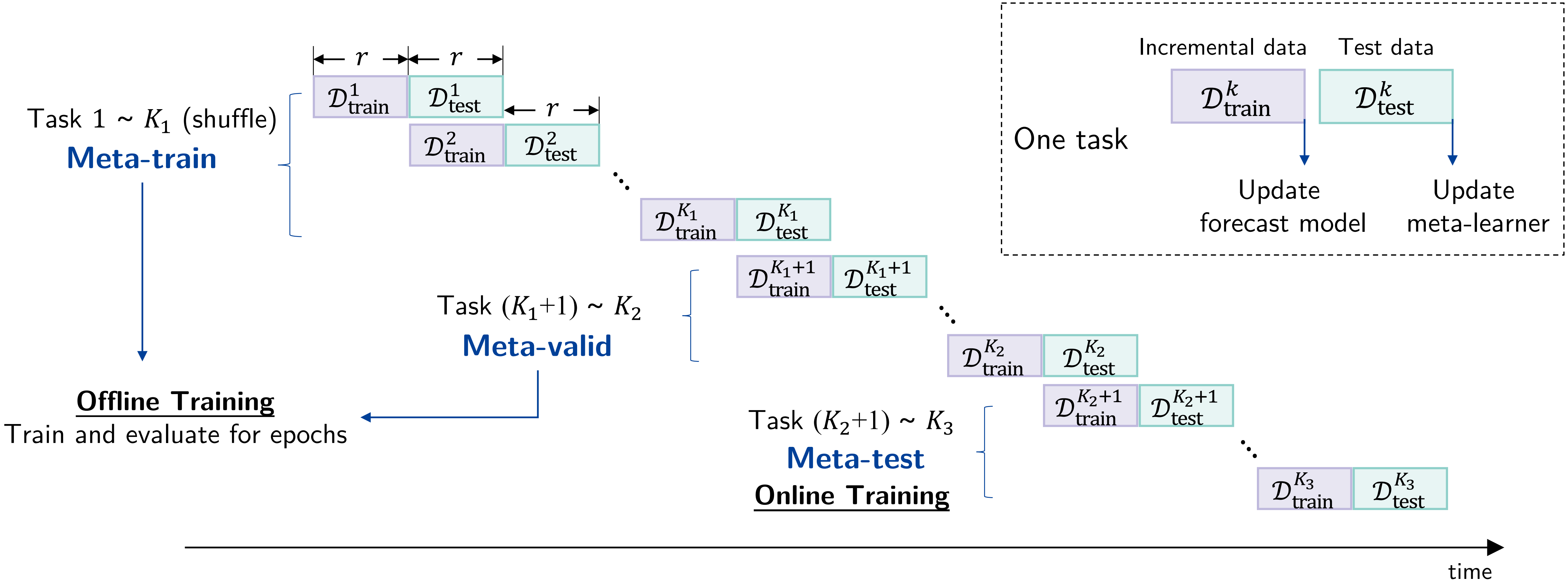
$\theta^{k} = \phi^{k-1} - \eta \nabla_{\phi} \mathcal{L}_{\text{train}}(\phi^{k-1})$

(One-step SGD)

$$\nabla_{\phi} \mathcal{L}_{\text{test}}(\theta^k) = \frac{\partial \mathcal{L}_{\text{test}}(\theta^k)}{\partial \theta^k} \cdot \frac{\partial \theta^k}{\partial \phi^{k-1}}$$
$$= \frac{\partial \mathcal{L}_{\text{test}}(\theta^k)}{\partial \theta^k} \cdot \frac{\partial (\phi^{k-1} - \eta \nabla_{\phi} \mathcal{L}_{\text{train}}(\phi^{k-1}))}{\partial \phi^{k-1}}$$
$$= \frac{\partial \mathcal{L}_{\text{test}}(\theta^k)}{\partial \theta^k} \left(1 - \eta \frac{\partial}{\partial \phi^{k-1}} \nabla_{\phi} \mathcal{L}_{\text{train}}(\phi^{k-1})\right)$$

gradient by gradient

Applying Meta-learning to Incremental Learning



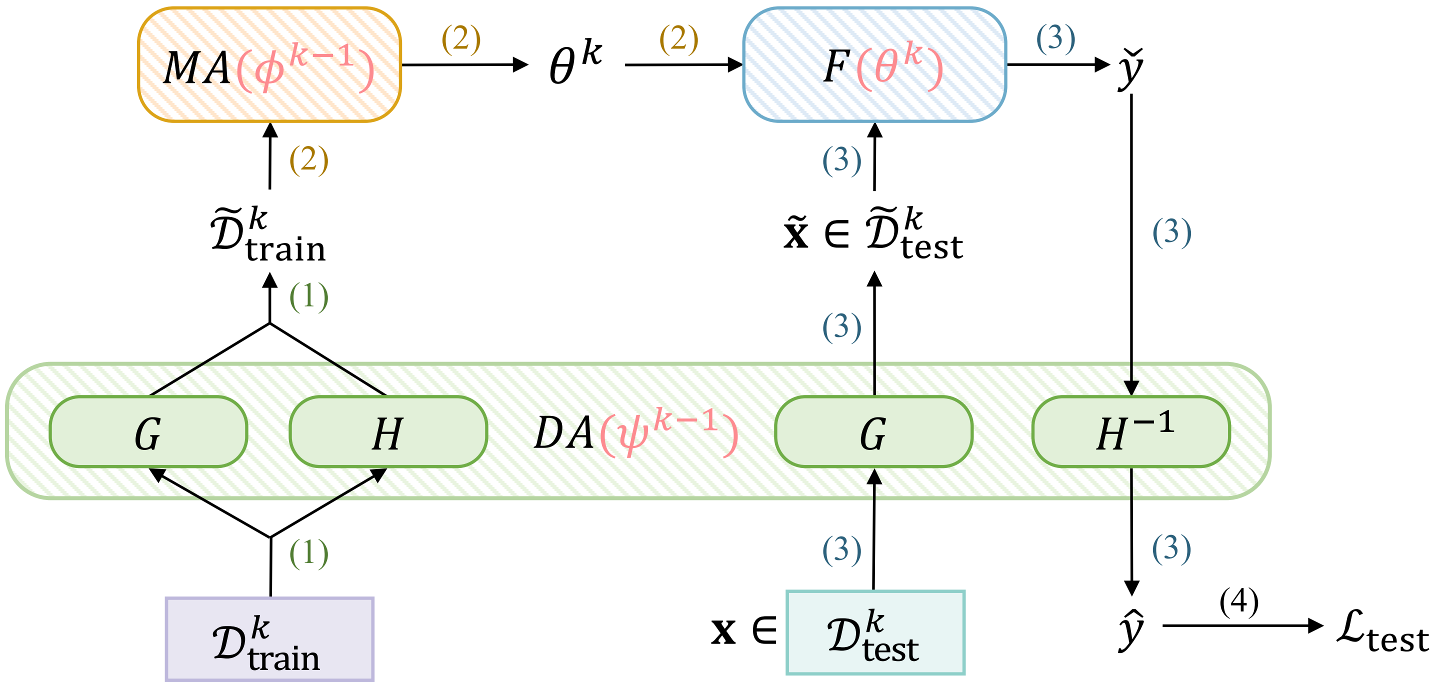
Incremental learning Tasks ⇒ a sequence of bi-level optimization problems ⇒ a sequence of meta-learning tasks

Put It All Together: The DoubleAdapt Approach

The DoubleAdapt Approach

Key Components:

- forecast model $F(\cdot\,;\theta)$
- model adapter $MA(\cdot\,;\phi)$
- data adapter† $DA(\cdot\,;\psi)$



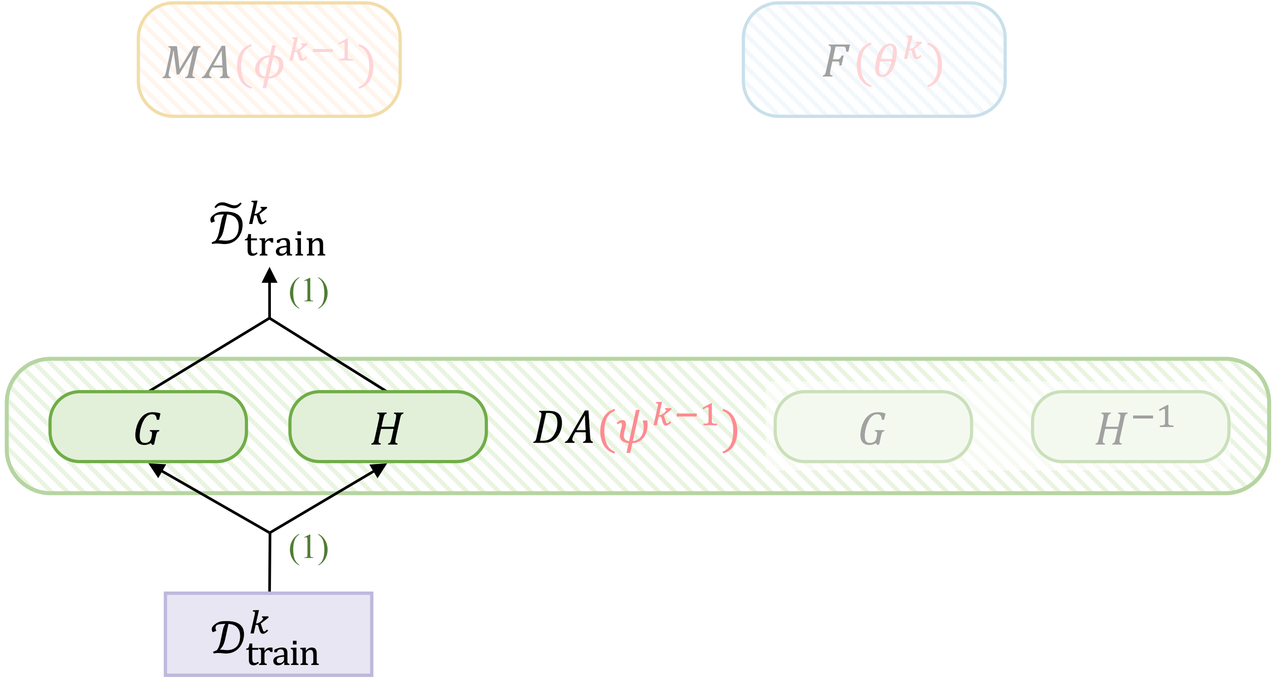
†Please refer to our paper for detailed implementation of DA.

Step (1): Incremental Data Adaptation

Given incremental data $\mathcal{D}^k_{\text{train}}$,

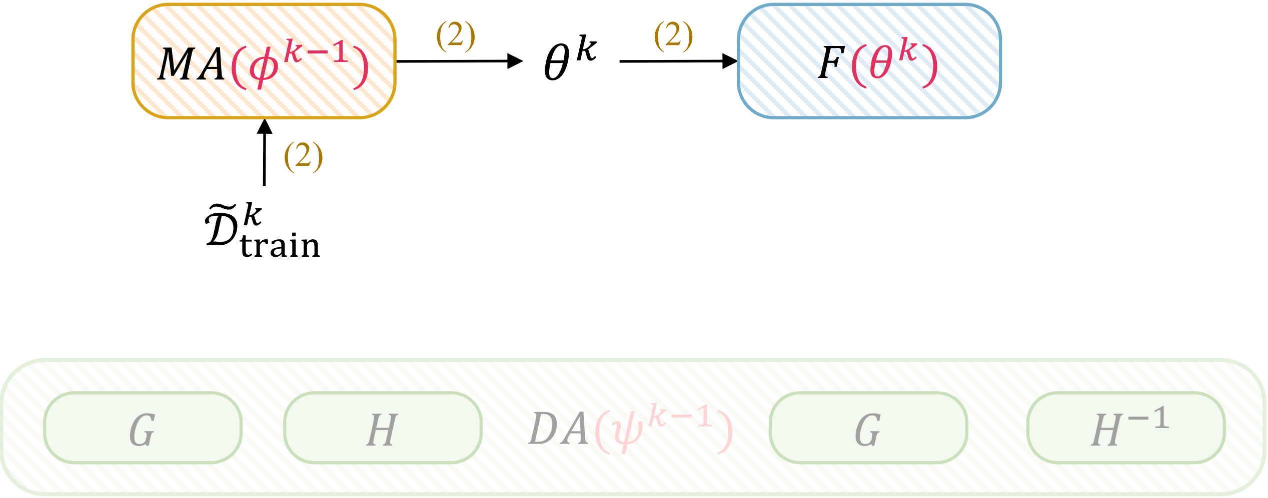
- G transforms each $\mathbf{x}$ into $\tilde{\mathbf{x}}$;
- H transforms each y into $\tilde{y}$.

Output: an adapted incremental dataset $\widetilde{\mathcal{D}}^k_{\text{train}}$



Step (2): Model Adaptation (Lower-level Optimization)

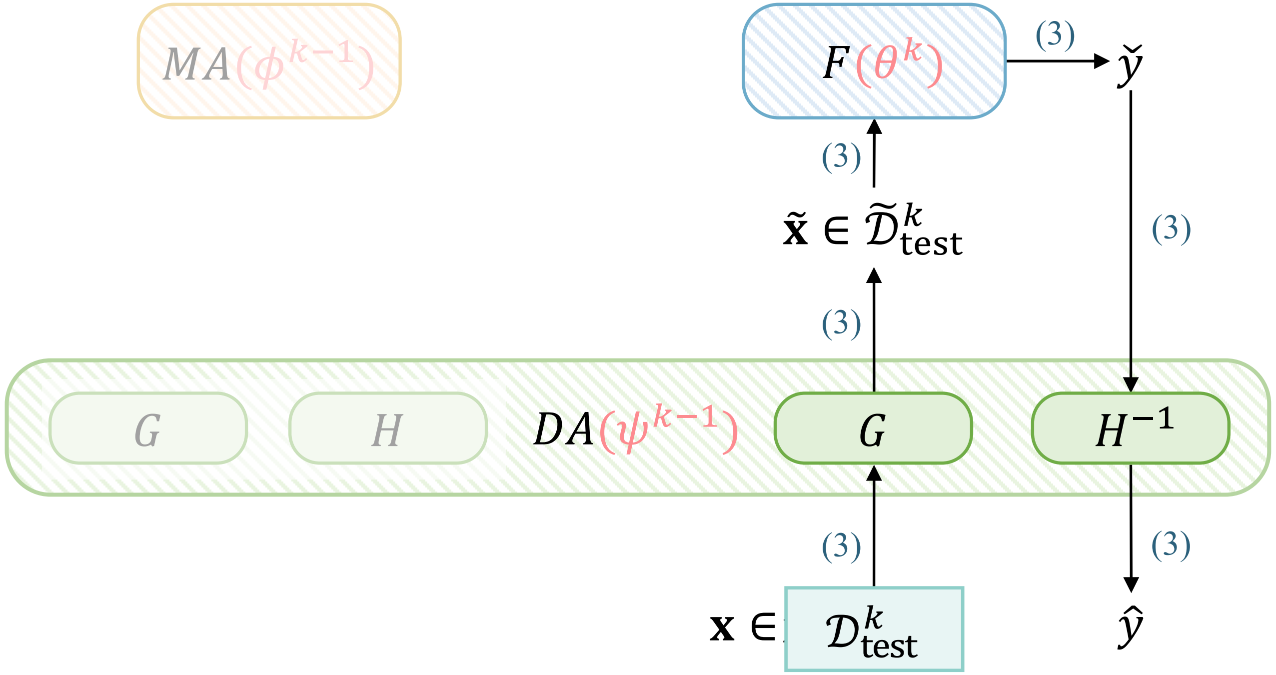
- MA initializes F by ϕ^{k-1}.
- F is finetuned on $\widetilde{\mathcal{D}}^k_{\text{train}}$, and its parameters become θ^k.
- $F(\cdot;\theta^k)$ is deployed online.



Step (3): Online Inference

Given each $\mathbf{x}$ of $\mathcal{D}^k_{\text{test}}$,

- G transforms $\mathbf{x}$ into $\tilde{\mathbf{x}}$;
- $F(\cdot;\theta^k)$ makes intermediate prediction $\check{y}$;
- H^{-1} transforms $\check{y}$ into final prediction $\hat{y}$;
- $\mathcal{L}_{\text{test}}$ is computed based on $\hat{y}$ and ground-truth y.



Step (4): Upper-level Optimization of Meta-learners

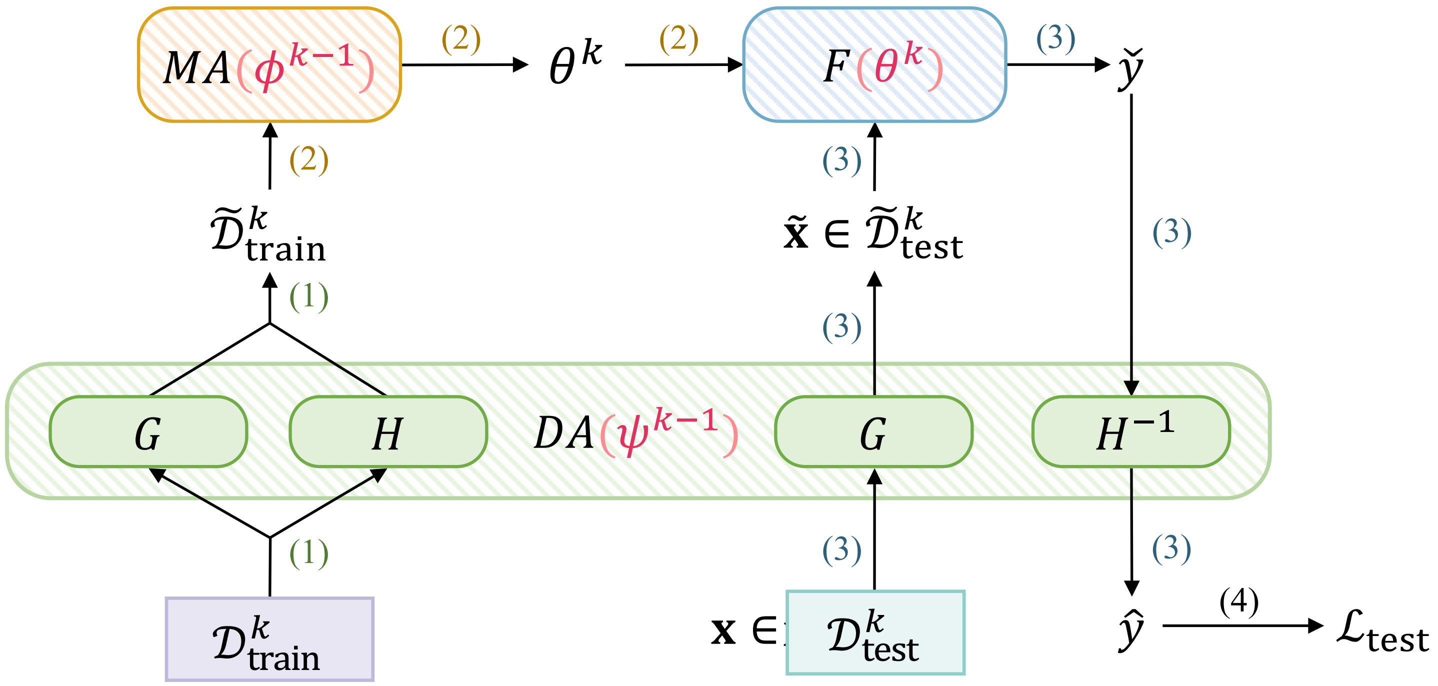
$$\phi^k, \psi^k = \arg\min_{\phi,\psi} \mathcal{L}_{\text{test}}(\widetilde{\mathcal{D}}^k_{\text{test}}; \theta^k),$$

$$\text{s.t.} \quad \theta^k = MA(\widetilde{\mathcal{D}}^k_{\text{train}}; \phi^{k-1}),$$

where

$$\widetilde{\mathcal{D}}^k_{\text{train}} = DA(\mathcal{D}^k_{\text{train}}; \psi^{k-1});$$

$$\widetilde{\mathcal{D}}^k_{\text{test}} = DA(\mathcal{D}^k_{\text{test}}; \psi^{k-1}).$$



Data Adapter

- Transform features of two data onto a new common hyperplane.
- Different types of feature vectors may require different transformations.
- Stock trends tend to bear similar shift patterns when the stocks belong to the same **concept**.
- $\mathbf{p}_i$: learnable embeddings of concept i.
- **concept-oriented adaptation**

  with multiple transformation heads

$$G(\mathbf{x}) \coloneqq \mathbf{x} + \sum_{i=1}^{N} s_i\, g_i(\mathbf{x})$$

$$g_i(\mathbf{x}) = \mathbf{W}_i \mathbf{x} + \mathbf{b}_i$$

$$s_i = \mathrm{softmax}\big(\mathrm{cosine}(\mathbf{x}, \mathbf{p}_i)\big)$$

$$H(y) \coloneqq \sum_{i=1}^{N} s_i\, h_i(y) \qquad H^{-1}(\hat{y}) \coloneqq \sum_{i=1}^{N} s_i\, h_i^{-1}(\hat{y})$$

$$h_i(y) = \gamma_i y + \beta_i \qquad h_i^{-1}(\hat{y}) = (\hat{y} - \beta_i)/\gamma_i$$

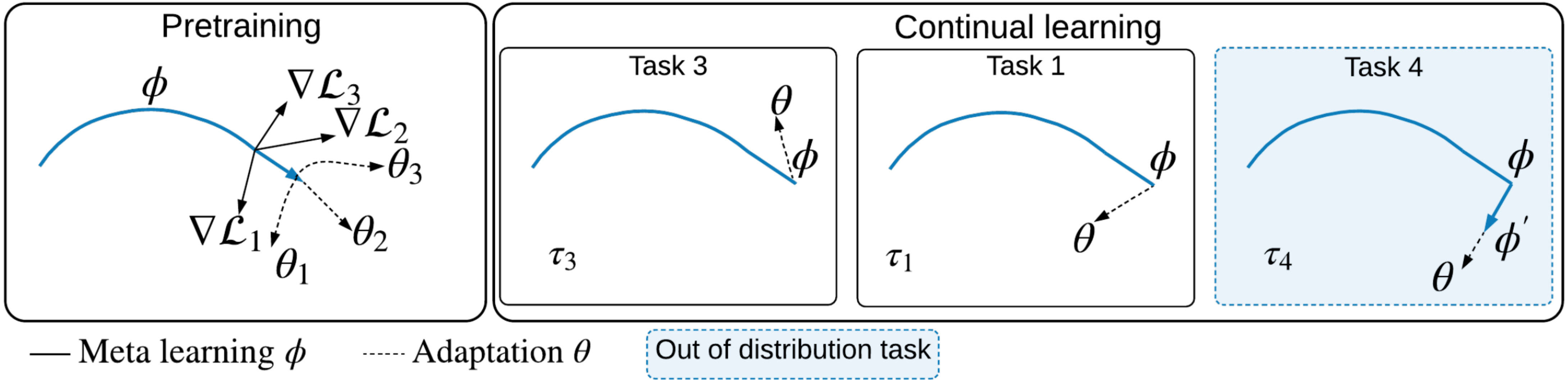
What are Comparison Methods?

C-MAML（NeurIPS'20）

Continual-MAML (C-MAML [1]) follows MAML to pretrain its meta-learner (with slow weights ϕ) that can produce a model (with fast weights θ) to accommodate new tasks.

At the online time, the meta-learner is updated only after detecting a OOD task.

CMAML considers distribution shifts between different tasks but **ignores** the shifts within a single task.

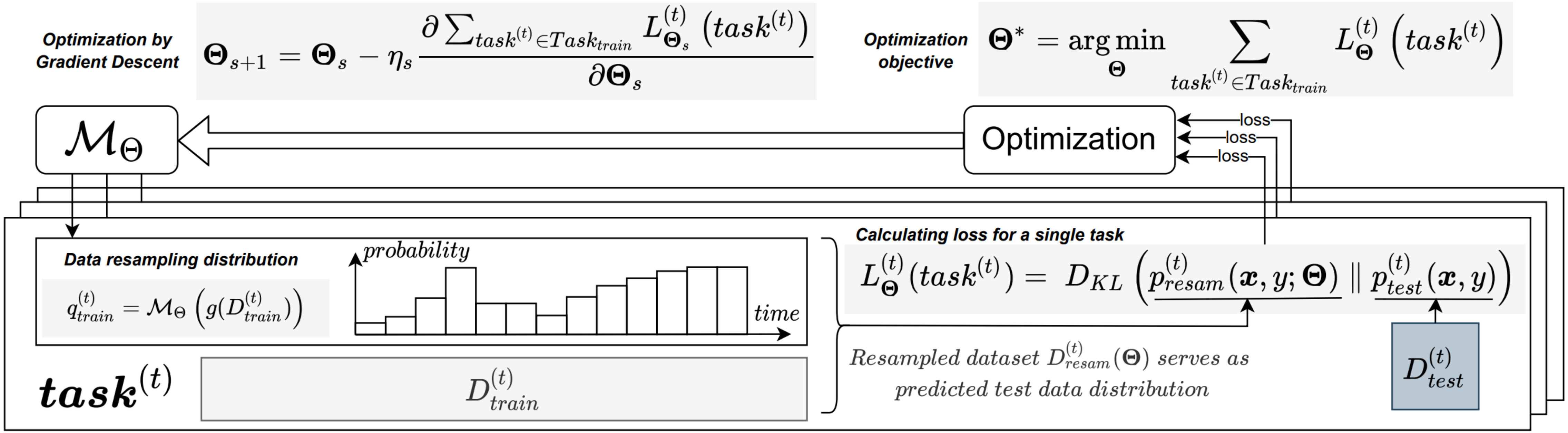


[1] Massimo Caccia et al., Online Fast Adaptation and Knowledge Accumulation (OSAKA): a New Approach to Continual Learning. NeurIPS'20.

DDG-DA（AAAI'22)

DDG-DA [2], an advanced RR method, copes with distribution shifts by predicting the future data distribution and resampling the training data for a similar distribution.

Specifically, DDG-DA adapts the training data by assigning samples grouped by periods with different weights.



[2] Wendi Li et al. DDG-DA: Data Distribution Generation for Predictable Concept Drift Adaptation. AAAI'22

Comparison with DoubleAdapt

C-MAML and DoubleAdapt use the first-order approximation version of MAML.

Method	Model Adaptation	Data Adaptation	Time complexity of the offline training	Time complexity of the k-th online task
DDG-DA	Full Retraining	Coarse-grained Resampling	$\mathcal{O}(ET^2S)$	$\mathcal{O}(E(T+kr)S)$
C-MAML	MAML+IL	×	$\mathcal{O}(TS)$	$\mathcal{O}(rS)$
DoubleAdapt	MAML+IL	Fine-grained Transformation	$\mathcal{O}(TS)$	$\mathcal{O}(rS)$

r: task interval; S: # of stocks; T: # of dates in the meta-train set; E: # of retraining epochs till convergence.

How does DoubleAdapt Perform on Real Data?

Experiments

Datasets : CSI 300, CSI500

- Alpha360: 60-day time series; 6 indicators on each day
  - opening price
  - closing price
  - highest price
  - lowest price
  - volume weighted average price
  - trading volume
- Date range:
  - training: 2018/01/01 — 2014/12/31
  - validation: 2015/01/01 — 2016/12/31
  - test: 2017/01/01 — 2020/07/31

Evaluation Metrics

- Ranking metrics:
  - IC
  - ICIR
  - Rank IC
  - Rank ICIR
- Portfolio metrics:
  - excess annualized return (Return)
  - Information ratio (IR)

DoubleAdapt outperforms IL by **6%** on IC and **14%** on excess annualized returns.
DoubleAdapt outperforms RR by **6%** on IC and **47%** on excess annualized returns.

Model	Method	CSI300						CSI500					
		IC	ICIR	Rank IC	Rank ICIR	Return	IR	IC	ICIR	Rank IC	Rank ICIR	Return	IR
Transformer	RR	0.0449	0.3410	0.0462	**0.3670**	0.0881	1.0428	0.0452	0.4276	0.0469	0.4732	0.0639	0.9879
	DDG-DA	0.0420	0.3121	0.0441	0.3420	0.0823	1.0018	0.0450	0.4223	0.0465	0.4634	0.0681	1.0353
	IL	0.0431	0.3108	0.0411	0.2944	0.0854	0.9215	0.0428	0.3943	0.0453	0.4475	0.1014	1.5108
	C-MAML	0.0479	0.3560	0.0448	0.3405	0.0986	1.0537	0.0477	0.4620	0.0468	0.4861	0.0930	1.4923
	DoubleAdapt	**0.0516**	**0.3889**	**0.0475**	0.3585	**0.1041**	**1.1035**	**0.0492**	**0.4653**	**0.0490**	**0.4970**	**0.1330**	**1.9761**
LSTM	RR	0.0592	0.4809	0.0536	0.4526	0.0805	0.9578	0.0642	0.6187	0.0543	0.5742	0.0980	1.5220
	DDG-DA	0.0572	0.4622	0.0528	0.4415	0.0887	1.0583	0.0636	0.6181	0.0540	0.5783	0.1061	1.6673
	IL	0.0594	0.4664	0.0546	0.4362	0.1089	1.2553	0.0576	0.5550	0.0553	0.5660	0.1249	1.8461
	C-MAML	0.0568	0.4601	0.0517	0.4381	0.0963	1.1145	0.0582	0.5863	0.0550	0.5898	0.1315	1.9770
	DoubleAdapt	**0.0632**	**0.5126**	**0.0567**	**0.4669**	**0.1117**	**1.3029**	**0.0648**	**0.6331**	**0.0594**	**0.6087**	**0.1496**	**2.2220**
ALSTM	RR	0.0630	0.5084	0.0589	**0.4892**	0.0947	1.1785	0.0649	0.6331	0.0575	0.6030	0.1211	1.8726
	DDG-DA	0.0609	0.4915	0.0581	0.4823	0.0966	1.2227	0.0645	0.6298	0.0573	0.6029	0.1042	1.6091
	IL	0.0626	0.4762	0.0585	0.4489	0.1171	1.3349	0.0596	0.5705	0.0579	0.5712	0.1501	2.1468
	C-MAML	0.0636	0.5064	0.0588	0.4765	0.1085	1.2432	0.0647	**0.6490**	0.0598	**0.6330**	0.1644	2.4636
	DoubleAdapt	**0.0679**	**0.5480**	**0.0594**	0.4882	**0.1225**	**1.4717**	**0.0653**	0.6404	**0.0607**	0.6170	**0.1738**	**2.5192**
GRU	RR	0.0629	0.5105	0.0581	0.4856	0.0933	1.1428	0.0669	0.6588	0.0586	0.6232	0.1200	1.8629
	DDG-DA	0.0623	0.5045	0.0589	0.4898	0.0967	1.1606	0.0666	0.6575	0.0582	0.6234	0.1264	1.9963
	IL	0.0633	0.4818	0.0596	0.4609	0.1166	1.3196	0.0637	0.6093	0.0617	0.6291	0.1626	2.3352
	C-MAML	0.0638	0.5085	0.0595	0.4865	0.1121	1.3210	0.0646	0.6498	0.0600	**0.6494**	0.1693	**2.5064**
	DoubleAdapt	**0.0687**	**0.5497**	**0.0621**	**0.5110**	**0.1296**	**1.5123**	**0.0686**	**0.6652**	**0.0632**	0.6445	**0.1748**	2.4578

Ablation Study

IL+MA+DA (*i.e.*, DoubleAdapt) is the best against different kinds of distribution shifts.

IL+MA+DA > IL+MA $\Rightarrow$ Data adaptation effectively facilitates model adaptation.

IL+MA+DA > IL+MA $\Rightarrow$ MA is necessary, especially under abrupt shifts.

IL+DA > IL+DA $\Rightarrow$ Data adaptation alone also beats one-sided model adaptation.

Method	Overall Performance				Gradual Shifts				Abrupt Shifts			
	IC	ICIR	RankIC	RankICIR	IC	ICIR	RankIC	RankICIR	IC	ICIR	RankIC	RankICIR
IL	0.0633	0.4818	0.0596	0.4609	0.0643	0.4936	0.0652	0.5161	0.0690	0.5134	0.0619	0.4581
+*DA*	0.0659	0.5279	0.0615	0.4993	0.0708	0.5938	0.0692	0.5897	0.0690	0.5271	0.0620	0.4677
+*MA*	0.0658	0.5160	0.0610	0.4910	0.0703	0.5703	0.0680	0.5686	0.0681	0.5085	0.0618	0.4594
+*MA*+*G*	0.0678	0.5360	0.0619	0.4978	0.0740	0.6155	0.0709	0.6060	0.0694	0.5224	0.0626	0.4672
+MA+H+H^{-1}	0.0660	0.5207	0.0614	0.4995	0.0714	0.5846	0.0701	0.5958	0.0680	0.5093	0.0616	0.4615
+*MA*+*DA*	**0.0687**	**0.5497**	**0.0621**	**0.5110**	**0.0755**	**0.6390**	**0.0713**	**0.6243**	**0.0699**	**0.5323**	**0.0620**	**0.4730**

CSI300+GRU

Time Cost Study

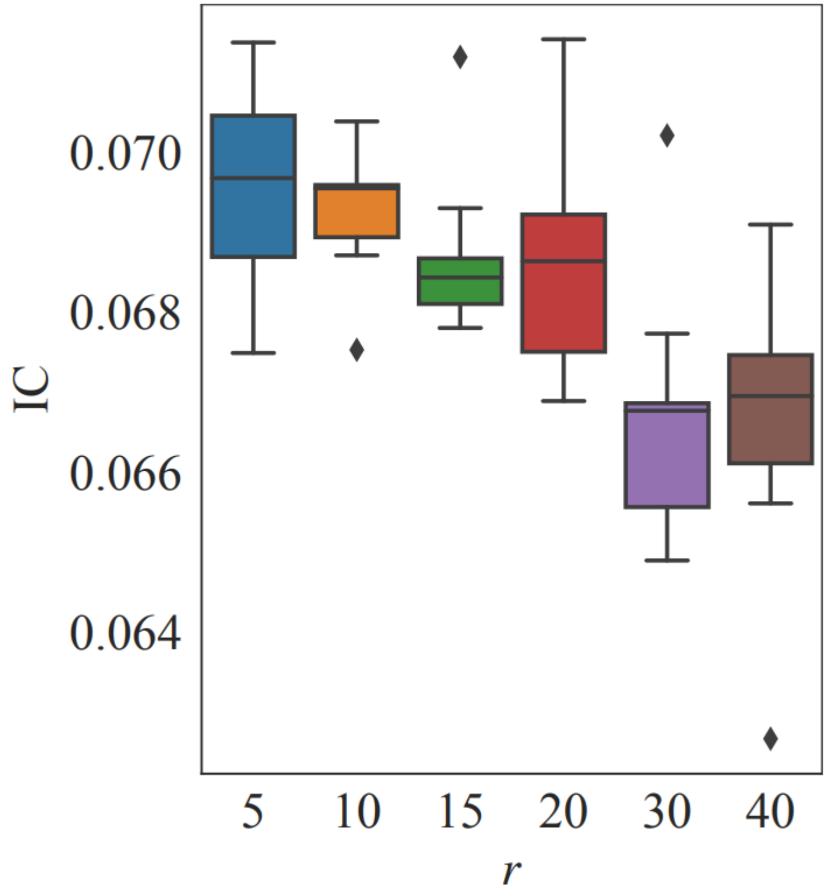
- DoubleAdapt is much more efficient than RR methods
- The overhead of DoubleAdapt is insignificant compared with other IL methods.

Table 3: Empirical time cost (in second) comparison.

Model	Method	CSI 300		CSI 500	
		Offline	Online	Offline	Online
GRU	RR	-	6064	-	10793
	DDG-DA	1862	6719	2360	10713
	IL	**256**	**58**	**394**	**75**
	C-MAML	314	62	533	77
	DoubleAdapt	356	61	677	79

Hyperparameter Study

- A small interval r (e.g., 5 trading days) is desirable.
- In contrast, RR methods with a smaller r will suffer from much more expensive time consumption.



Conclusion

- DoubleAdapt is practical and efficient for stock trend forecasting.
- DoubleAdapt proposes data adaptation and model adaptation that tackle the challenge of distribution shifts.

Future Directions (1/3)

- **Issue:**

  catastrophic forgetting

- **Direction:**

  combine DoubleAdapt with RR

  (*e.g.*, incrementally update the adapters every week and fully retrain them on an enlarged meta-train set after one-quarter incremental learning)

Future Directions (2/3)

- **Issue:**

  using a fixed task interval r to decide when to incrementally update model

- **Direction:**

  dynamically decide the task interval

  (e.g., in a relatively stable environment, update model with a large r)

Future Directions (3/3)

- **Issue:**

  incremental data contains deficient samples

- **Direction:**

  Integrate our fine-grained data adaptation with coarse-grained data resampling

  (e.g., enlarge the incremental data with a few previous samples selected by DDG-DA; then perform DoubleAdapt)

SHANGHAI JIAO TONG UNIVERSITY

THANK YOU!