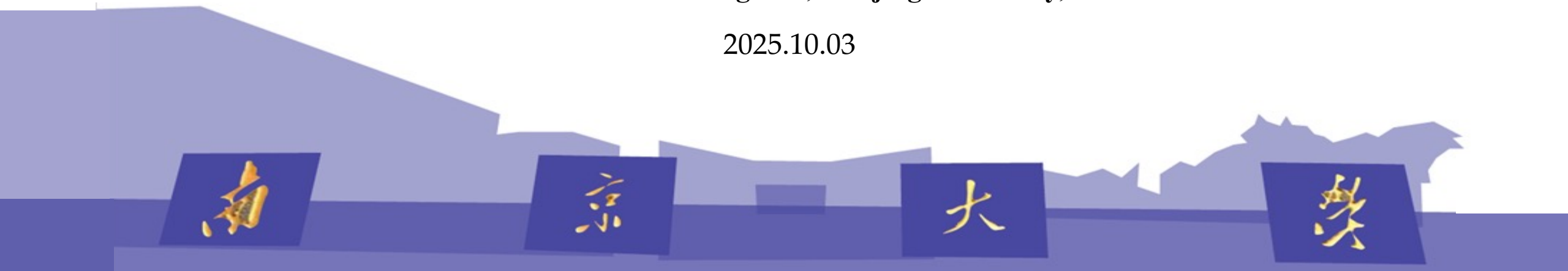
Efficiently Adapting to Online Distribution Shift: Theory and Practice

Presented by **Yu-Yang Qian**

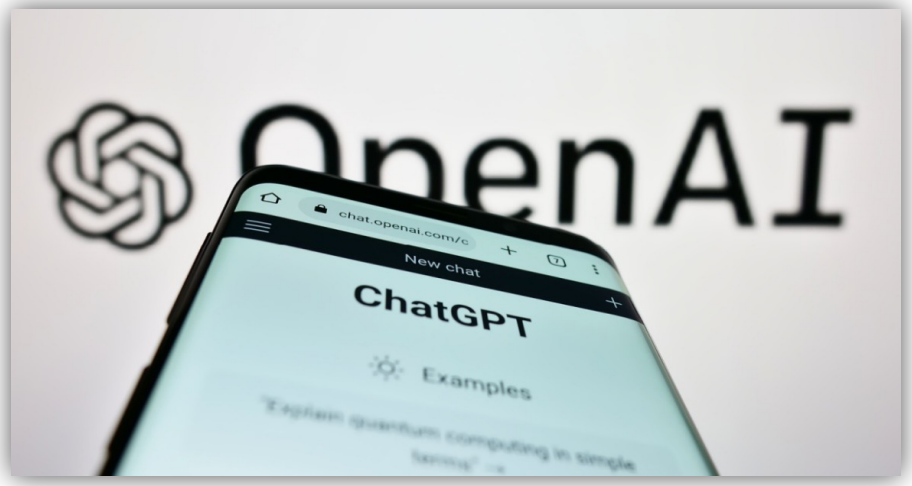
School of Artificial Intelligence, Nanjing University, China

2025.10.03



Background

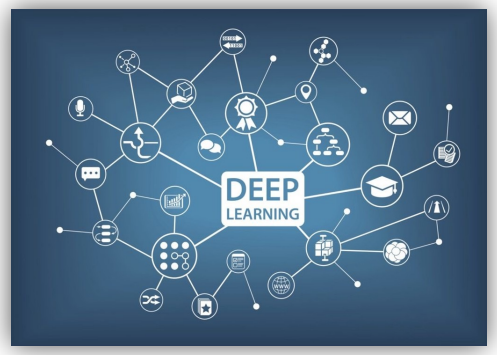
- Nowadays, machine learning has gained ***remarkable success***, with ***large-size models*** have been developed and used:



However, previous methods typically focused on ***closed-world, static environments.***

Background

- Machine learning in ***non-stationary*** environments:



Static environments

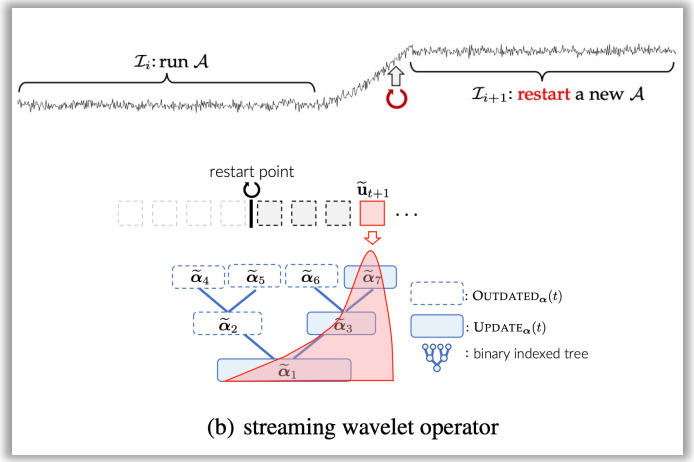
- Training and test data from same distribution

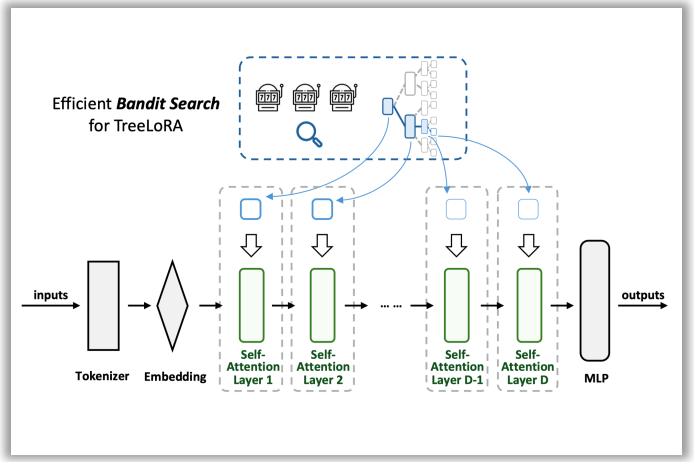
Non-Stationary environments

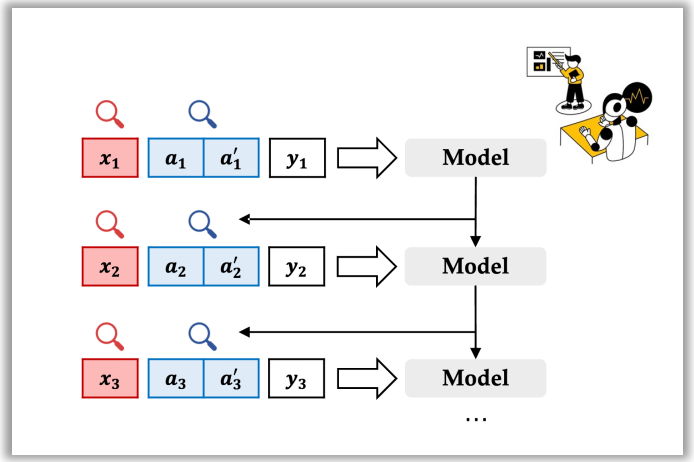
- **Data distribution change**
- New class emerge
- Feature space evolve…

Outline

- Goal: Adapting to **online distribution shift:**







Convex (exp-concave) cases
[ICML'24]

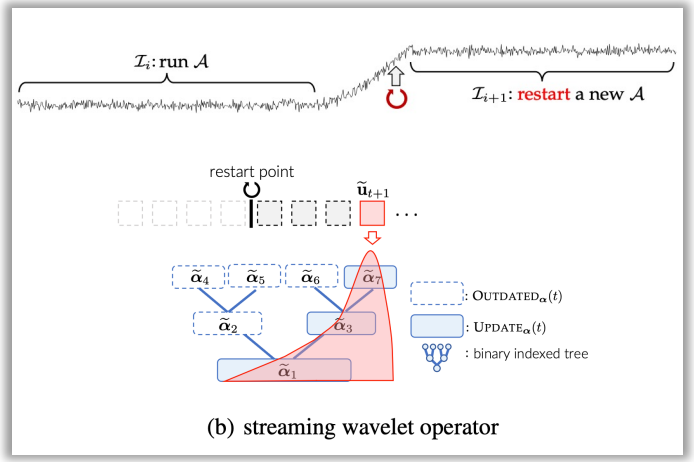
Towards modern ***LLMs***
[ICML'25]

Online (one-pass) ***RLHF***
[NeurIPS'25]

With both theoretical guarantee (***convex cases***) and practical implementation.

Outline: for the classic convex cases

- Goal: Adapting to **online distribution shift:**



(b) streaming wavelet operator

Convex (exp-concave) cases
[ICML'24]

Towards modern *LLMs*
[ICML'25]

Online (one-pass) *RLHF*
[NeurIPS'25]

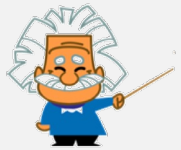
Theoretical guarantee under ***convex (exp-concave)*** cases.

Qian et al., Efficient non-stationary online learning by wavelets with applications to online distribution shift adaptation. **ICML'24.**

Problem Formulation

- Protocol of the **Online Learning** problem:

At each round $t = 1, \ldots, T$:



(1) the learner submits a prediction $\boldsymbol{\theta}_t \in \Theta$;

⇩

(2) simultaneously, the environment picks loss $f_t(\cdot) : \Theta \mapsto \mathbb{R}$;

⇩

(3) the learner suffers loss $f_t(\boldsymbol{\theta}_t)$ and updates model.

Previous Measures

- Worst-case Dynamic Regret

Worst-case Dynamic Regret: compare with the function minimizers

$$\mathbf{Reg}_T^{\mathbf{d}}(\{f_t, \boldsymbol{\theta}_t^\star\}_{t=1}^T) \triangleq \sum_{t=1}^T f_t(\boldsymbol{\theta}_t) - \sum_{t=1}^T f_t(\boldsymbol{\theta}_t^\star)$$

$$(\boldsymbol{\theta}_t^\star \in \arg\min_{\boldsymbol{\theta}\in\Theta} f_t(\boldsymbol{\theta}))$$

However: May lead to ***overfitting*** to sample randomness.

- Consider online supervised learning with loss $f_t(\boldsymbol{\theta}) = \sum_{(\mathbf{x},y)\in S_t} \ell(y; \mathbf{x}^\top \boldsymbol{\theta})$
- Only obtain f_t, but the *expected* $F_t(\boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_t}[\ell(y; \mathbf{x}^\top \boldsymbol{\theta})]$ is our goal

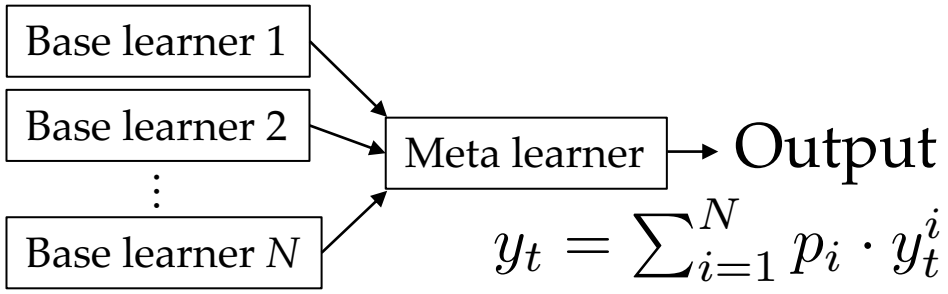
Previous Measures

Universal Dynamic Regret: compare with *any* comparators $\mathbf{u}_1, \ldots, \mathbf{u}_T$

$$\mathbf{Reg}_T^{\mathbf{d}}(\{f_t, \mathbf{u}_t\}_{t=1}^T) \triangleq \sum_{t=1}^T f_t(\boldsymbol{\theta}_t) - \sum_{t=1}^T f_t(\mathbf{u}_t)$$

$\Longrightarrow$ hold ***universally*** for arbitrary comparator sequence.

However: typically need to deploy a two-layer ***online-ensemble***.



- need to maintain multiple ($\approx \log(T)$) base models, when using complex models (such as DNNs), may become ***computational costly.***

Motivation

Motivation: how to attain optimal dynamic regret *without* deploying an ensemble of multiple base models?

weaker → stronger

Worst-case Dynamic regret

$$\sum_{t=1}^{T} f_t(\boldsymbol{\theta}_t) - \sum_{t=1}^{T} f_t(\boldsymbol{\theta}_t^\star)$$

Underlying Dynamic regret

Universal Dynamic regret

$$\sum_{t=1}^{T} f_t(\boldsymbol{\theta}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t)$$

Underlying Dynamic Regret: compare with ground-truth minimizers

$$\mathbf{Reg}_T^{\mathbf{d}}(\{f_t, \mathring{\mathbf{u}}_t\}_{t=1}^T) \triangleq \sum_{t=1}^{T} f_t(\boldsymbol{\theta}_t) - \sum_{t=1}^{T} f_t(\mathring{\mathbf{u}}_t)$$

where $\mathring{\mathbf{u}}_t \in \Theta$ is the ground-truth comparator characterizing the ***underlying distribution*** at round t.

Problem Setup

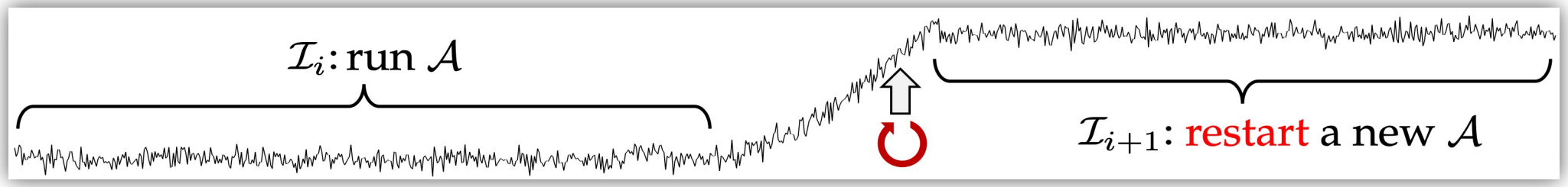
- Although $\mathring{\mathbf{u}}_t$ is not observable, we can obtain a ***noisy estimation*** $\widetilde{\mathbf{u}}_t$

Assumption 1 (observation model). *The learner observes* $\widetilde{\mathbf{u}}_t$ *satisfying* $\mathbb{E}[\widetilde{\mathbf{u}}_t] = \mathring{\mathbf{u}}_t$, *with a bounded variance of* σ^2, *i.e.,* $\mathbb{V}[\widetilde{\mathbf{u}}_t] = \frac{1}{d}\|\widetilde{\mathbf{u}}_t - \mathring{\mathbf{u}}_t\|_2^2 \leq \sigma^2$.

- $\widetilde{\mathbf{u}}_t$ can be obtained by construct an unbiased estimator;
- Sufficiently ***general*** to encompass many real learning problem of interest: online label/covariate shift, etc.;
- By focusing on the specific structure of the stochastic comparator, we achieve a tight regret bound with a ***single-layer algorithm.***

Method: Detection-Restart Framework

- *Overview* of our framework:



- ➢ "Denoise" the noisy observation $\tilde{\mathbf{u}}_t$ to approximate $\mathring{\mathbf{u}}_t$;
- ➢ Restart the algorithm once the abrupt changes are ***detected.***

Intuition: divide the ***non-stationary*** stream into multiple ***stationary*** intervals

Step 1: Detect restarting points

- Detect environment non-stationarity based on ***wavelets***[1, 2]:

Algorithm 1: Detection Module by Wavelets

Input: Restart threshold γ; online algorithm $\mathcal{A}$.
Initialize: coefficient matrix $\widetilde{\boldsymbol{\alpha}} = \mathbf{0}$, time $s = 1$;
for $t = 1, \ldots, T$ **do**
  Update coefficient matrix $\widetilde{\boldsymbol{\alpha}}_{[s,t]}$ as Sec 3.2;
  if $\|\delta_\gamma(\widetilde{\boldsymbol{\alpha}}_{[s,t]})\|_{\mathrm{F}} > \gamma$ **then** ($\gamma \approx 1/\text{Variance}$)
    Restart the online algorithm $\mathcal{A}$;
    Reset coefficient matrix $\widetilde{\boldsymbol{\alpha}} = \mathbf{0}$, set $s = t + 1$;
  Output the prediction $\boldsymbol{\theta}_t$ using $\mathcal{A}$;
  Suffer loss $f_t(\boldsymbol{\theta}_t)$, observe $\widetilde{\mathbf{u}}_t$, and update $\mathcal{A}$;
end

- Maintain only ***one*** learner $\mathcal{A}$
- Maintain wavelet coefficients of the empirical sequence $\{\widetilde{\mathbf{u}}_t\}$
- ***Restart*** $\mathcal{A}$ once the norm of coefficients exceed threshold

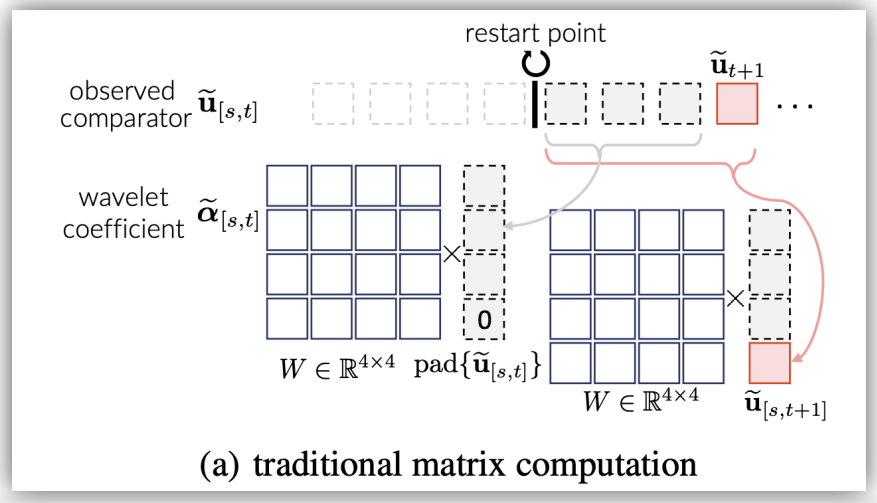
Dheeraj Baby and Yu-Xiang Wang. Online Forecasting of Total-Variation-bounded Sequences. **NeurIPS** 2019.
Dheeraj Baby and Yu-Xiang Wang. Adaptive online estimation of piecewise polynomial trends. **NeurIPS** 2020.

Step 2: Compute the Wavelets

- *Efficiently* calculate wavelet coefficients in an online manner:

Traditional Computation

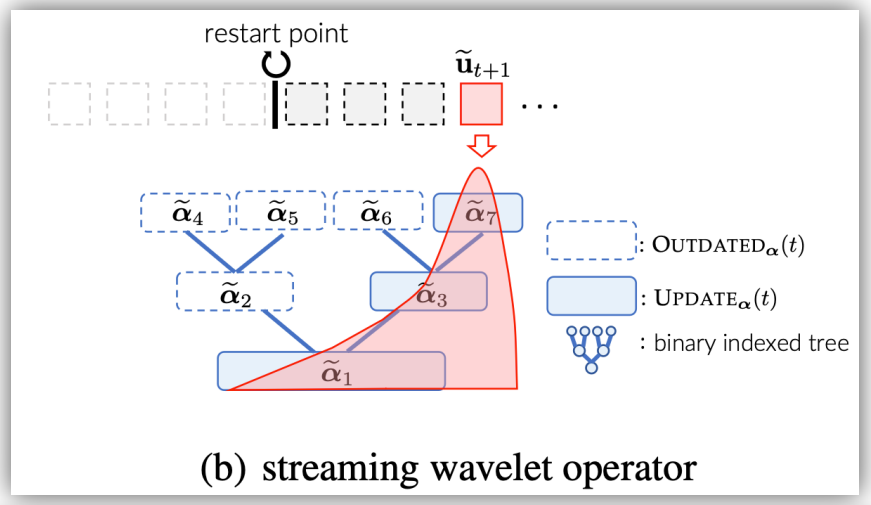
- Matrix multiplication $\widetilde{\boldsymbol{\alpha}}_{[s,t]} = W_{[s,t]}^{\top} \widetilde{\mathbf{u}}_{[s,t]}$
- Store all data, and recalculate all coeffi.
- $\mathcal{O}(T)$ complexity ☹



(a) traditional matrix computation

our *Streaming Wavelet Operator*

- Use a binary indexed tree
- Only lazily update *a portion of* coeffi.
- Only maintain the ***norm*** information
- $\mathcal{O}(\log T)$ complexity 😄



(b) streaming wavelet operator

Baby and Wang (NeurIPS'19) have already proposed efficient updates for ***Haar wavelet***, we further extend to higher-order ***CDJV wavelet***.

Step 3: Combined with Online Alg

- We require online algorithm $\mathcal{A}$ to have the following property:

> The online algorithm $\mathcal{A}$ should perform well in stationary environments.

- Formally,

Requirement 1. *An online algorithm $\mathcal{A}$ running over **any** interval $\mathcal{I}_i = [s_i, e_i] \subseteq [T]$ is required to satisfy*

$$\sum_{t=s_i}^{e_i} f_t(\boldsymbol{\theta}_t) - \sum_{t=s_i}^{e_i} f_t(\mathring{\mathbf{u}}_t) \leq \mathcal{O}\left(\sqrt{|\mathcal{I}_i|} + C_{\mathcal{I}_i}^k\right).$$

where $C_{\mathcal{I}_i}^k$ is the comparator gap, quantify smoothness of underlying comparators.

As it can be seen later, many existing algorithms satisfy this requirement.

Theoretical Guarantee

- Theoretical Guarantees of ***dynamic regret***:

Theorem 1. With prob. at least $1-2/T$, using our detection-restart framework in Algorithm 1 with a $\mathcal{A}$ satisfying Requirement 1, we have

- for ***convex*** function:

$$\mathbf{Reg}_T^{\mathbf{d}}(\{f_t, \mathring{\mathbf{u}}_t\}_{t=1}^T) \leq \widetilde{\mathcal{O}}\left(T^{\frac{2}{3}}(P_T)^{\frac{1}{3}}\right)$$

- for ***exp-concave*** function:

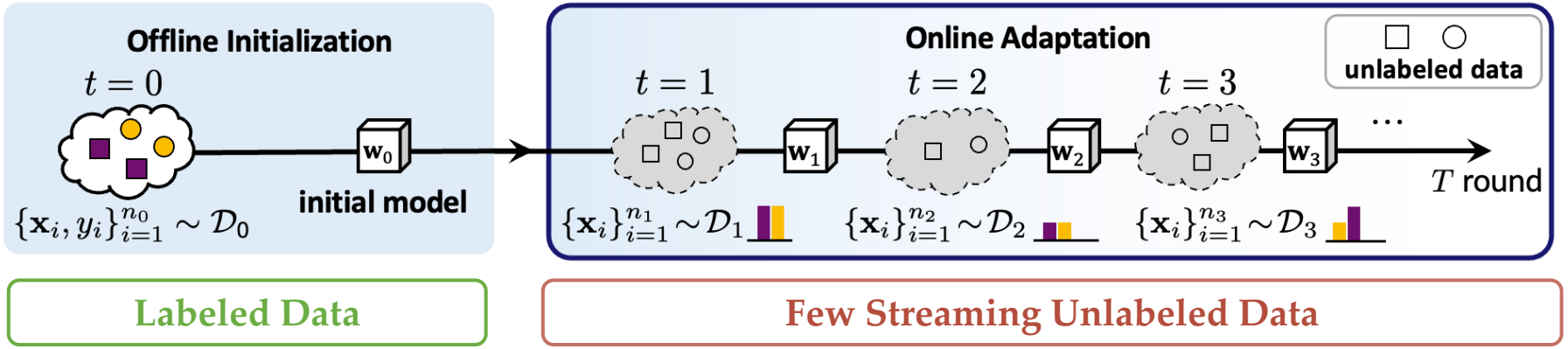
$$\mathbf{Reg}_T^{\mathbf{d}}(\{f_t, \mathring{\mathbf{u}}_t\}_{t=1}^T) \leq \widetilde{\mathcal{O}}\left(T^{\frac{1}{3}}(P_T)^{\frac{2}{3}}\right)$$

where $P_T = \sum_{t=2}^T \|\mathring{\mathbf{u}}_{t-1} - \mathring{\mathbf{u}}_t\|_1$ is the path length.

Remark: *optimal* for exp-concave scenario, but *sub-optimal* for general convex functions.

Application: Online Label Shift (OLS)

- Problem setup of OLS:



Label distribution $\mathcal{D}_t(y)$ changes over time, conditional $\mathcal{D}_t(\mathbf{x} \mid y)$ unchanged.

- Goal: minimize the ***cumulative expected risk***:

$$\mathbf{Reg}_T^{\mathbf{d}}(\{R_t, h_t^\star\}_{t=1}^T) \triangleq \sum_{t=1}^T R_t(\widehat{h}_t) - \sum_{t=1}^T R_t(h_t^\star)$$

$$(h_t^\star \in \arg\min_{h \in \mathcal{H}} R_t(h))$$

Application: Online Label Shift (OLS)

- Goal: minimize the ***cumulative expected risk***:

$$\mathbf{Reg}_T^{\mathbf{d}}(\{R_t, h_t^\star\}_{t=1}^T) \triangleq \sum_{t=1}^T R_t(\widehat{h}_t) - \sum_{t=1}^T R_t(h_t^\star)$$

- ***Reduction*** the OLS problem to *underlying dynamic regret*:

Lemma. Assume $R_t(h)$ is L Lipschitz w.r.t. any $h \in \mathcal{H}$, we have

$$\sum_{t=1}^T R_t(h_t) - \sum_{t=1}^T R_t(h_t^\star) \leq L \sum_{t=1}^T \|\widehat{\boldsymbol{\mu}}_t - \boldsymbol{\mu}_t\|_2,$$

where $\widehat{\boldsymbol{\mu}}_t$ is our predicted class prior and $\boldsymbol{\mu}_t = \mathcal{D}_t(y)$ is the ground-truth one.

Match the form of ***underlying dynamic regret***

Although $\boldsymbol{\mu}_t$ is unknown, BBSE can be used to obtain an ***unbiased*** estimation $\widetilde{\boldsymbol{\mu}}_t$

Application: Online Label Shift (OLS)

- Apply our detection-restart framework to solve OLS:
  - (i) Get unbiased estimation using BBSE (previous label shift estimator);
  - (ii) Maintain wavelet coefficients of the estimated label distribution;
  - (iii) Setting $\mathcal{A}$ as Reweighting or OGD, ***restart*** $\mathcal{A}$ if detecting changes.

- Combined with two kinds of online algorithms:

Reweighting Update as $\mathcal{A}$	OGD Update as $\mathcal{A}$
$[h_t(\mathbf{x})]_j = \frac{1}{Z(\mathbf{x})} \frac{[\widehat{\boldsymbol{\mu}}_t]_j}{\mathcal{D}_0(y=j)} [h_0(\mathbf{x})]_j,\ \forall j \in [K]$	$\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}[\mathbf{w}_t - \eta_t \nabla \widehat{R}_t(\mathbf{w}_t)]$, with $\eta_t = 1/\sqrt{t-s}$
(Update $\widehat{\boldsymbol{\mu}}_t$ by ONS: $\widehat{\boldsymbol{\mu}}_{t+1} = \Pi_{\Delta_K}^{A_t} \left[\widehat{\boldsymbol{\mu}}_t - \frac{1}{\varepsilon} A_t^{-1} \nabla \widetilde{L}_t(\widehat{\boldsymbol{\mu}}_t)\right]$)	$(\widehat{R}_t(\mathbf{w}) = \sum_{j=1}^{K} [\widetilde{\boldsymbol{\mu}}_t]_j \cdot R_0^j(\mathbf{w}))$

Application: Online Label Shift (OLS)

- Theoretical Guarantees for OLS problem:

Theorem 4. *The reweighting-based update satisfies Requirement 1, and using wavelet-based detection-restart framework with this update as $\mathcal{A}$ ensures*

$$\mathbb{E}\left[\mathbf{Reg}_T^{\mathbf{d}}\right] \leq \widetilde{\mathcal{O}}\left(\max\{T^{\frac{k+2}{2k+3}}(P_T^k)^{\frac{1}{2k+3}}, \sqrt{T}\}\right).$$

where $P_T^k = T^k \|\boldsymbol{D}^{k+1}\boldsymbol{\mu}_{[1,T]}\|_1$ is the k-th order path length.

Theorem 5. *The OGD-based update update satisfies Requirement 1, and using wavelet-based detection-restart framework with this update as $\mathcal{A}$ ensures*

$$\mathbb{E}\left[\mathbf{Reg}_T^{\mathbf{d}}\right] \leq \widetilde{\mathcal{O}}\left(\max\{T^{\frac{2}{3}}(P_T^0)^{\frac{1}{3}}, \sqrt{T}\}\right),$$

where $P_T^0 = \sum_{t=2}^{T} \|\boldsymbol{\mu}_t - \boldsymbol{\mu}_{t-1}\|_1$ is the path length.

Both achieving the ***optimal*** dynamic regrets for online label shift.

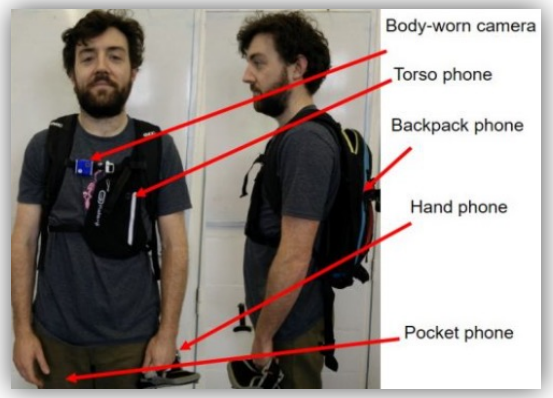
Experiments

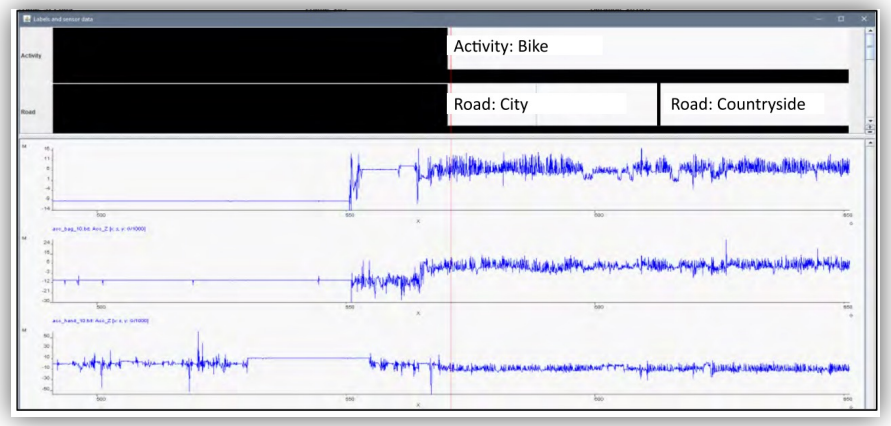
- Online label shift scenario:
  - Compare with SOTA algorithms in *five* benchmark datasets
  - Simulate different ***patterns*** of label shifts

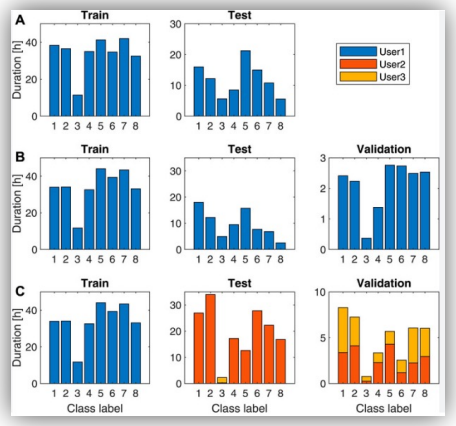
	Linear Shift							Square Shift						
	FIX	**OGD**	**Reweight**	**FTFWH**	**ATLAS**	**Wav-O**	**Wav-R**	**FIX**	**OGD**	**Reweight**	**FTFWH**	**ATLAS**	**Wav-O**	**Wav-R**
CIFAR10	∘20.89 ±0.13	15.92 ±0.13	∘18.25 ±0.45	∘ 17.32 ±0.15	15.75 ±0.12	**15.52** ±0.15	15.68 ±0.14	∘20.79 ±0.04	∘16.31 ±0.13	∘17.38 ±0.15	∘16.70 ±0.14	∘15.21 ±0.08	**14.72** ±0.11	15.55 ±0.13
CINIC10	∘34.56 ±0.24	∘27.31 ±0.21	∘32.42 ±2.55	∘ 28.55 ±0.12	∘26.44 ±0.21	**26.11** ±0.13	28.21 ±0.11	∘34.01 ±0.12	∘28.96 ±0.10	∘28.62 ±0.13	∘28.01 ±0.05	∘27.01 ±0.11	**26.12** ±0.05	26.65 ±0.13
EuroSAT	∘15.42 ±0.12	9.13 ±0.15	∘12.34 ±3.17	∘ 11.35 ±0.12	7.21 ±0.13	**7.15** ±0.12	7.25 ±0.12	∘14.19 ±0.15	7.33 ±0.11	∘ 8.88 ±0.09	∘10.19 ±0.12	**6.99** ±0.09	7.43 ±0.05	7.82 ±0.03
Fashion	∘11.35 ±0.05	7.98 ±0.06	8.15 ±0.05	**7.84** ±0.08	8.39 ±0.09	8.34 ±0.13	8.37 ±0.08	∘11.94 ±0.13	∘ 8.46 ±0.07	∘ 8.67 ±0.11	∘ 8.28 ±0.13	∘ 8.13 ±0.12	7.88 ±0.07	**7.32** ±0.07
MNIST	∘ 1.72 ±0.03	1.13 ±0.03	∘ 1.32 ±0.04	∘ 1.25 ±0.03	**1.07** ±0.05	1.09 ±0.02	1.13 ±0.03	∘ 1.83 ±0.08	∘ 1.17 ±0.07	∘ 1.36 ±0.08	∘ 1.17 ±0.07	1.05 ±0.02	1.12 ±0.04	**1.03** ±0.03

Experiments

- Online label shift scenario:
  - Compare with SOTA algorithms in ***real-world*** locomotion dataset
  - ***Goal:*** distinguish the human locomotion in real-life
  - The distribution of human motion types changes over time, leading to the ***label shift*** occurring in the data stream







Experiments

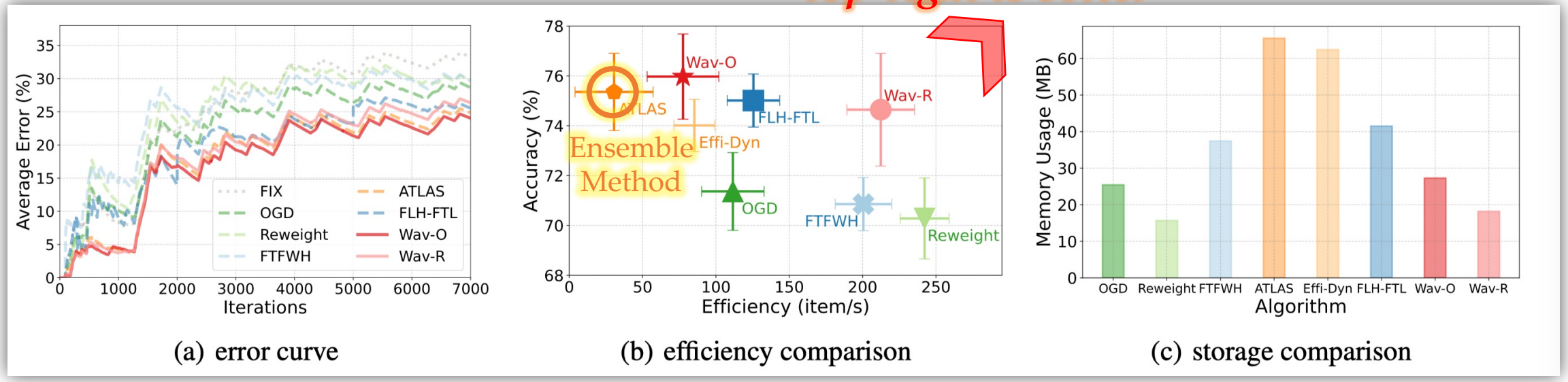
- Online label shift scenario:
  - Compare with SOTA algorithms in ***real-world*** locomotion dataset
  - ***Goal:*** distinguish the human locomotion in real-life
  - The distribution of human motion types changes over time, leading to the ***label shift*** occurring in the data stream

Table 3: Average error (%) of different algorithms on the real-world *online label shift application*: locomotion dataset (Gjoreski et al., 2018), where *Wav-O* represents using our wavelet-based detection-restart framework with OGD, and *Wav-R* represents using our framework with reweighting. We report the mean and standard deviation over five runs.

	FIX	OGD	Reweight	FTFWH	ATLAS	Effi-Dyn	FLH-FTL	LAME	Wav-O			Wav-R		
									$k=0$	$k=1$	$k=2$	$k=0$	$k=1$	$k=2$
Locomotion	33.32 ±1.05	28.64 ± 1.67	29.72 ± 1.57	29.14 ± 1.12	24.64 ± 1.55	25.45 ± 2.35	24.83 ± 1.45	24.92 ± 1.23	24.03 ± 1.71	24.12 ± 1.54	**23.99 ± 1.65**	25.35 ± 2.23	25.41 ± 2.35	25.36 ± 2.23

Experiments

- Online label shift scenario:
  - Compare with SOTA algorithms in ***real-world*** locomotion dataset

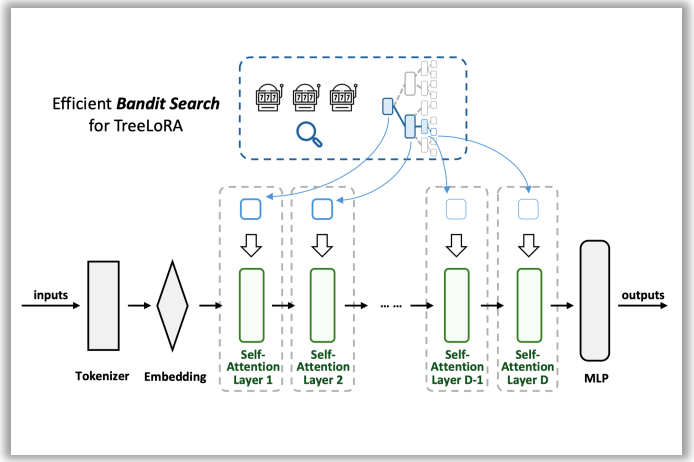


Validate the **effectiveness** and **efficiency** of our algorithms.

Outline: towards modern LLMs

- Goal: Adapting to **online distribution shift:**

Convex (exp-concave) cases
[ICML'24]



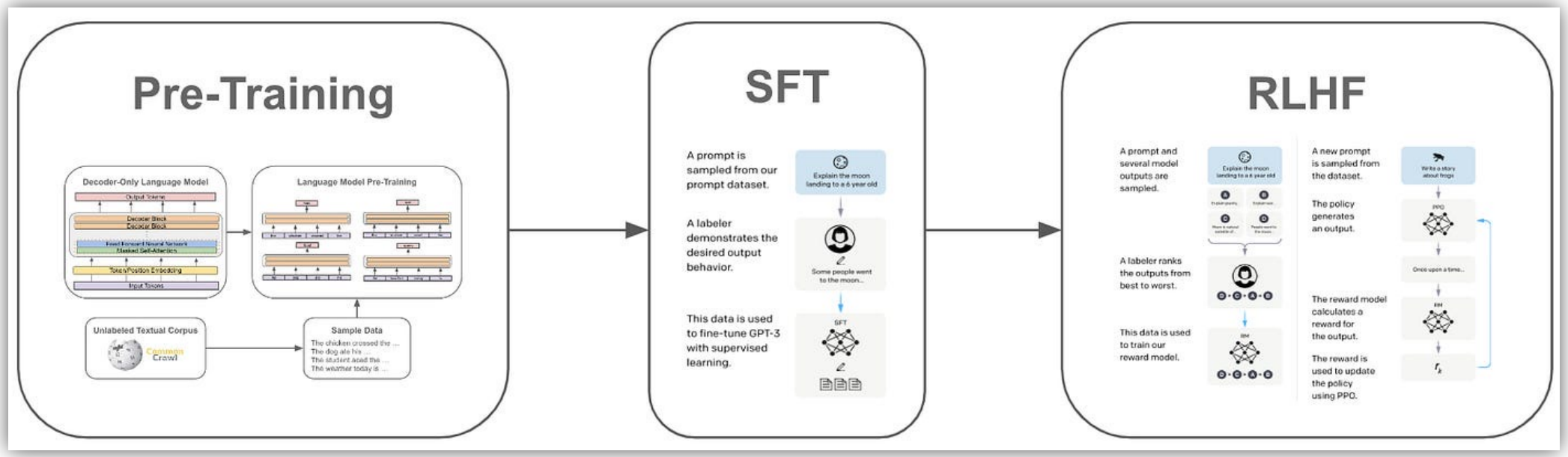
Towards modern ***LLMs***
[ICML'25]

Online (one-pass) *RLHF*
[NeurIPS'25]

Practical implementation for modern LLMs.

Qian et al., TreeLoRA: Efficient Continual Learning via Layer-Wise LoRAs Guided by a Hierarchical Gradient-Similarity Tree. **ICML'25.**

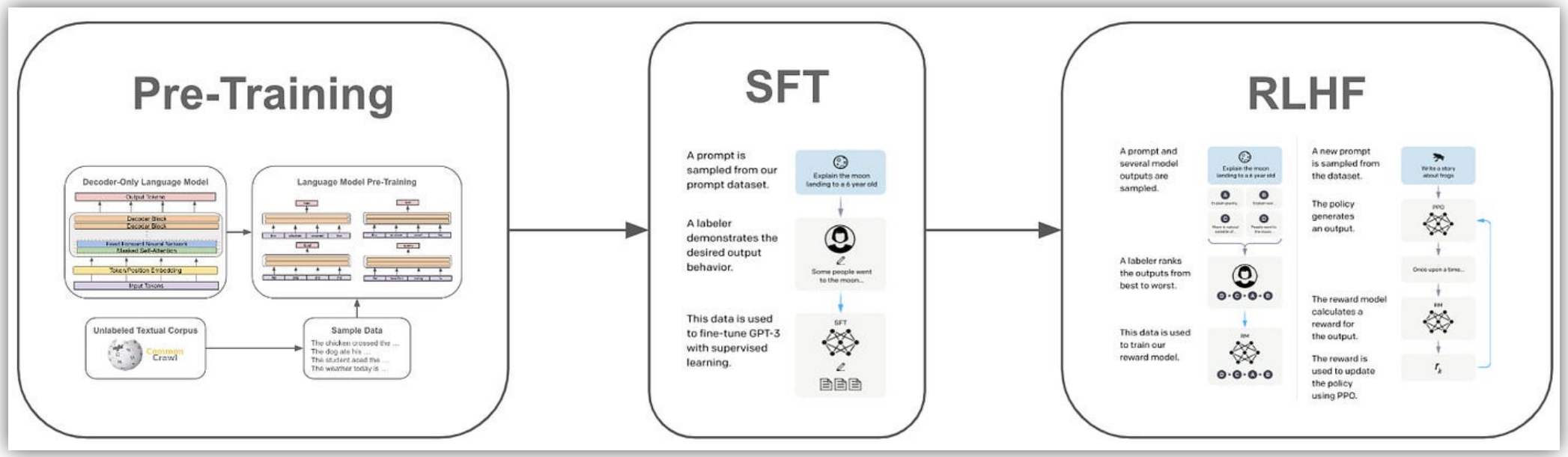
Warm Up: Large Pretrained Models (PRMs)



- Three training-stages of Pre-trained Models (PRMs):
  - 1. ***Pre-Training***: obtain basic knowledge.
  - 2. ***SFT***: improve model's ability to follow instructions.
  - 3. ***RLHF*** (or ***preference optimization***): align models to human values.

Post-Training

Warm Up: Large Pretrained Models (PRMs)



How to develop an ***efficient*** continual learning approach, particularly fitted for ***large pre-trained models***?

Key Observation and Motivation

- Our key motivation and observation:
  - ***Reuse*** historical tasks to accelerate and improve current task's learning!

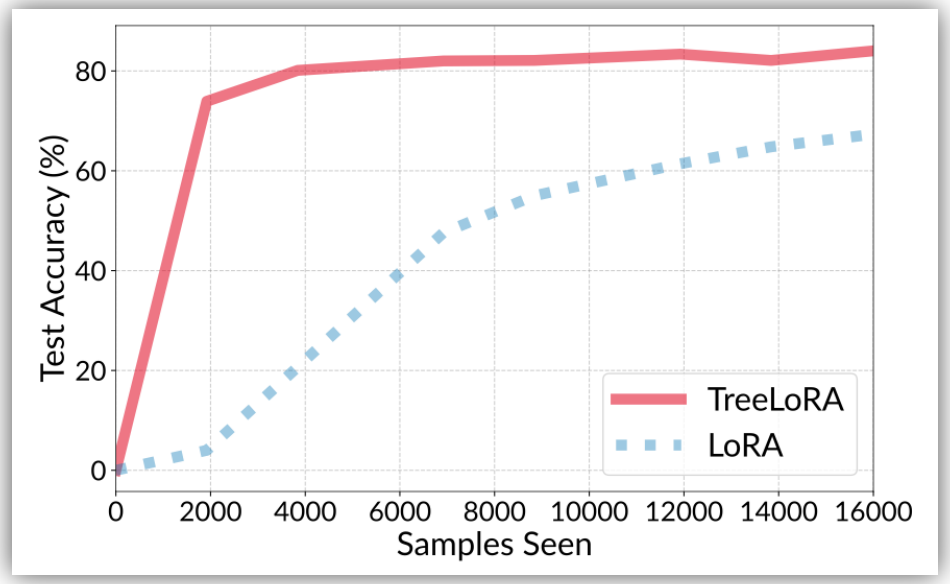
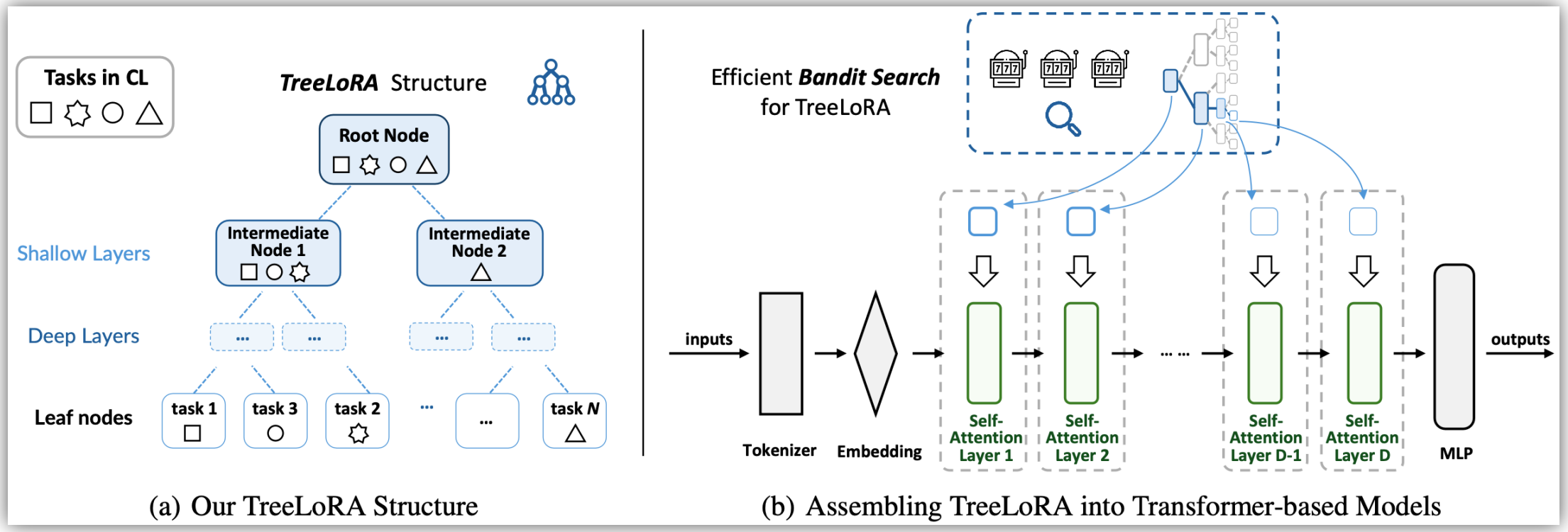


Figure 1: Test accuracy vs. the number of arrived samples for our *TreeLoRA* and the vanilla *LoRA* (Hu et al., 2022) on the Split CIFAR-100 dataset when encountering a new task, demonstrating our efficiency to adapt to new tasks, owing to mining the hierarchical gradient-similarity structure to explore task-shared knowledge.

How to ***efficiently*** mine and manage of ***task-similarity structure***?

Method: *TreeLoRA* (K-D Tree of Low-Rank Adapters)



(a) Our TreeLoRA Structure (b) Assembling TreeLoRA into Transformer-based Models

- Root Node: all tasks
- Tree Node: a task group
- Leaf Node: one task

- Cluster task by *gradients*:

$$\|\mathbf{g}_i - \mathbf{g}_{i'}\|_1 \leq \delta, \forall i, i' \in \mathcal{N}$$

$$\mathbf{g}_k \triangleq \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_n}\left[\nabla \ell(\mathbf{w}_k; \mathbf{x}, y)\right]$$

Method: ***Bandit Algorithm*** to capture task structure

- Adaptive Tree Construction via ***Bandit Algorithm***:
  - Use Lower Confidence Bound (LCB) to efficiently search similar task:

$$\widehat{\mu}_k = \frac{1}{|\text{Select}_k|} \sum_{\tau \in \{\text{Select}_k\}} \hat{\xi}_\tau^k \qquad \text{LCB}_k = \begin{cases} \widehat{\mu}_k - 2\sqrt{\frac{\log t}{n_k}}, & \text{if } k \in \mathcal{L} \\ \max\left\{ \min_{j \in \mathcal{C}} \left\{ \widehat{\mu}_j - 2\sqrt{\frac{\log t}{n_j}} - \delta \right\} \right\}, & \text{if } k \notin \mathcal{L} \end{cases}$$

Theorem 1 (Regret Bound). Assume A_η with an exponential sequence $\delta_d = \delta\gamma^d$. Then, with probability at least $1 - 1/T$, the cumulative regret of TreeLoRA after T rounds satisfies:

$$\text{Reg}(T) \leq \mathcal{O}\left(\sqrt{T\,|J_\eta| \log\left(\frac{NT}{|J_\eta|}\right)} + \frac{\delta^c}{\eta^{2+c}} \log\left(\frac{NT}{\eta^2}\right) \right),$$

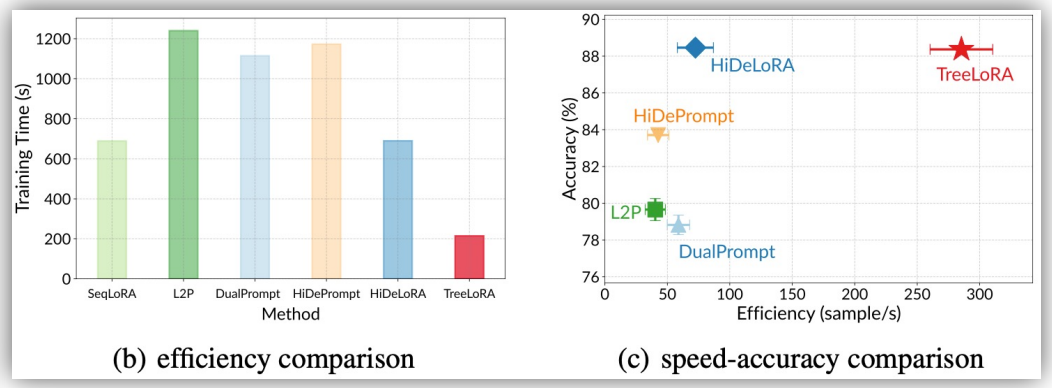
Avoiding $O(T)$ computational complexity each round, ***only need O(1).***

Experiments: ViTs and LLMs

Learning And Mining from DatA

- (i) On vision transformers (ViTs): ViT-B/16 model iBOT-21K as backbone, three benchmark datasets.

	SeqLoRA	GEM	EWC	L2P	DualPrompt	HiDePrompt	HiDeLoRA	TreeLoRA
				Split CIFAR-100				
Acc (%) ↑	10.35 ± 0.35	76.46 ± 0.35	84.79 ± 0.09	79.66 ± 0.60	78.83 ± 0.53	83.71 ± 0.03	88.46 ± 0.04	**88.54 ± 0.05**
Bwt (%) ↓	97.62 ± 0.11	4.37 ± 0.30	5.22 ± 0.04	4.40 ± 0.01	4.89 ± 1.67	5.34 ± 0.26	4.33 ± 0.41	4.37 ± 0.15
Time (s) ↓	694 ± 8	22456 ± 31	4091 ± 14	1240 ± 12	1173 ± 14	1205 ± 11	692 ± 7	**214 ± 4**
				Split ImageNet-R				
Acc (%) ↑	8.65 ± 0.15	54.45 ± 0.76	55.43 ± 0.54	56.75 ± 0.61	54.81 ± 0.40	62.94 ± 0.91	**72.28 ± 0.15**	71.94 ± 0.47
Bwt (%) ↓	82.55 ± 0.93	2.16 ± 0.40	9.10 ± 0.97	3.66 ± 0.16	4.39 ± 0.11	3.18 ± 0.56	4.16 ± 0.05	4.06 ± 0.40
Time (s) ↓	797 ± 11	17244 ± 29	4727 ± 21	1431 ± 10	1357 ± 15	1354 ± 13	796 ± 9	**260 ± 5**
				Split CUB-200				
Acc (%) ↑	9.10 ± 0.41	61.28 ± 0.45	36.34 ± 0.34	39.72 ± 0.34	37.52 ± 0.37	63.86 ± 0.11	73.48 ± 1.35	**73.66 ± 0.22**
Bwt (%) ↓	83.45 ± 0.23	2.64 ± 0.30	6.65 ± 0.53	7.98 ± 0.70	11.91 ± 0.04	5.57 ± 0.23	5.38 ± 0.21	4.87 ± 0.30
Time (s) ↓	194 ± 11	7233 ± 18	1702 ± 9	333 ± 5	315 ± 4	307 ± 1	194 ± 3	**86 ± 3**



(b) efficiency comparison (c) speed-accuracy comparison

Performance vs. Efficiency

- (ii) On different large language models (LLMs):

	FIX (ICL)	SeqLoRA	OGD	GEM	EWC	L2P	DualPrompt	HiDeLoRA	O-LoRA	TreeLoRA
					mistralai / Mistral-7B-Instruct-v0.3					
Op (%) ↑	45.04±0.3	46.94±1.2	45.44±1.6	52.32±1.2	52.45±1.3	49.32±0.8	51.14±1.2	51.81±0.9	52.02±0.8	**54.77±1.1**
Bwt (%) ↓	—	11.41±0.6	12.75±0.7	6.01±0.3	5.98±0.8	5.34±0.6	6.13±0.5	6.25±0.3	8.13±0.6	3.77±0.4
Time (s) ↓	—	1091±25	6952±156	7937±167	52739±198	975±23	983±24	1288±28	1302±29	**510±20**
					meta-llama / LLaMA-2-7B-Chat					
Op (%) ↑	38.94±0.3	34.30±1.2	42.09±1.6	40.08±1.6	42.36±1.2	36.23±0.8	37.69±1.2	41.60±0.8	42.78±0.8	**43.52±1.0**
Bwt (%) ↓	—	18.50±0.8	8.06±1.2	6.77±1.2	5.97±0.8	8.25±0.8	8.03±0.8	7.12±0.4	7.16±0.4	3.46±0.4
Time (s) ↓	—	1132±26	6416±145	7385±158	50283±195	899±22	912±23	1286±27	1293±28	**485±18**
					google / Gemma-2B-it					
Op (%) ↑	32.30±0.2	31.89±0.8	32.85±1.4	26.48±1.5	28.35±1.6	31.14±1.2	32.42±1.0	33.25±0.9	**33.73±0.8**	33.41±0.9
Bwt (%) ↓	—	15.28±0.4	12.27±0.9	18.25±0.9	16.96±1.2	15.77±0.7	14.25±0.5	13.66±0.5	12.36±0.4	8.50±0.5
Time (s) ↓	—	510±15	2534±85	2892±92	17791±185	413±13	415±13	598±16	592±16	**271±15**
					meta-llama / LLaMA-3.2-1B-Instruct					
Op (%) ↑	31.16±0.4	29.73±1.6	30.12±2.0	32.19±2.0	31.96±1.6	29.38±1.2	30.76±1.2	33.73±1.2	32.94±0.8	**36.14±0.7**
Bwt (%) ↓	—	17.03±1.2	15.20±1.6	10.74±1.6	11.62±1.2	13.57±0.8	11.34±0.8	12.36±0.8	12.89±1.2	7.36±0.8
Time (s) ↓	—	482±14	1345±45	1547±52	8893±165	270±11	281±11	450±14	448±14	**179±13**

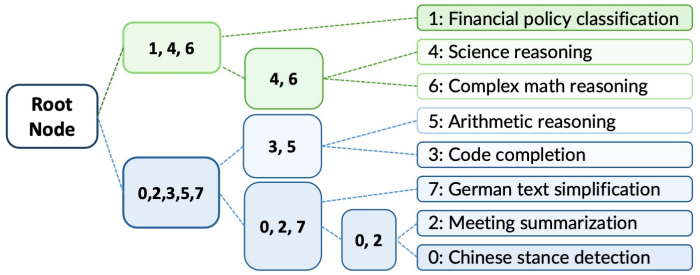


Figure 5: Visualization of the tree structure learned by our approach, using Mistral-7B-Instruct-v0.3 as the foundation model.

Capture the task structure.

Reg coeff λ		Learning rate α	
0.1	88.54 ± 0.05	0.001	87.97 ± 0.05
0.3	88.58 ± 0.04	0.003	88.56 ± 0.15
0.5	88.51 ± 0.03	0.005	88.54 ± 0.05
0.7	88.49 ± 0.05	0.007	88.56 ± 0.03
1.0	88.52 ± 0.02	0.010	88.34 ± 0.17

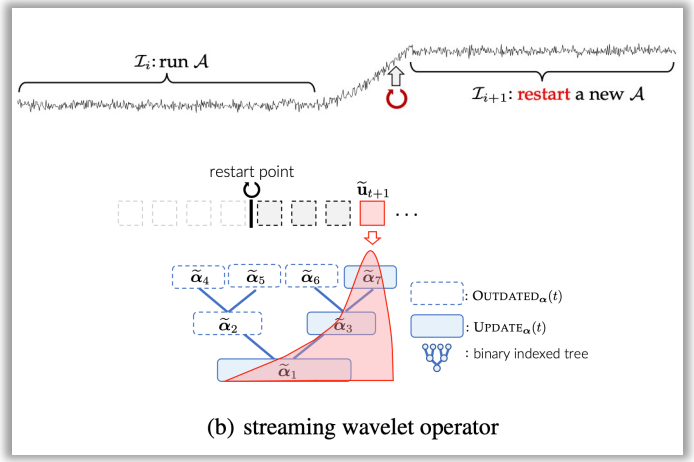
Method	FLOPs	Parameter Complexity
OGD	2.8×10^{10}	$\mathcal{O}(mn)$
LoRA	4.2×10^{6}	$\mathcal{O}((m+n)r)$
O-LoRA	4.2×10^{7}	$\mathcal{O}((m+n)rN)$
TreeLoRA	4.2×10^{6}	$\mathcal{O}((m+n)r+Nr)$

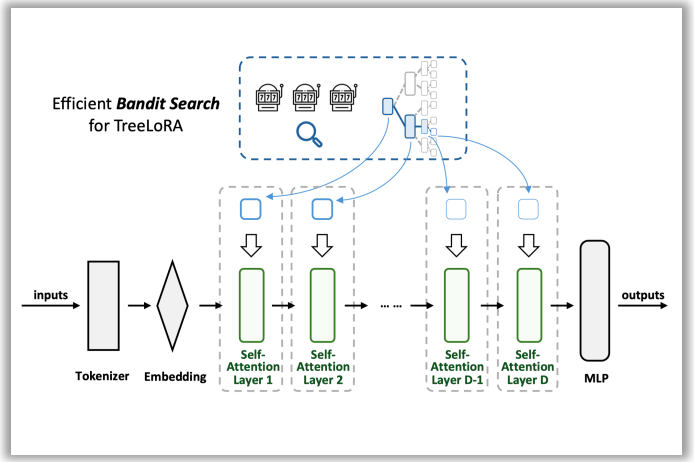
Better efficiency.

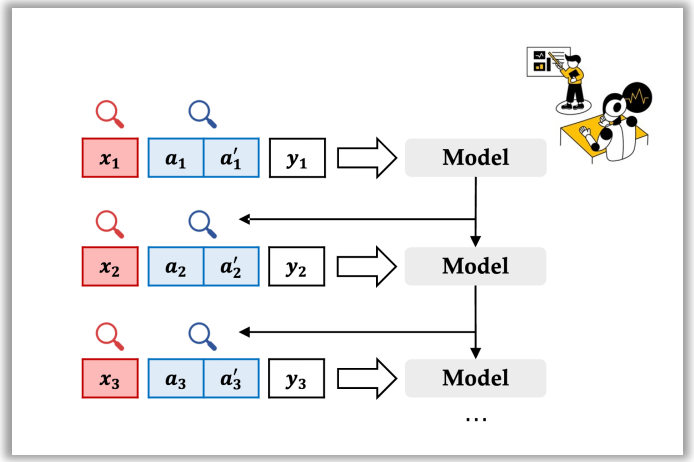
Achieves comparable performance with SOTAs, while achieving ***better efficiency.***

Outline: Online *RLHF*

- Goal: Efficiently Online *RLHF*:







Convex (exp-concave) cases
[ICML'24]

Towards modern ***LLMs***
[ICML'25]

Online (one-pass) ***RLHF***
[NeurIPS'25]

With both theoretical guarantee (***convex cases***) and practical implementation.

WarmUp: *RLHF*

- Problem Formulation in RLHF (Reinforcement Learning from Human Feedback):
  - **Input:** a preference 4-argument tuple (x, a, a', y)
    - x: the prompt: "Please write a joke for me."
    - a: the first response: "Sorry, I can't."
    - a': the second response: "Here is a joke for you: …"
    - $y \in \{0, 1\}$: the label (human's preference): a'
  - RLHF wants to use input to improve LLM
    - i.e. ***align LLM with human's preference***
  - **Output:** First train a good ***reward model;*** then a ***fine-tuned LLM***

Previous Method for *RLHF*

- **Step 1:** Sample prompts and queries using SFT model:

Prompt x —SFT model→ $(a, a') \sim \pi_{\text{sft}}(a \mid x)$ —Query→ Which one is better?

- **Step 2:** use ***Bradley-Terry Model*** **(BT Model)** for the reward model:

Definition 1 (Bradley-Terry Model). Given a context $x \in \mathcal{X}$ and two actions $a, a' \in \mathcal{A}$, the probability of the human preferring action a over action a' is given by

$$\mathbb{P}\left(a \succ a' \mid x\right) = \frac{\exp\left(r\left(x, a\right)\right)}{\exp\left(r\left(x, a\right)\right) + \exp\left(r\left(x, a'\right)\right)}$$

r is the latent function.

- **Step 3:** Learning reward model using *Maximum Likelihood Estimation* (MLE):

$$\arg\min_{r_\phi} \mathcal{L}_R\left(r_\phi, \mathcal{D}\right) = -\mathbb{E}_{(x, a_w, a_l) \sim \mathcal{D}}\left[\log \sigma\left(r_\phi\left(x, a_w\right) - r_\phi\left(x, a_l\right)\right)\right] \quad \left(\sigma(w) = \frac{1}{1 + e^{-w}}\right)$$

Motivation and Goal: *Efficient* RLHF

- ***Computational challenge*** of previous MLE-based methods:
  - Need to **store all previous data**, storage expensive;
  - Need to **re-compute all previous data**, costly for online scenarios.

Computational and storage cost at round t is $O(t)$!

- We aim to take a ***unified view*** for the ***efficient RLHF***:

Goal: statistically and computationally efficient
online RLHF algorithms via ***one-pass reward modeling***.

Method: a Unified View from *Contextual Bandits*

- Revisiting RLHF from the perspective of ***contextual bandits***:

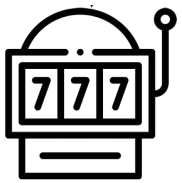
- prompt $\boldsymbol{x}$
- actions a, a'
- ***Goal*:** better reward model $r(\cdot,\cdot)$

Bradley-Terry (BT) Model

$$\mathbb{P}\left(a \succ a' \mid x\right) = \frac{\exp\left(r\left(x, a\right)\right)}{\exp\left(r\left(x, a\right)\right) + \exp\left(r\left(x, a'\right)\right)}$$

Linear reward model $r(x, a) = \phi(x, a)^\top \theta^*$

- context $\boldsymbol{x}$
- arms a, a'
- ***Goal*:** better model $\tilde{\theta}$



Logistic Contextual Bandits

$$\mathbb{P}\left[y = 1 \mid x, a, a'\right] = \sigma\left(\phi(x, a)^\top \theta^* - \phi\left(x, a'\right)^\top \theta^*\right)$$

$\phi(\cdot,\cdot)$ is the known feature mapping

Method: from *MLE* to *OMD*

- Replace MLE with **OMD** inspired by [Zhang and Sugiyama, 2023] :

MLE

$$\widehat{\theta}_{k+1,h} = \arg\min_{\theta\in\mathbb{R}^d} \sum_{i=1}^{k} \underbrace{\sum_{s'\in\mathcal{S}_{i,h}} -y_{i,h}^{s'} \log p_{i,h}^{s'}(\theta) + \frac{\lambda_k}{2}\|\theta\|_2^2}_{\triangleq \ell_{i,h}(\theta)}$$

one-pass update by OMD ⇩ "lookahead" local norm to keep historical information

***implicit* OMD**

$$\bar{\theta}_{k+1,h} = \arg\min_{\theta\in\Theta} \left\{ \ell_{k,h}(\theta) + \frac{1}{2\eta} \left\| \theta - \bar{\theta}_{k,h} \right\|_{\bar{\mathcal{H}}_{k,h}}^2 \right\},$$

where $\bar{\mathcal{H}}_{k,h} \triangleq \sum_{i=1}^{k-1} H_{i,h}\left(\bar{\theta}_{i+1,h}\right)$ is a particular local norm.

Still no closed-form solution!

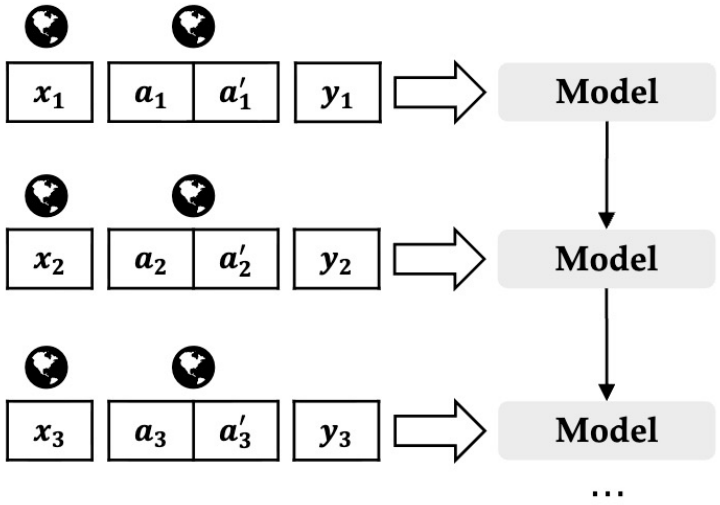
second-order Taylor expansion ⇩ $\widetilde{\ell}_{k,h}(\theta) = \ell_{k,h}(\widetilde{\theta}_{k,h}) + \langle \nabla \ell_{k,h}(\widetilde{\theta}_{k,h}), \theta - \widetilde{\theta}_{k,h} \rangle + \frac{1}{2}\|\theta - \widetilde{\theta}_{k,h}\|_{H_{k,h}(\widetilde{\theta}_{k,h})}^2$

standard OMD

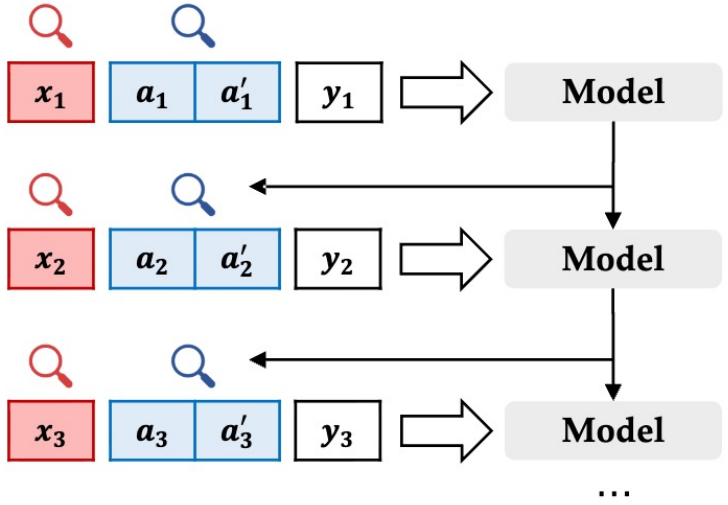
$$\widetilde{\theta}_{k+1,h} = \arg\min_{\theta\in\Theta} \left\{ \left\langle \nabla \ell_{k,h}(\widetilde{\theta}_{k,h}), \theta - \widetilde{\theta}_{k,h} \right\rangle + \frac{1}{2\eta} \left\| \theta - \widetilde{\theta}_{k,h} \right\|_{\widetilde{\mathcal{H}}_{k,h}}^2 \right\},$$

where $\widetilde{\mathcal{H}}_{k,h} = \eta H_{k,h}(\widetilde{\theta}_{k,h}) + \sum_{i=1}^{k-1} H_{i,h}(\widetilde{\theta}_{i+1,h})$ incoporates additional ***second-order quantity***.

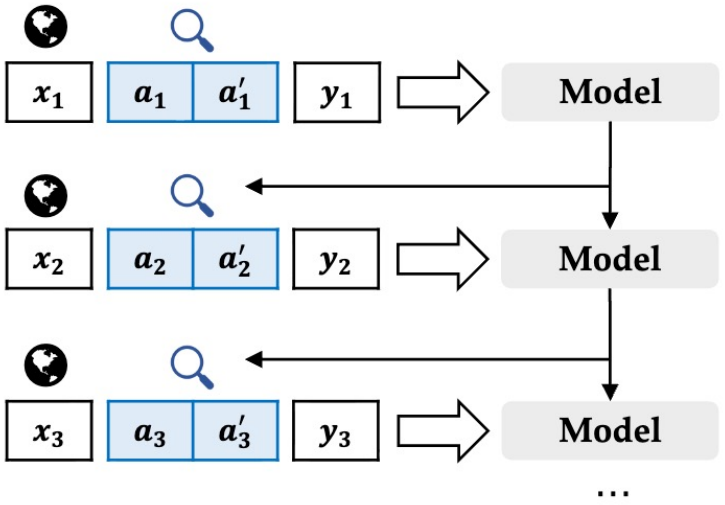
Method: Applied in Three Stages



(a) Passive Data Collection



(b) Active Data Collection



(c) Deployment-Time Adaptation

OMD update:

$$\widetilde{\theta}_{t+1} = \arg\min_{\theta\in\Theta} \left\{ \langle g_t(\widetilde{\theta}_t), \theta \rangle + \frac{1}{2\eta} \left\| \theta - \widetilde{\theta}_t \right\|_{\tilde{\mathcal{H}}_t}^2 \right\}$$

Active select samples:

$$(x_t, a_t, a_t') = \arg\max_{x,a,a'\in\mathcal{X}\times\mathcal{A}\times\mathcal{A}} \left\{ \left\| \phi(x,a) - \phi(x,a') \right\|_{\mathcal{H}_t^{-1}} \right\}$$

Choose dual actions:

$$a_t = \arg\max_{a\in\mathcal{A}} \phi(x_t, a)^\top \tilde{\theta}_t, \text{ and}$$

$$a_t' = \arg\max_{a\in\mathcal{A}} \phi(x_t, a)^\top \tilde{\theta}_t + 2\tilde{\beta} \left\| \phi(x_t, a) - \phi(x_t, a_t) \right\|_{\mathcal{H}_t^{-1}}$$

+ OMD update:

$$\widetilde{\theta}_{t+1} = \arg\min_{\theta\in\Theta} \left\{ \langle g_t(\widetilde{\theta}_t), \theta \rangle + \frac{1}{2\eta} \left\| \theta - \widetilde{\theta}_t \right\|_{\tilde{\mathcal{H}}_t}^2 \right\}$$

Can be applied to ***a variety of*** online/offline RLHF scenarios

Theoretical Improvements

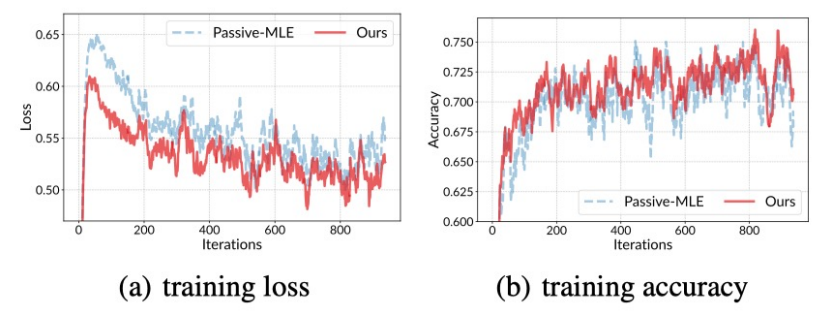
- ***Theoretically:*** improved *statistical* and *computational* efficiency:

Setting	Context	Action	Gap/Regret	Time	Storage	Reference
Passive	🌍	🌍	$\widetilde{\mathcal{O}}(\sqrt{d}\cdot\kappa\|\Phi\|_{V_T^{-1}})$	$\mathcal{O}(\log T)^*$	$\mathcal{O}(t)$	Zhu et al. [2023]
			$\widetilde{\mathcal{O}}(\sqrt{d}\cdot\|\Phi\|_{\mathcal{H}_T^{-1}})$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	Ours (Theorem 1)
Active	🔍	🔍	$\widetilde{\mathcal{O}}(d\sqrt{\kappa/T})$	$\mathcal{O}(t\log t)$	$\mathcal{O}(t)$	Das et al. [2024]
			$\widetilde{\mathcal{O}}(d\sqrt{\kappa/T})$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	Ours (Theorem 2)
Deployment	🌍	🔍	$\widetilde{\mathcal{O}}(d\kappa\sqrt{T})$	$\mathcal{O}(t\log t)$	$\mathcal{O}(t)$	Saha et al. [2023]
			$\widetilde{\mathcal{O}}(d\sqrt{\kappa T})$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	Ours (Theorem 3)

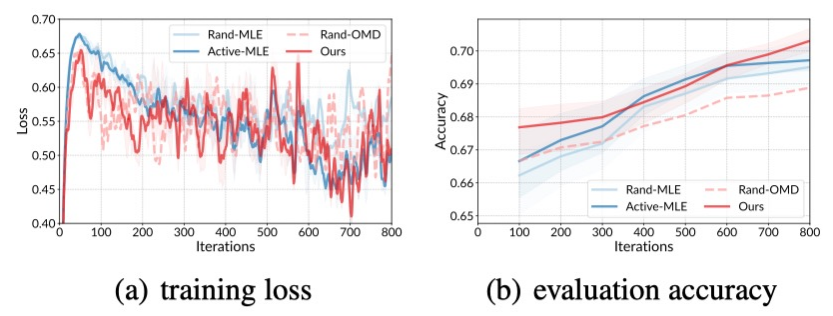
Achieving better regrets with ***less*** time/storage complexity

Experiments: LLM's RLHF

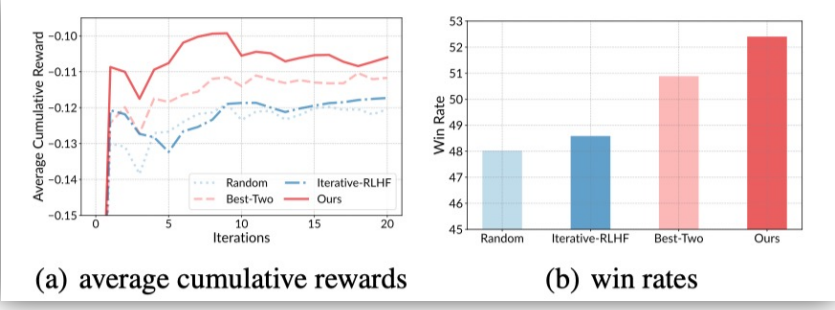
- Foundation Models: ***LLaMA 3 8B*** & ***Qwen 2.5-7B***
- Dataset: ***Ultrafeedback*** & ***Mixture2***



(1) Passive Data Collection



(2) Active Data Collection

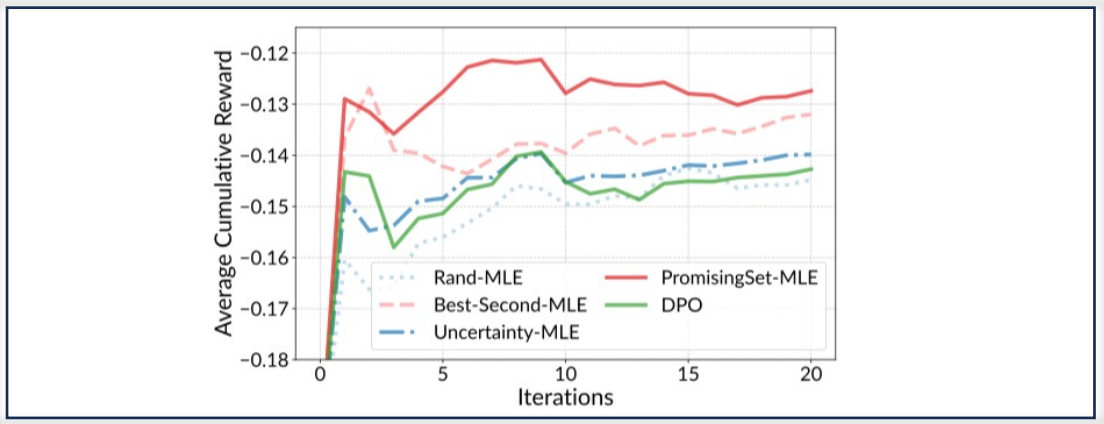


(3) Deployment-time Adaptation

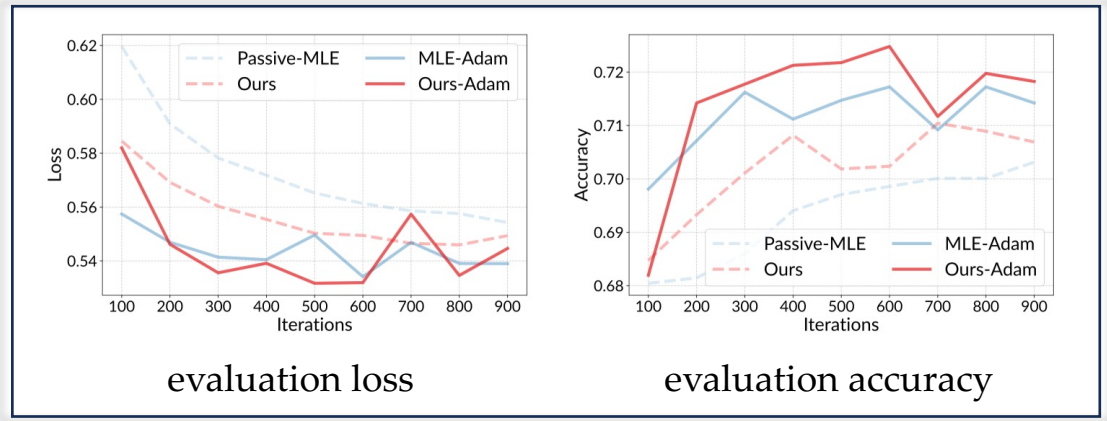
Method	ACC (%)	Time (s)
Rand-MLE	69.51±0.5	4876±47
Active-MLE	69.82±0.4	4982±52
Rand-OMD	68.97±0.6	1456±31
Ours	70.43±0.3	1489±36

(4) Efficiency Comparison

Experiments

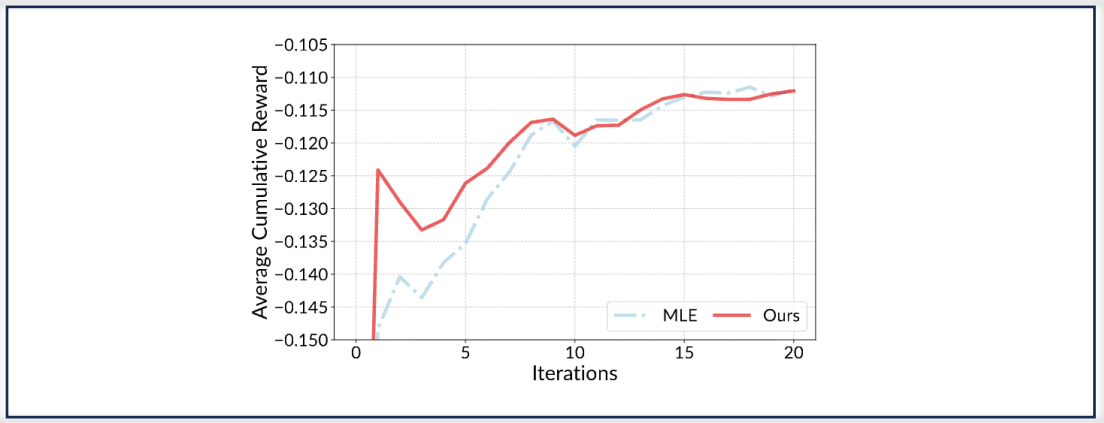


Our method surpass ***DPO*** in deployment stage.



evaluation loss evaluation accuracy

Our method is also comparable with ***Adam***.



Full parameter update experiment.

Achieve comparable performance with ***improved efficiency***.

Efficient **Non-stationary Online Learning** by *Wavelets* with Applications to Online Distribution Shift Adaptation

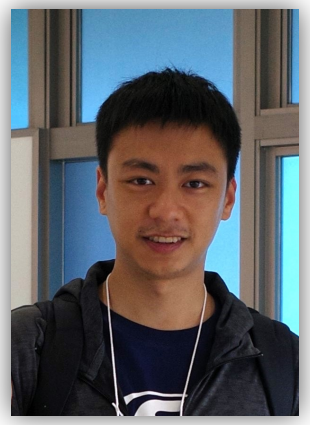
Yu-Yang Qian [1 2] **Peng Zhao** [1 2] **Yu-Jie Zhang** [3] **Masashi Sugiyama** [4 3] **Zhi-Hua Zhou** [1 2]

[1] National Key Laboratory for Novel Software Technology, Nanjing University, China
[2] School of Artificial Intelligence, Nanjing University, China
[3] The University of Tokyo, Chiba, Japan
[4] RIKEN AIP, Tokyo, Japan

[ICML'24]

Yu-Yang Qian

Peng Zhao



Yu-Jie Zhang

Masashi Sugiyama

Zhi-Hua Zhou

NANJING UNIVERSITY

Learning And Mining from DatA

東京大学 THE UNIVERSITY OF TOKYO

RIKEN AIP

ICML International Conference On Machine Learning ◆ VIENNA

TreeLoRA: *Efficient Continual Learning* via Layer-Wise LoRAs Guided by a Hierarchical Gradient-Similarity Tree

Yu-Yang Qian [1 2] **Yuan-Ze Xu** [1 2] **Zhen-Yu Zhang** [3] **Peng Zhao** [1 2] **Zhi-Hua Zhou** [1 2]

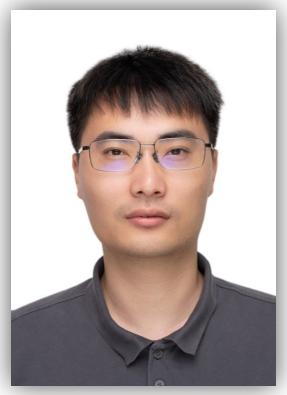
[1] National Key Laboratory for Novel Software Technology, Nanjing University, China
[2] School of Artificial Intelligence, Nanjing University, China
[3] RIKEN AIP, Tokyo, Japan

[ICML'25]

Yu-Yang Qian

Yuan-Ze Xu

Zhen-Yu Zhang



Peng Zhao

Zhi-Hua Zhou

NANJING UNIVERSITY

Learning And Mining from DatA

RIKEN AIP

ICML
International Conference
On Machine Learning
VANCOUVER

Provably Efficient Online RLHF with *One-Pass* Reward Modeling

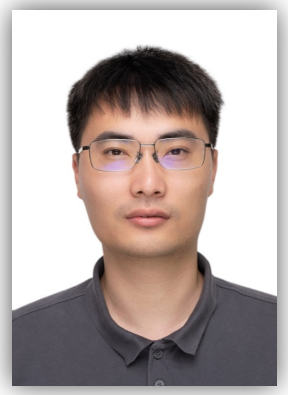
Long-Fei Li*, Yu-Yang Qian*, Peng Zhao, Zhi-Hua Zhou

[NeurIPS'25]

National Key Laboratory for Novel Software Technology, Nanjing University, China
School of Artificial Intelligence, Nanjing University, China,
{lilf, qianyy, zhaop, zhouzh}@lamda.nju.edu.cn

Long-Fei Li*

Yu-Yang Qian*



Peng Zhao

Zhi-Hua Zhou

NANJING UNIVERSITY

Learning And Mining from DatA

NEURAL INFORMATION PROCESSING SYSTEMS SAN DIEGO

Summary

- We investigate the problem of "How to ***efficiently*** adapting to ***online distribution shift***?"
  - [ICML'24]: For convex (exp-concave) cases, ***Detect-restart framework***
  - [ICML'25]: Towards LLMs, ***Efficient continual learning by tree structure***
  - [NeurIPS'25]: Explore ***Online RLHF*** (one-pass update) method for LLMs
- Key techniques: ***Detecting*** & ***Ensemble*** to handle non-stationarity
- Future work: Lower bound of complexity? Other applications? Efficient inference/reasoning?...

Thanks!

References

- Qian Y.-Y., Zhao P., Zhang Y.-J., Sugiyama M., and Zhou Z.-H. Efficient non-stationary online learning by wavelets with applications to online distribution shift adaptation. In Proceedings of the 41st International Conference on Machine Learning (**ICML**), pages 41383-41415, 2024.

- Qian Y.-Y., Wang. Y.-H., Zhang Z.-Y., Jiang Y., and Zhou Z.-H. Adapting to Generalized Online Label Shift by Invariant Representation Learning. In Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (**KDD**), pages 1161-1172, 2025.

- Qian Y.-Y., Xu Y.-Z., Zhang Z.-Y., Zhao P., and Zhou Z.-H. TreeLoRA: Efficient Continual Learning via Layer-Wise LoRAs Guided by a Hierarchical Gradient-Similarity Tree. In Proceedings of the 42nd International Conference on Machine Learning (**ICML**), pages to appear, 2025.

References

- Zachary C. Lipton, Yu-Xiang Wang, and Alexander J. Smola. Detecting and correcting for label shift with black box predictors. In Proceedings of the 35th International Conference on Machine Learning (**ICML**), pages 3128–3136, 2018.

- Dheeraj Baby and Yu-Xiang Wang. Online Forecasting of Total-Variation-bounded Sequences. In: Advances in Neural Information Processing Systems 32 (**NeurIPS**), pages 11069-11079, 2019.

- Ruihan Wu, Chuan Guo, Yi Su, and Kilian Q. Weinberger. Online adaptation to label distribution shift. In Advances in Neural Information Processing Systems 34 (**NeurIPS**), pages 11340–11351, 2021.

References

Yong Bai*, Yu-Jie Zhang*, Peng Zhao, Masashi Sugiyama, and Zhi-Hua Zhou. Adapting to Online Label Shift with Provable Guarantees. In: Advances in Neural Information Processing Systems 35 (**NeurIPS**), page 29960-29974, 2022.

Dheeraj Baby, Saurabh Garg, Thomson Yen, Sivaraman Balakrishnan, Zachary C. Lipton, and Yu-Xiang Wang. Online Label Shift: Optimal Dynamic Regret meets Practical Algorithms. In Advances in Neural Information Processing Systems 36 (**NeurIPS**), pages 65703–65742, 2023.

Peng Zhao, Yu-Jie Zhang, Lijun Zhang, and Zhi-Hua Zhou. Adaptivity and Non-stationarity: Problem-dependent Dynamic Regret for Online Convex Optimization, Journal of Machine Learning Research (**JMLR**), 25(98):1–52, 2024.

Discussion

LAMDA
Learning And Mining from DatA

Online Ensemble	*Wavelet Restart*
• Maintain $\mathcal{O}(\log T)$ base models	• Maintain $\mathcal{O}(\log T)$ wavelet coeffs
• Ensemble multiple models	• Ensemble multiple wavelet bases
• Hedge to handle uncertainty.	• Restart to handle uncertainty.

⇨ Future Work: whether the computational overhead of $\Omega(\log T)$ is *necessary* to handle non-stationary uncertainty?