UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

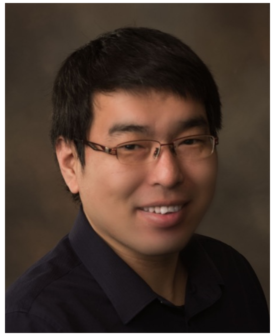
NSF

AIFARMS
Artificial Intelligence for Future Agricultural Resilience, Management, and Sustainability

IBM

Adaptive Test-Time Personalization for Federated Learning






Wenxuan Bao* **Tianxin Wei*** **Haohan Wang** **Jingrui He**

University of Illinois Urbana-Champaign

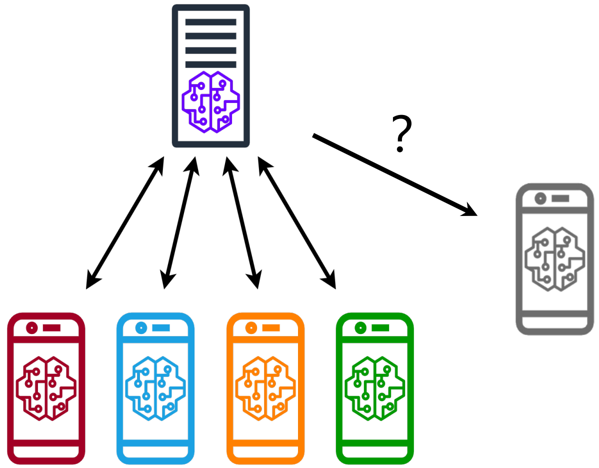
{wbao4,twei10,haohanw,jingrui}@illinois.edu

baowenxuan.github.io, weitianxin.github.io

haohanwang.github.io, hejingrui.org

Cross-Device Federated Learning

- **Federated Learning** (FL): Multiple clients collaborate to train a machine learning model under the orchestration of a central server, without sharing their raw data [1].
- **Cross-Device FL**: The clients are a very large number of mobile/IoT devices.
  - Only a small part of clients (a.k.a. *source clients*) are sampled for training.
  - However, the model also needs to be deployed on clients that do not participate in FL training (a.k.a. *target clients*).
  - Clients typically have their own distributions with *distribution shifts*, e.g., feature shift, label shift.

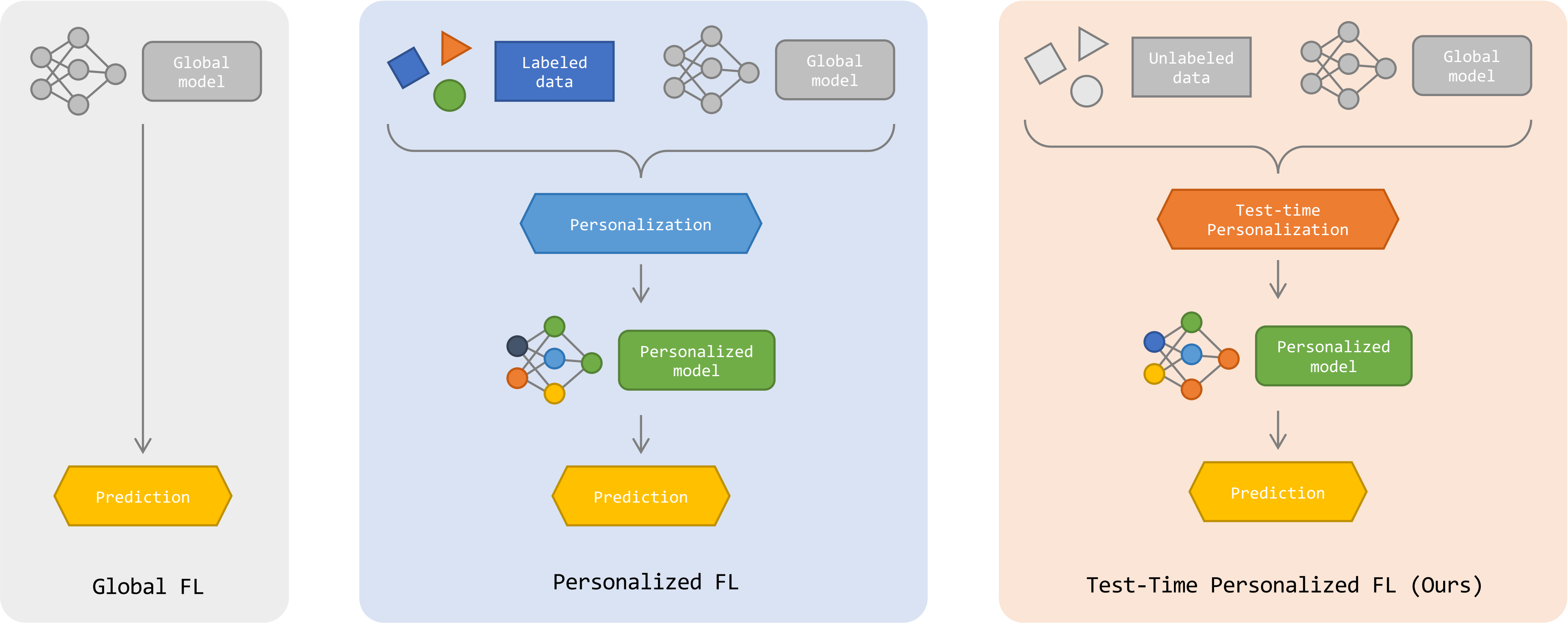


- **Question**: *How to generalize to unparticipating clients under distribution shifts?*

[1] Peter Hairouz, H. Brendan Mcmahan et al. Advances and Open Problems in Federated Learning. Found. Trends Mach. Learn. 14(1-2): 1-210 (2021)
Image source: https://en.wikipedia.org/wiki/Federated_learning

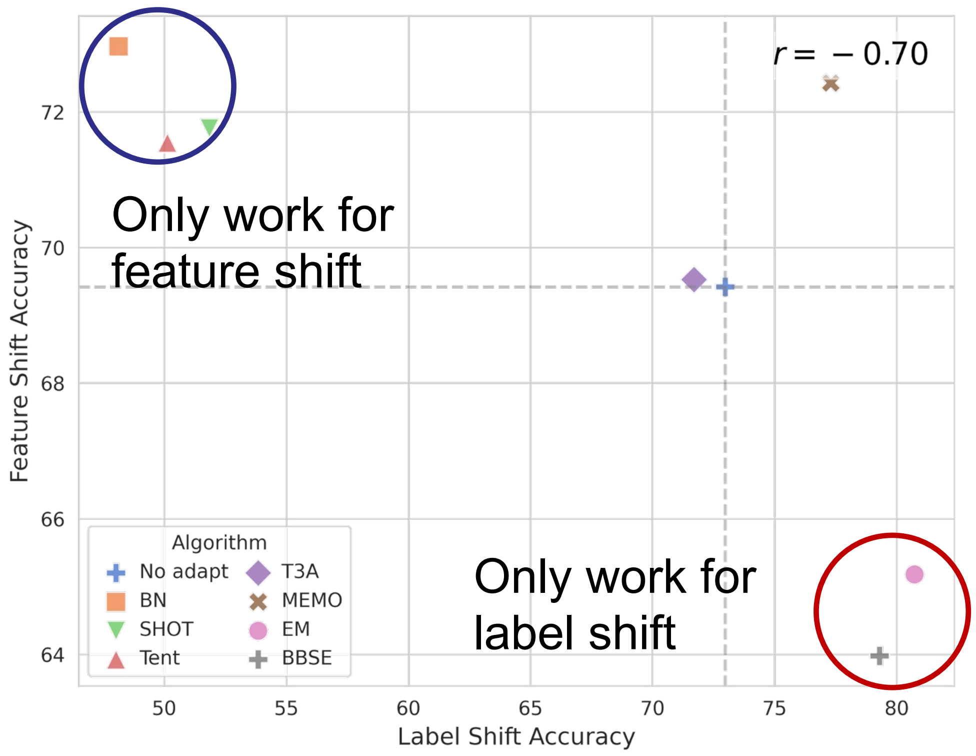
Generalization to Target Clients

	Global FL	Personalized FL	Test-Time Personalized FL
Adaptation to each client	No ☹	Yes 😊	Yes 😊
Data requirement	No 😊	Additional labeled data ☹	Unlabeled testing data 😊



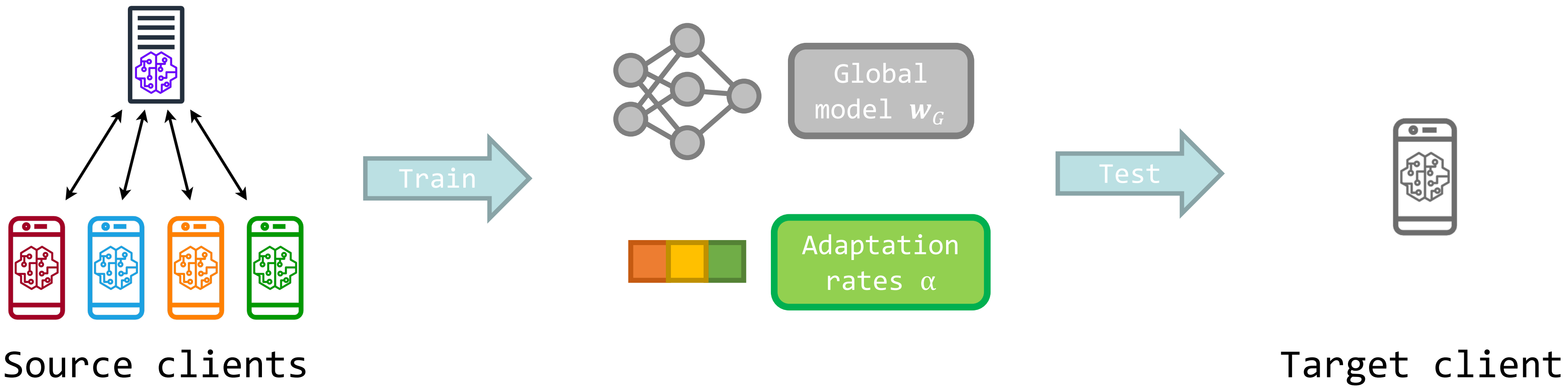
Drawbacks of Current Methods

- **Test-Time Adaptation** (TTA) methods can be applied to TTPFL.
- *Drawback 1*: TTA assumes single source domain and neglects the interrelationship among source clients.
- *Drawback 2*: Most TTA methods are customized for specific distribution shifts and lack the flexibility to address diverse types of distribution shifts in FL.
  - The inflexibility largely results from their **predefined selection of modules to adapt.**

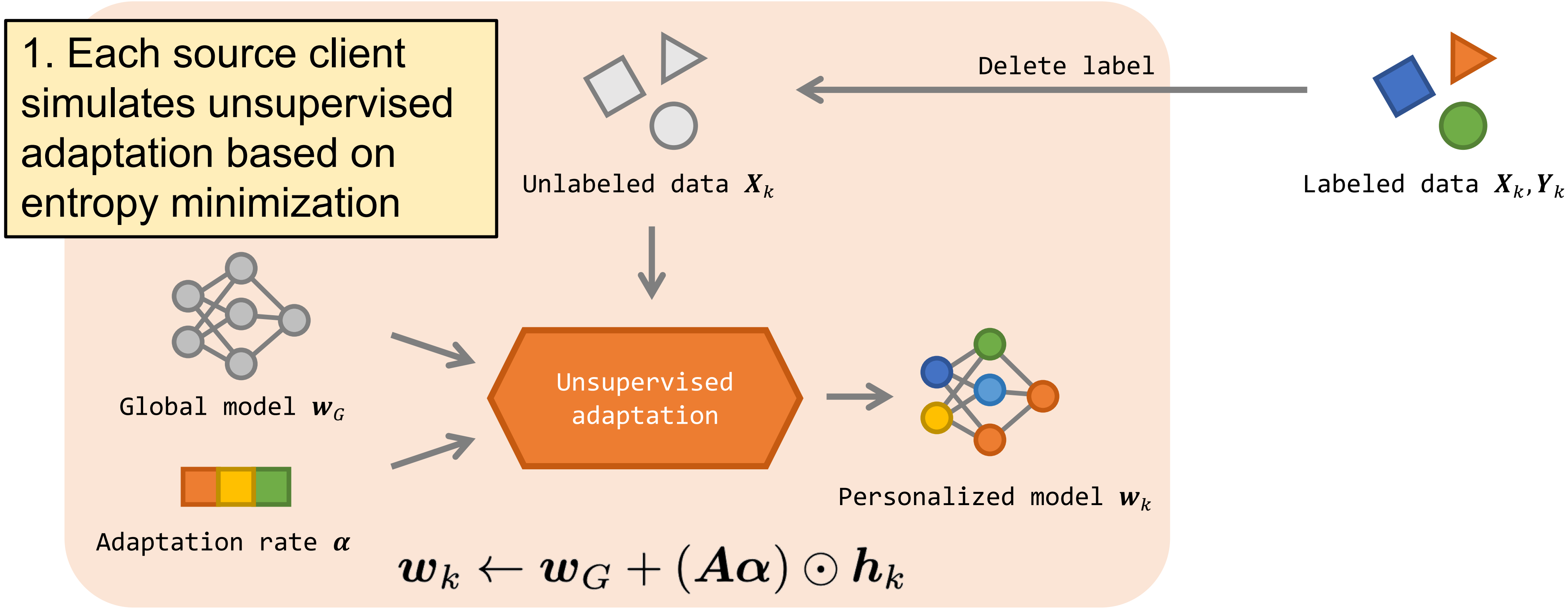


Adaptive Test-Time Personalization

- **Motivation**: *Which modules to adapt should depend on the type of distribution shifts among clients, which can be inferred from source clients.*
- We propose **Adaptive Test-Time Personalization** (ATP) to learn the **adaptation rates** for each module.
  - Modules with larger adaptation rates are adapted to a greater extend, and vice versa.

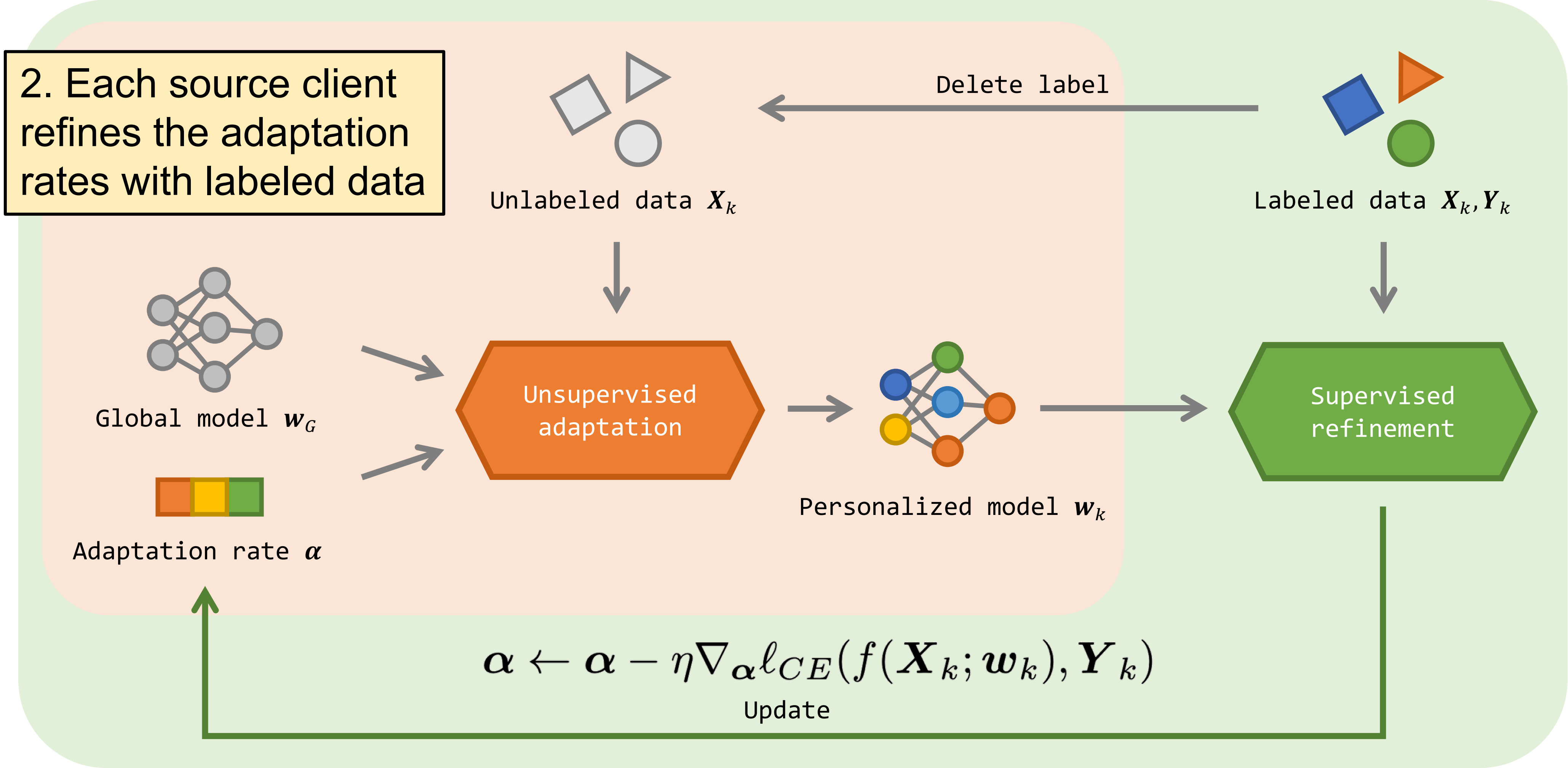


Adaptive Test-Time Personalization



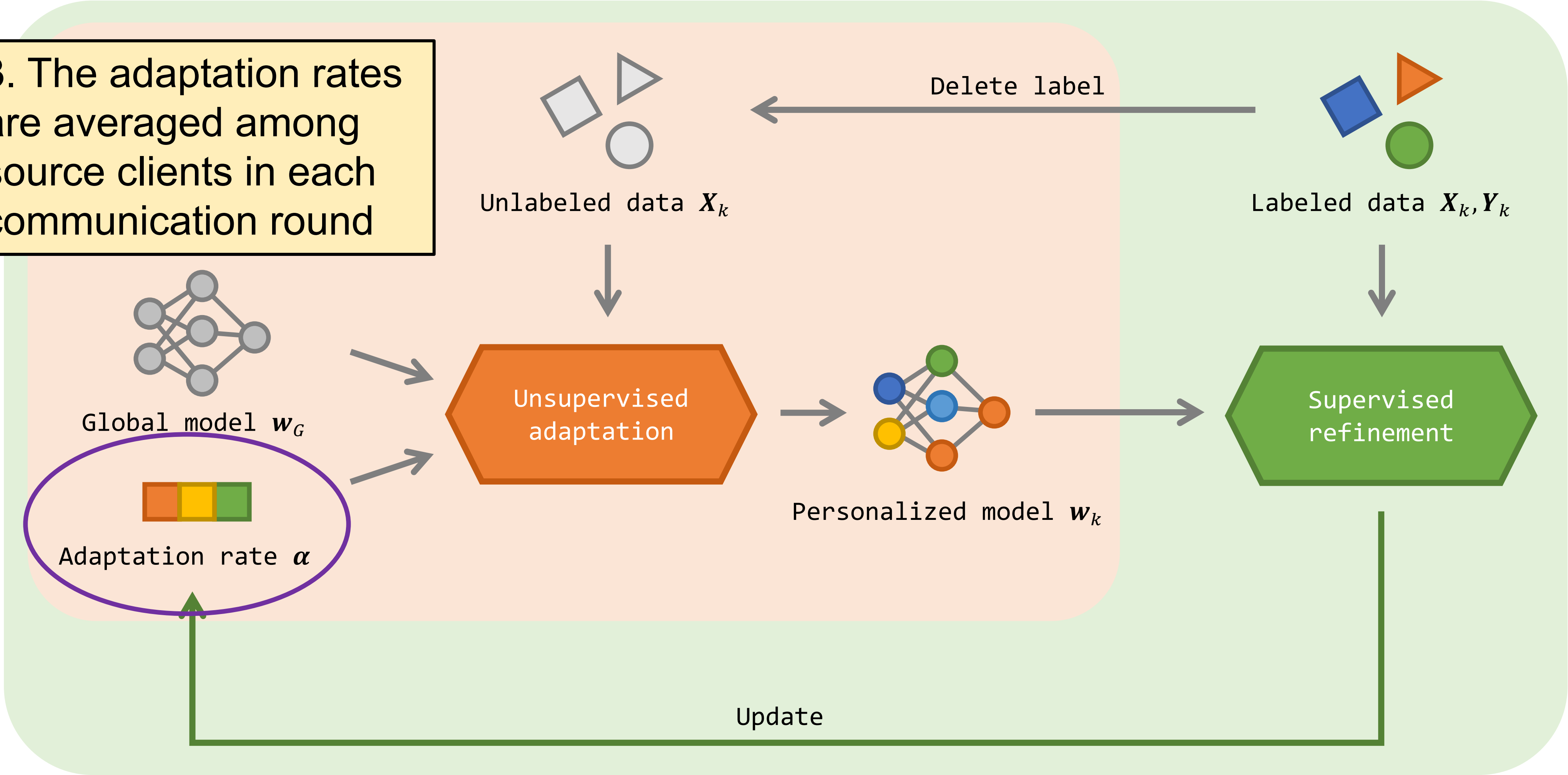
$\boldsymbol{h}_k$ is the update direction of unsupervised adaptation,
$\boldsymbol{A}$ maps each adaptation rate to the model parameters

Adaptive Test-Time Personalization



Adaptive Test-Time Personalization

3. The adaptation rates are averaged among source clients in each communication round



Generalization Guarantee

Theorem 5.1 (Generalization for hypothesis space). Let $\mathcal{H} = \{\boldsymbol{\alpha} : \|\boldsymbol{\alpha}\|_2 \leq R\}$ be the hypothesis space (space of adaptation rates), N be the number of source clients, and K be the number of data batches on each source client. Assuming (1) L-Lipschitz model, and (2) H-module-wise-bounded update direction. For any fixed global model $\boldsymbol{w}_G$ and any $\epsilon > 0$, we have

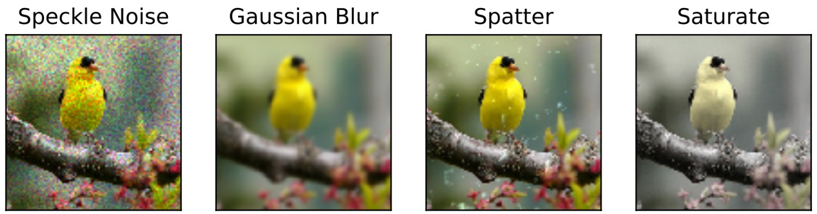
$$\Pr\left(\sup_{\boldsymbol{\alpha}\in\mathcal{H}} |\varepsilon(\boldsymbol{\alpha}) - \hat{\varepsilon}(\boldsymbol{\alpha})| \geq \epsilon\right) \leq \left(\frac{12LHR}{\epsilon}\right)^d \cdot 4\exp\left(-\frac{NK\epsilon^2}{2(\sqrt{K}+1)^2}\right)$$

where $\hat{\varepsilon}(\boldsymbol{\alpha})$ is the average **post-adaptation** error rate on source clients, and $\varepsilon(\boldsymbol{\alpha})$ is the expected **post-adaptation** error rate on clients' population.

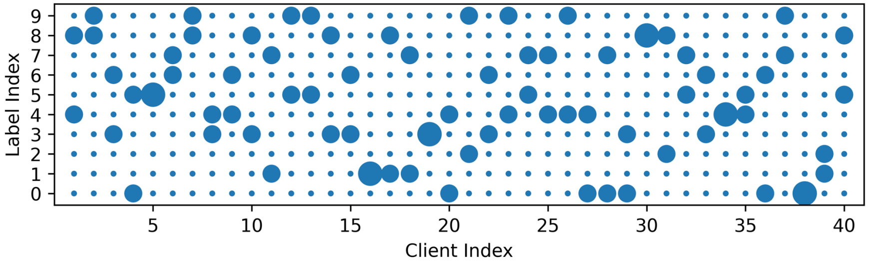
- **Finding 1**: Generalization benefits from low dimensionality of adaptation rates d
- **Finding 2**: Generalization benefits from utilizing multiple sources.
  The bound gets loose if merging N source domains with K samples into one domain with NK samples $(N, K) \leftarrow (1, NK)$

ATP Can Handle Different Distribution Shifts

- CIFAR-10(C) experiment
  - Feature shift: Each client has a random type of image corruption [1].



  - Label shift: Each client has 2 majority classes and 8 minority classes.



  - Hybrid shift: Feature + label shift.

- ATP consistently improves the performance across different types of distribution shifts.

Accuracy over target clients (mean ± s.d.)

Method	Feature shift	Label shift	Hybrid shift	Avg. Rank
No adaptation	69.42 ± 0.13	72.98 ± 0.24	63.68 ± 0.24	7.7
BN-Adapt	73.52 ± 0.22	54.54 ± 0.10	50.42 ± 0.39	7.0
SHOT	71.76 ± 0.17	48.13 ± 0.18	44.68 ± 0.32	9.3
Tent	71.76 ± 0.09	50.13 ± 0.21	46.05 ± 0.26	8.3
T3A	69.53 ± 0.08	71.70 ± 0.32	62.17 ± 0.17	8.0
MEMO	72.43 ± 0.22	77.30 ± 0.15	68.07 ± 0.28	4.3
EM	65.18 ± 0.12	80.73 ± 0.18	69.85 ± 0.43	5.0
BBSE	63.98 ± 0.17	79.30 ± 0.17	67.96 ± 0.43	6.7
Surgical	69.85 ± 0.22	76.00 ± 0.17	66.94 ± 0.43	6.3
ATP-batch	73.68 ± 0.10	79.90 ± 0.22	73.05 ± 0.35	2.3
ATP-online	$\mathbf{74.06 \pm 0.18}$	$\mathbf{81.96 \pm 0.14}$	$\mathbf{75.37 \pm 0.22}$	**1.0**

(We also conduct experiments on Digits-5 and PACS.)

[1] Dan Hendrycks, Thomas G. Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. ICLR 2019.

ATP Learns Shift-Specific Adaptation Rates

- We train and test adaptation rates with different types of distribution shift.
  - ATP performs the best when training and testing under the same type of distribution shift.

Accuracy over target clients (mean ± s.d.)

Train	Test		
	Feature shift	Label shift	Hybrid shift
No adaptation	69.42 ± 0.13	72.98 ± 0.24	63.68 ± 0.24
Feature shift	$\mathbf{73.68 \pm 0.10}$	65.05 ± 1.82	60.64 ± 1.43
Label shift	67.99 ± 0.28	$\mathbf{79.90 \pm 0.22}$	69.50 ± 0.52
Hybrid shift	72.69 ± 0.14	78.92 ± 0.34	$\mathbf{73.05 \pm 0.35}$

ATP Learns Shift-Specific Adaptation Rates

- We train and test adaptation rates with different types of distribution shift.
  - ATP performs the best when training and testing under the same type of distribution shift.
  - The adaptation rates trained under feature shifts have negative impact on label shifts, and vice versa.

Accuracy over target clients (mean ± s.d.)

Train	Test		
	Feature shift	Label shift	Hybrid shift
No adaptation	69.42 ± 0.13	72.98 ± 0.24	63.68 ± 0.24
Feature shift	**73.68 ± 0.10**	65.05 ± 1.82	60.64 ± 1.43
Label shift	67.99 ± 0.28	**79.90 ± 0.22**	69.50 ± 0.52
Hybrid shift	72.69 ± 0.14	78.92 ± 0.34	**73.05 ± 0.35**

ATP Learns Shift-Specific Adaptation Rates

- We train and test adaptation rates with different types of distribution shift.
  - ATP performs the best when training and testing under the same type of distribution shift.
  - The adaptation rates trained under feature shifts have negative impact on label shifts, and vice versa.
  - The adaptation rates trained under hybrid shift are also beneficial for feature and label shifts.

Accuracy over target clients (mean ± s.d.)

Train	Test: Feature shift	Test: Label shift	Test: Hybrid shift
No adaptation	69.42 ± 0.13	72.98 ± 0.24	63.68 ± 0.24
Feature shift	$\mathbf{73.68 \pm 0.10}$	65.05 ± 1.82	60.64 ± 1.43
Label shift	67.99 ± 0.28	$\mathbf{79.90 \pm 0.22}$	69.50 ± 0.52
Hybrid shift	72.69 ± 0.14	78.92 ± 0.34	$\mathbf{73.05 \pm 0.35}$

Key Takeaways

- **TTPFL framework**: It is important and feasible to personalize a model on novel unlabeled clients in cross-device federated learning.
- **ATP algorithm**: Which modules to adapt should depend on the type of distribution shifts among clients, which can be inferred from source clients.

Paper

Code

Personal