POSTER: FLOW – FINE-GRAINED PERSONALIZED FEDERATED LEARNING THROUGH DYNAMIC ROUTING

Kunjal Panchal [1] **Hui Guan** [1]

ABSTRACT

Personalization in Federated Learning (FL) has been proven effective for incentivizing clients to participate in the training. However, personalization has been only studied at a coarse granularity where all the input instances of a client (heterogeneous or otherwise) only use its individual local model, despite it being limited to only that client's data. *Flow* explores instance-level personalization through dynamically making routing decisions between the local and the global model, with the aim of achieving superior personalized performance for a given instance. Besides, as cross-device FL deals with millions of resource-constrained client devices, we push towards stateless personalization where a client doesn't need to carry its personalized state across FL rounds.

1 INTRODUCTION

Federated Learning (FL) allows resource-constrained edge devices (called *clients*) to collaboratively train a global machine learning model while locally keeping the training data. Due to the heterogeneity in clients' data distribution, the global model could perform worse than purely locally-trained model for some clients. *Personalization* is an effective approach to incentivize these clients to participate in FL by offering them improved prediction accuracy (Tan et al., 2022; Fallah et al., 2020; Zhang et al., 2021).

Existing personalization approaches, however, fall short on addressing two remarkable properties of FL. First, they personalize at *client level* (Li et al., 2020), which is coarse-grained; here, all instances on a client utilizes the client's personalized model for prediction, with the same execution path. However, instances that fall well under the global data distribution can benefit more by using a global model, which has been trained on larger and more heterogeneous data. Second, many of these approaches are *stateful*, which require each client to carry its personalized state across FL rounds (Deng et al., 2020; Tan et al., 2022). A client has to participate frequently in stateful personalization in order to avoid its personalized model from turning stale and thus less accurate (Deng et al., 2020). However, it is impractical due to edge devices' resource constraints and also fails to scale to a large number of clients.

This work addresses the above shortcomings by proposing *Flow*, a fine-grained and stateless personalized federated learning framework. *Flow* creates a personalized model that can dynamically make decisions on when to use a client's local parameters and when to use the global parameters, depending on the input it receives. By "fine-grained", we mean that every input instance on a client can pick an instance-specific execution path to improve its prediction accuracy. It allows instances which fall well under the global data distribution to use the global model route, while the other instances would use the local route for better feature representations. "Stateless" implies that no client needs to persist any personalized states across FL rounds. Instead, at the beginning of each FL round, personalized model states on a participating client are created based on the global model. It allows FL to easily scale with number of clients.

This paper makes the following contributions: (1) This is the first work on instance-level personalization in FL. (2) The use of dynamic routing during joint-optimization strikes a balance between generalizability of the global model and utility of the local model. (3) Evaluation on language tasks demonstrate *Flow* gives competitive accuracy against state-of-the-art per-client personalization approaches.

2 OUR APPROACH

In each round of *Flow*, participating clients follow two major steps to collaboratively train a global model, and to personalize it. (1) After receiving the global model from the server, each participating client first derives *local parameters* by finetuning its global model counterpart on the client's local data. (2) Next, the client creates a *personalized model* which consists of the global parameters, local parameters, and a routing policy component. It trains the routing policy component, as well as the global model parameters.

[1] University of Massachusetts, Amherst. Correspondence to: Kunjal Panchal <kpanchal@umass.edu>.

The global model parameter updates from each client will be aggregated by the server for the next FL round. Now we elaborate on these two steps:

Client-level Personalization Assume the global model $\boldsymbol{w}^g$ consists of three sets of parameters: embedding parameters $\boldsymbol{w}^g_{\mathsf{emb}}$, encoder parameters $\boldsymbol{w}^g_{\mathsf{enc}}$, and decoder parameters $\boldsymbol{w}^g_{\mathsf{dec}}$. $\boldsymbol{w}^g := \{\boldsymbol{w}^g_{\mathsf{emb}}, \boldsymbol{w}^g_{\mathsf{enc}}, \boldsymbol{w}^g_{\mathsf{dec}}\}$. A client derives local parameters $\boldsymbol{w}^\ell := \{\boldsymbol{w}^\ell_{\mathsf{enc}}\}$ by finetuning $\boldsymbol{w}^g_{\mathsf{enc}}$ on the client's local data for a single epoch. We kept the finetuning to one epoch since an edge device has limited computation capacity, and this also avoids overfitting for the clients with small number of training samples.

Instance-level Personalization The personalized model $\boldsymbol{w}^p$ consists of the global model $\boldsymbol{w}^g$, the local model $\boldsymbol{w}^\ell$, and the routing policy parameters $\boldsymbol{\theta}$. Instance-level personalization trains $\boldsymbol{w}^p$ on each client to maximize its prediction accuracy.

Since the shallow sequential models are computationally less intensive, and still allow dynamic routing temporally, our approach focuses on RNN-based language models. At a timestep t, the routing policy outputs the probability of route choices between the local and the global model parameters, $\mathbf{r}_t = \text{softmax}_\tau(f_{\theta_t}([x_t; \mathbf{h}_{(t-1)}]))$, where $f_{\theta_t} : \mathbb{R}^{n+d} \to \mathbb{R}^2$ is the routing policy function based on the policy parameters $\theta_t \in \boldsymbol{\theta}$, considering the resource constraints of a client, we have used only one fully-connected layer as f. τ is *temperature* hyperparameter. We have set $\tau = 0.75$. $x_t \in \mathbb{R}^n$ is the input instance at timestep t, and $\mathbf{h}_{(t-1)} \in \mathbb{R}^d$ is the hidden state of previous timestep. $\mathbf{h}_t$ is derived using encoders of $\boldsymbol{w}^g$ and $\boldsymbol{w}^\ell$ as follows, $\mathbf{h}_t = [g_{\boldsymbol{w}^g_{\mathsf{enc}}}(\mathbf{h}_{t-1}, \mathbf{x}_t); \ell_{\boldsymbol{w}^\ell_{\mathsf{enc}}}(\mathbf{h}_{t-1}, \mathbf{x}_t)] \cdot \mathbf{r}_t$. Here, g and ℓ are the global model and local model encoders respectively. The goal is to use the global encoder if it predicts output for $\mathbf{x}_t$ with higher confidence than the local encoder.

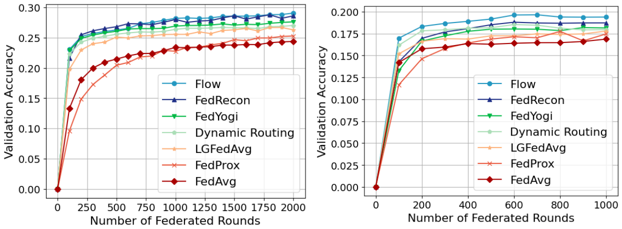
Inference Since a client doesn't persist any local states, it needs to create $\boldsymbol{w}^p$ for inference with two steps. The client (1) pulls the global model $\boldsymbol{w}^g$ and generates $\boldsymbol{w}^\ell$, and (2) trains the routing policy $\boldsymbol{\theta}$ with $\boldsymbol{w}^g$ and $\boldsymbol{w}^\ell$ being frozen. Updating $\boldsymbol{w}^\ell$ and $\boldsymbol{\theta}$ before inference allows clients to benefit from the latest feature representation learned by the global model, improving inference performance of instances favors global model parameters instead of local model parameters.

3 Evaluation

To validate the effectiveness of our approach, we perform next word prediction tasks on Stackoverflow, and Reddit datasets. We compared *Flow* against the following baselines: FedAvg and FedYogi which do not personalize for any client, Ditto and APFL which are stateful personalization methods, FedProx, FedRecon, and LG-FedAvg which are per-client stateless personalization methods. We also compare *Flow* with "dynamic routing" where the global model itself is a dynamic network and clients finetune it to create personalized models. Figure 1 and Table 1 report the validation and test accuracies. For Stackoverflow, dynamic routing outperforms stateful per-client approaches, and for Reddit, dynamic routing almost performs as well as the same stateful per-client approaches. This provides incentive for integrating dynamic routing with per-client personalization as in *Flow*. Overall, *Flow* outperforms all the baselines thanks to finer-grained personalization.

Table 1. Test accuracies for Stackoverflow and Reddit. ◇ = Server-side Optimization, † = Stateful Personalization, ‡ = Stateless Personalization, § = Per-Client Personalization, ♭ = Per-Instance Personalization.

Methods	Stackoverflow	Reddit
FedAvg ◇	24.98%	17.30%
FedYogi ◇	27.90%	18.85%
APFL † §	25.48%	18.48%
Ditto † §	25.55%	18.74%
FedProx ‡ §	25.38%	17.58%
LG-FedAvg ‡ §	26.78%	18.59%
FedRecon (500 OOV) ‡ §	28.26%	18.89%
Dynamic Routing ‡ § ♭	26.54%	18.70%
Flow (Ours) ‡ § ♭	**29.16%**	**19.36%**



(a) Stackoverflow (b) Reddit

Figure 1. Average validation performance across last 100 rounds for *Flow* and baselines.

References

Deng, Y., Kamani, M. M., and Mahdavi, M. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020.

Fallah, A., Mokhtari, A., and Ozdaglar, A. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. In *Advances in Neural Information Processing Systems*, 2020.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems*, 2020.

Tan, A. Z., Yu, H., Cui, L., and Yang, Q. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. doi: 10.1109/TNNLS.2022.3160699.

Zhang, M., Sapra, K., Fidler, S., Yeung, S., and Alvarez, J. M. Personalized federated learning with first order model optimization. In *International Conference on Learning Representations*, 2021.