On the Generalization and Personalization Tradeoff for Agentic AI Networks

Xubo Li*, Yong Xiao*§¶, Yingyu Li‡
*School of Elect. Inform. & Commun., Huazhong Univ. of Science & Technology, China
‡School of Mech. Eng. & Elect. Inform., China Univ. of Geosciences, Wuhan, China
§Peng Cheng Laboratory, Shenzhen, China
¶Pazhou Laboratory (Huangpu), Guangzhou, China

Abstract*—Agentic AI networking (AgentNet) has attracted significant interest recently due to its promising potential in supporting proactive learning and seamless collaboration among distributed task-oriented agents in various environments. However, existing solutions face inherent dilemmas. On the one hand, developing a single globalized model that can generalize well for diverse agents incurs inconsistent and unreliable performance due to the neglect of their distinct deployment environments. On the other hand, constructing personalized models that are tailored according to the individual needs of each agent often suffers from resource inefficiency and under-utilization of shared knowledge among agents. To overcome these challenges, we propose MAN, a novel Meta learning-based AgentNet architecture that optimally balances generalization and personalization for all the agents when performing different tasks in dynamic environments. Specifically, MAN adopts a bi-level optimization framework to develop foundation meta-models as the shared initialization of all agents, and each agent can then fine-tune the meta model to enable personalized deployment. We derive theoretical bounds on both global generalization and local personalization errors, demonstrating that the fundamental tradeoff between these two can only be optimized but cannot be fully eliminated. Extensive experiments on real-world datasets validate the performance of the proposed MAN and confirm the generality of our theoretical insights.

***Index Terms*—AgentNet, generalization, personalization.**

I. Introduction

With the increasing proliferation of AI-based solutions in various parts of telecommunication networks, the next generation mobile systems, especially 6G and beyond, will be largely defined by a complex ecosystem consisting of diverse AI-based functions, models, and applications [1]. However, this shift presents a significant challenge to the existing communication networking architectures. In particular, it is known that the continuous training and real-time inference of AI models will generate substantial data traffic and introduce computational demands, exceeding the capacity of the already congested networking infrastructure. Also, existing AI models are passive learning frameworks that are trained based on static and pre-collected datasets and cannot be directly applied in tasks and scenarios that have different and unseen features compared to the training dataset. It is also uneconomic and unfeasible to develop an individually isolated solution for every unique task and scenario. There is an imperative to develop a unified, AI-native networking architecture that is designed to support the efficient communication, coordination, and reuse of various AI models, functional modules, and knowledge [2].

Motivated by the above observation, the agentic AI networking (AgentNet) has attracted significant interest from both industry and academia due to its promising solutions to address the limitations of existing AI-based solutions, and enable autonomous and goal-oriented learning[3]. Different from the traditional AI-based networking systems, in which individual AI models are trained for singular, pre-defined tasks—such as chatbots for customer service or traffic prediction—an AgentNet systems is comprised of various types of autonomous AI agents that are capable of proactively recognizing and tracking the requirements of the users or systems, breaking down complex tasks, making independent decisions, and executing multi-step plans and management without constant oversight of human operators[4]. More specifically, in AgentNet, all the agents do not operate in isolation, but interact and cooperate with each other to achieve a common and network-wide objective. They can share data, negotiate resources, and jointly solve problems that cannot be handled by a single agent, leading to a new self-learning networking framework that can dynamically adapt to changing conditions, predict and prevent failures, and offer a more resilient and efficient networking service[5]. Despite still being in the early stage of development, considerable efforts have been directed toward investigating various parts of AgentNet. More specifically, in [3], we have defined the basic concepts of AgentNet including its unique features, architectural framework and key performance metrics. A generative foundation model-based implementation framework has been proposed and two application scenarios, digital twins-based industrial automation and metaverse-based infotainment system have been presented. In [6], the authors explored the potential of solving complex tasks by leveraging multi-agent collaboration.

One of the key features of AgentNet is that the knowledge and models learned by some agents can be reused by or transferred to other agents for similar or even different tasks. As discussed in [3], generalization and personalization are two critical metrics for AgentNet because they collectively determine the system's overall adaptability and user-centric effectiveness. More specifically, generalization capability ensures each agent can apply previously learned knowledge or skills across diverse, unseen tasks, environments, and personalization capability allows the agent to tailor its behavior, responses, or decision-making to individual users' personal preferences, histories, or unique needs, fostering relevance, trust, and long-term engagement. Together, these metrics ensure AgentNet system is both robust across varied scenarios and deeply aligned with human users' personal needs, making it practical for real-world deployment.

Unfortunately, simultaneously achieving both broad generalization and deep personalization is a significant challenge in existing AI model-based networking frameworks because these two goals are often at odds. In particular, improving generalization performance requires an AI model to learn common patterns and features from a wide, diverse dataset so that it can perform well on unseen data across a wide range of tasks and use scenarios [7]. This process may smooth out individual variations to train a universal model. In contrast, enhancing the personalization capability requires that a model focus on the unique preferences and personalized behaviors of a single user or a small group [8]. This means learning a specific, often idiosyncratic, pattern that may not apply to anyone else. Trying to optimize for one can negatively impact the other. For example, a model trained to be able to generalize across diverse tasks and scenarios might miss subtle individual cues needed for effective personalization, while a highly personalized model might overfit to one user's data and perform poorly when applied to new users and tasks. The tension lies in the fundamental trade-off between creating a model that is universally applicable and one that is uniquely tailored.

To address the above problems, we propose MAN, a meta-learning-based framework that strikes an effective balance between global generalization and local personalization in AgentNet. Specifically, MAN adopts a bi-level optimization framework: *(i) cross-agent meta-optimization* and *(ii) agent-specific adaptation*. In the meta-optimization stage, an agent controller is introduced to periodically aggregate common knowledge from diverse agents operating across different environments to construct shared meta-models. Then, in the agent-specific adaptation stage, each agent fine-tunes the meta-model to adapt to its environment based on local environmental observations. Our propose architecture allows newly added agents to take advantage of the already trained meta-models to accelerate their adaptation in new unseen environments. We derive theoretical bounds for both generalization and personalization errors and prove the superiority of MAN in balancing their tradeoff. Extensive experiments are conducted based on real-world datasets. Our experimental results demonstrate that proposed MAN can improve the tradeoff between personalization and generalization in various environments and application scenarios.

II. System Model and Problem Formulation

We consider an AgentNet system consisting of a set $\mathcal{K}$ of K agents deployed in different environments, and can be assigned a specific task. Let $\mathcal{M}$ be the set of all tasks. Suppose each agent is deployed in a specific environment characterized by an environmental state, defined as a set of environmental variables that affect the task performance of the agent. We assume that each agent can only be deployed in one environment and assigned one task. We can therefore abuse the notation and use e_k to denote the state of the environment of the kth agent.

We consider a task-driven model adaptation and fine-tuning problem for agents deployed in time-varying environments. In this case, the environment of each agent can change with time and we assume the environment evolution process can be slotted, and in each time slot, the environment state can be assumed to be fixed. We use the subscript t to denote the environmental state in the tth time slot. A set of foundation models can be trained and distributed to all the agents to perform their assigned tasks. Let w_0^m be the foundation model pre-trained for task m. These models can be updated and fine-tuned by each agent according to the dynamics of its environment. The basic idea is to allow all the agents to jointly construct and maintain the foundational models to improve the generalization capability, and also each agent can fine-tune the model according to its local dataset, with the main objective to enhance the personalization performance. Let $\mathcal{D}_{k,t}$ be the dataset observed by agent k in time slot t. In this case, we assume an agent controller can be deployed to keep track of the generalization performance of all the agents, and each agent will keep track of its personalization performance based on its local observation $\mathcal{D}_{k,t}$.

Before we formulate the problem, let us first introduce the following metrics to quantify the generalization and personalization performance of agents.

Suppose each agent k can obtain a foundation model $w_{0,t}^m$ at the beginning of time slot t from the agent controller for task m. We assume each agent k can adopt a personalization solution, denoted as $w_{k,t}^m = g_k(w_{0,t}^m, \mathcal{D}_{k,t})$, to fine-tune the obtained foundation models for task m.

We can then define the generalization error of an agent as the performance gap caused by the fact that the agent's model $w_{k,t}^m$ cannot fully capture the real probability distribution of the environmental state of agent k in time slot t.

Definition 1 (Generalization Error)**.** *We define the generalization error $\mathcal{E}_{gen}$ of all the agents as the overall performance discrepancy between the loss obtained by model $w_{k,t}^m$ trained based on the dataset $\mathcal{D}_{k,t}$, denoted as $\hat{\mathcal{L}}_k^m(w_{k,t}^m, \mathcal{D}_{k,t})$, and that obtained by the ideal model $w_{k,t}^m$ developed based on the ground truth distribution of the environmental state, denoted as $\mathcal{L}^m(w_{k,t}^m)$, given by*

$$\mathcal{E}_{gen} := \frac{1}{K}\sum_{k\in\mathcal{K}} \mathbb{E}_{\mathcal{D}_{k,t}}[\mathcal{L}_k^m(w_k^m) - \hat{\mathcal{L}}_k^m(w_{k,t}^m, \mathcal{D}_{k,t})]. \quad (1)$$

We define the personalization error of each agent k as the difference between the loss obtained by the locally adapted model and that of the locally optimal model.

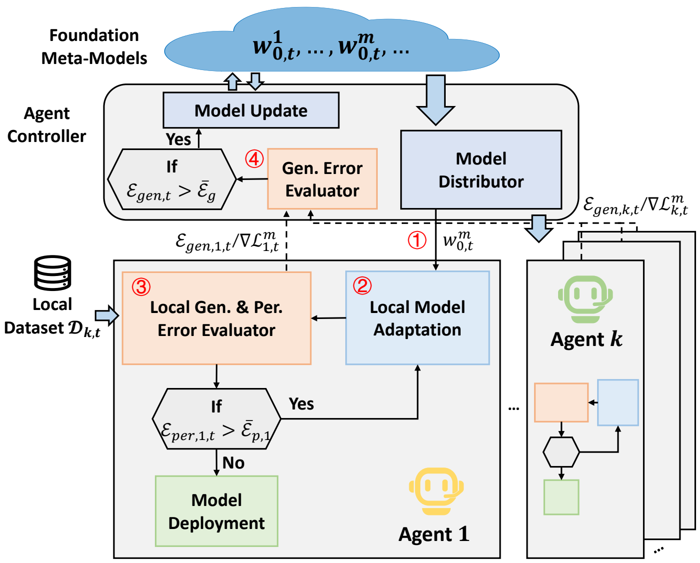


Fig. 1: MAN Architecture

Definition 2 (Personalization Error). *Suppose that for each agent k, there exists a locally optimal model $w_{k,t}^{m*}$. We define the personalization error of agent k when using personalization solution g_k to fine-tune the foundation model $w_{0,t}^m$ as:*

$$\mathcal{E}_{per} := \|\mathcal{L}_k(w_{k,t}^{m*}) - \mathcal{L}_k(g_k(w_{0,t}^m, \mathcal{D}_{k,t}))\|. \tag{2}$$

As mentioned earlier, the main objective is to develop a unified AgentNet architecture for all the agents personalization to construct task-oriented models that minimize their local task-specific losses and at the same time ensure both generalization and personalization errors are below tolerable levels. We can then formulate the optimization problem as follows:

$$\min_{w_{k,t}^m} \ \hat{\mathcal{L}}_k^m(w_{k,t}^m, \mathcal{D}_{k,t}), \quad \forall k \in \mathcal{K} \tag{3}$$

$$\text{s.t.} \quad \mathcal{E}_{gen} \le \bar{\mathcal{E}}_g \ \text{ and } \ \mathcal{E}_{per} \le \bar{\mathcal{E}}_p. \tag{4}$$

where $\bar{\mathcal{E}}_g$ and $\bar{\mathcal{E}}_p$ are the maximum tolerable levels of generalization and personalization errors.

From Definitions 1 and 2, we can observe that the generalization error is evaluated based on all the agents deployed in different environments. The personalization error is evaluated by each agent based on its local dataset. In the rest of this paper, we will propose a meta-learning-based AgentNet framework that allows autonomous meta-model distribution and updating coordinated by the agent controller, as well as local model fine-tuning at individual agents according to the dynamics of the environments.

III. Architecture of MAN

Let us now introduce MAN, a meta-learning-based AgentNet framework, for autonomous task-driven model adaptation and updating. MAN consists of a two-layer training architecture, aiming at constructing a shared meta-initialization that enables rapid adaptation to each agent after it is slightly updated, as illustrated in Fig. 1. We elaborate on the workflow as follows.

Model distribution: First, the agent controller will identify the task m handled by each agent, and distribute the corresponding pre-trained meta-model $w_{0,t}^m$ for it based on its task.

Local model adaptation: Upon receiving the meta-model, agent k will finetune the meta-initialization $w_{0,t}^m$ via E steps of stochastic gradient descent (SGD) to obtain a personalized local model $w_{k,t}^m$. In practice, for computational efficiency, we typically consider the case of a single adaptation stp, i.e., $E = 1$, resulting in:

$$w_{k,t}^m = g_k(w_{0,t}^m, \mathcal{D}_{k,t}) := w_{0,t}^m - \beta \nabla \hat{\mathcal{L}}_k^m(w_{0,t}^m, \mathcal{D}_{k,t}^{in}), \tag{5}$$

where $\mathcal{D}_{k,t}^{in} \subset \mathcal{D}_{k,t}$ denotes the dataset sampled from $\mathcal{D}_{k,t}$ for local adaptation, with size equal to half of $|\mathcal{D}_{k,t}|$. β denotes the local learning rate.

Local personalization error evaluation: Based on the developed adapted model, each agent will evaluate its personalization error locally according to the current local dataset $\mathcal{D}_{k,t}$. Specifically, the personalization error is computed on the other half of dataset $\mathcal{D}_{k,t}^{out} = \mathcal{D}_{k,t} \setminus \mathcal{D}_{k,t}^{in}$ to assess how well the adapted model fits the agent's specific environment. If this error exceeds a pre-defined threshold $\bar{\mathcal{E}}_{per,k}$, the agent triggers a re-adaptation process to refine the local model. Otherwise, the model is deemed sufficiently personalized and is deployed for local utilization.

Generalization error evaluation: As previously discussed, generalization error is evaluated across all the agents deployed in diverse environments. Therefore, each agent will periodically compute its local generalization error based on $\mathcal{D}_{k,t}^{out}$. The evaluation results are then uploaded to the agent controller for global aggregation and assessment. If the generalization error exceeds a predefined threshold $\bar{\mathcal{E}}_{gen}$, the agent controller will request the corresponding agent to upload its model gradient, computed as $\nabla \hat{\mathcal{L}}_k^m(w_{k,t}^m, \mathcal{D}_{k,t}^{m,out})$, to enable meta-model refinement. The meta-optimization step at the controller is then performed via:

$$w_{0,t}^m \leftarrow w_{0,t}^m - \frac{\gamma}{K} \sum\nolimits_{k \in \mathcal{K}} \nabla \hat{\mathcal{L}}_k^m(w_{k,t}^m, \mathcal{D}_{k,t}^{m,out}), \tag{6}$$

where γ denotes the meta-learning rate. This update mechanism ensures that the meta-initialization $w_{0,t}^m$ evolves based on cross-agent feedback, thereby enhancing its ability to generalize across heterogeneous environments.

This two-layer nested optimization structure ensures that the meta-model is updated to achieve better generalization performance across all agents, while the adapted model serves to meet the personalized objective of each agent. By adjusting the relationship between the local learning rate and meta-learning rate, and the number of steps of the inner SGD, a balance between generalization and personalization can be achieved. The detailed workflow are summarized in Algorithm 1.

Algorithm 1: MAN Workflow

Input: Inner layer step size β; Outer layer step size γ.
Output: Meta-model $w_{0,t}^m$; Personalized models for each agent $\langle w_{k,t}^m \rangle_{k\in\mathcal{K}}$.

```
// Each Agent
```
for *agent* $k \in \mathcal{K}$ **do**
  while $\mathcal{E}_{per,k} > \bar{\mathcal{E}}_{per,k}$ **do**
    Sample a subset of data $\mathcal{D}_{k,t}^{in}$;
    Compute adapted model by using (5);
  Periodically evaluate local generalization error by (1);
  Upload $\mathcal{E}_{per,k}$ to agent controller;

```
// Agent Controller
```
Periodically evaluate generalization error;
for $\mathcal{E}_{gen} > \bar{\mathcal{E}}_g$ **do**
  Update meta-model by using (6);

IV. Theoretical Results

This section presents the theoretical analysis of generalization and personalization errors associated with the proposed framework and discusses the fundamental tradeoff between these two key aspects. For notational simplicity, we omit the task-specific superscript m and time-specific subscript t throughout this section.

A. Generalization Error

We derive the generalization bound by establishing its algorithmic uniform stability properties.

Definition 3 (Modified Uniform Stability)**.** *Take the dataset $\tilde{\mathcal{D}}$ which is the same as $\mathcal{D}$, except that for only one $i \in \mathcal{K}$, $\tilde{\mathcal{D}}_i^{in}$ and $\tilde{\mathcal{D}}_i^{out}$ differ from $\mathcal{D}_i^{in}$ and $\mathcal{D}_i^{out}$ in at most B and one data points, respectively. Then, for any $\tilde{x}, \tilde{y}$ and any B distinct state-act pairs $\{(x_1, y_1), \ldots, (x_B, y_B)\}$, the algorithm is called (δ, B)-uniform stability if the following condition holds,*

$$\begin{aligned}\mathbb{E}\big|&\mathcal{L}(w_{\mathcal{D}} - \beta\nabla\hat{L}(w_{\mathcal{D}}, \{(x_j, y_j)\}_{j=1}^B); \tilde{x}, \tilde{y}) \\ &- \mathcal{L}(w_{\tilde{\mathcal{D}}} - \beta\nabla\hat{L}(w_{\tilde{\mathcal{D}}}, \{x_j, y_j\}_{j=1}^B); \tilde{x}, \tilde{y})\big| \le \delta,\end{aligned} \tag{7}$$

where $w_{\mathcal{D}}$ is the output of the algorithm given dataset $\mathcal{D}$.

The main importance of this definition is its connection with generalization error. In particular, it can be proved that if an algorithm is (δ, B)-uniform stability, then the generalization error of its output is bounded above by δ. We list the assumptions used to derive the upper bound for generalization error as follows.

Assumption 1. *For any $x \in \mathcal{X}, y \in \mathcal{Y}$, the loss function $\mathcal{L}(., x, y)$ is twice continuously differentiable. Furthermore, we assume it satisfies the following properties for any $w, w' \in \mathcal{W}$.*

(i) *The loss function is L-smooth, i.e., $\|\nabla\mathcal{L}(w; x, y) - \nabla\mathcal{L}(w'; x, y)\| \le L\|w - w'\|$.*

(ii) *The loss function is μ-strongly convex, i.e., $\frac{\mu}{2}\|w - w'\|^2 \le \mathcal{L}(w; x, y) - \mathcal{L}(w'; x, y) - (w - w')^\top\nabla\mathcal{L}(w'; x, y)$*

(iii) *The loss function $\mathcal{L}(w, x, y)$ is ρ-Lipschitz continuous, i.e., $\|\mathcal{L}(w; x, y) - \mathcal{L}(w'; x, y)\| \le \rho\|w - w'\|$.*

(iv) *The gradient norm of the loss function is uniformly bounded by G, i.e., $\|\nabla\mathcal{L}(w; x, y)\| \le G$.*

The upper bound of the generalization error is provided below.

Theorem 1 (Generalization Error)**.** *If Assumption 1 holds, then for $\beta \le \min\{\frac{1}{2L}, \frac{\mu}{8\rho G}\}$ and $\gamma \le \frac{1}{4L + 2\beta\rho G}$, the generalization error of the foundation meta-model obtained by Algorithm 1 can be bounded by:*

$$\mathcal{E}_{gen} \le \delta := \mathcal{O}(\frac{G^2(1 + \beta LB)S}{KD})[1 - (1 - \gamma c_g)^{ET}], \tag{8}$$

where $c_g = \frac{2\mu(2L+\rho\beta G)}{16(2L+\rho\beta G)+\mu}$, and T represents the iterations.

Theorem 1 guarantees that the generalization error of the foundation meta-model decays at a rate of $\mathcal{O}(BS/KD)$, where S is the number of collaborative agents on a specific task and D is the number of available labeled data samples maintained by each agent. Conversely, our bound demonstrates that increasing inner-layer learning rate β, mini-batch size B, or local adaptation round E degrades the generalization performance. In essence, while a larger mini-batch size B seems beneficial for optimization toward the local objectives of agents and global convergence speed, it can inadvertently lead to poor generalization if not balanced with appropriate values for β, E, T. Notably, this bound is in the asymptotic form concerning T, which is more conducive to an accurate characterization of generalization performance at any training stage.

B. Personalization Error

We first state the additional assumption for proving the following upper bound on personalization error.

Assumption 2 (Bounded Gradient Variance)**.** *Assume that for any $x \in \mathcal{X}, y \in \mathcal{Y}$, the variance of stochastic gradients is bounded, i.e., $\mathbb{E}[\|\nabla\mathcal{L}(.; x, y) - \mathbb{E}[\nabla\mathcal{L}(.; x, y)]\|^2] \le \sigma^2$.*

Theorem 2 (Personalization Error)**.** *Define the gap between the foundation meta-model w_0 and the optimal model of target agent w_k^* as c_I, i.e., $\|w_0 - w_k^*\| \le c_I$. If Assumption 1-2 hold. then with $\beta = \frac{2}{\mu(E+1)}$, we have*

$$\mathcal{E}_{per} \le \underbrace{\mathcal{O}(\frac{\sigma^2}{\mu E})}_{\text{Adaptation error}} + \underbrace{\mathcal{O}(\frac{L}{2}c_I^2(1 - \frac{\mu}{L})^E)}_{\text{Residual initial deviation}}. \tag{9}$$

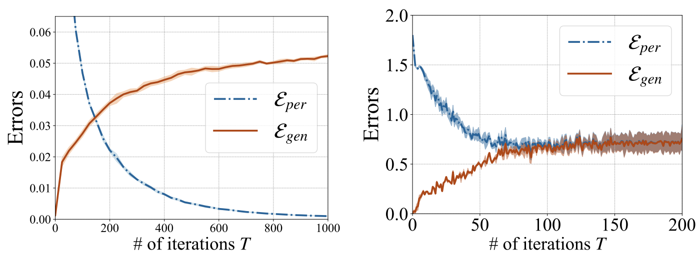
Theorem 2 indicates that as the number of local adaptation rounds E increases, the first term on the RHS denotes the adaptation error, which decays at a rate of $\mathcal{O}(\frac{1}{E})$, reflecting improved convergence towards the local optimum. The second term denotes residual initial deviation, which diminishes exponentially as the initial model discrepancy c_I is progressively corrected. Moreover, the term c_I captures the data distribution mismatch across agents due to environmental differences.

C. Tradeoff between Personalization and Generalization

Based on the results in Theorems 1 and 2, we can observe a fundamental tradeoff between generalization and personalization

TABLE I: Recognition accuracy of agents deployed at different environments trained by different algorithms, including local training, joint training, and the proposed MAN based on (a) MNIST and (b) Widar3.0.

Dataset	Eval. Condition	Local Training	Joint Training	Ours
MNIST	seen (worst)	96.17±0.27	95.59±0.45 ↓	97.28±0.04 ↑
	seen (best)	98.39±0.08	97.88±0.24 ↓	99.33±0.03 ↑
	unseen (worst)	/	93.52±0.82	96.43±0.28 ↑
	unseen (best)	/	96.85±0.73	97.98±0.26 ↑
Widar3.0	seen (worst)	81.03±0.53	68.06±1.01 ↓	85.06±0.78 ↑
	seen (best)	96.22±0.43 ↑	84.50±1.42 ↓	95.11±0.69
	unseen (worst)	/	83.25±1.77 ↑	82.79±1.12
	unseen (best)	/	84.51±1.62	86.69±0.48 ↑



(a) (b)

Fig. 2: Generalization and personalization performance under different training iterations based on (a) MNIST and (b) Widar3.0.

errors. Also, the tradeoff point between both errors is controlled by a combination of adjustable variables including T, S, E, and β. From the generalization perspective, larger T and S indeed benefit in-depth empirical training, but overfitting to the training data harms generalization and potentially affects personalization. As the factors associated with local adaptation, increasing E or β reduces personalization errors but amplifies generalization error, and vice versa. This is due to the fact that excessive local adaptation traps the meta-model in agent-specific short-sighted objectives, compromising its ability to capture generalizable patterns. To mitigate these non-benign tradeoffs, the hyperparameters need to be carefully coordinated to ensure all error terms converge at the same rate. We will provide a more detailed discussion later in the next section.

V. Experimental Results

A. Experimental Setup

In this section, we evaluate the performance of our proposed MAN using the following datasets and model configuration.

Dataset: For broader verification, we consider two different types of real-world datasets: MNIST [9] for image identification and Widar3.0 [10] for radio frequency (RF) signal identification. MNIST is a widely used image dataset containing 70,000 handwritten digit images with 28×28 gray-scale pixels ranging from 0-9. Widar3.0 is an RF-based human gesture recognition dataset comprising 18 distributed Wi-Fi receivers. Each receiver collects

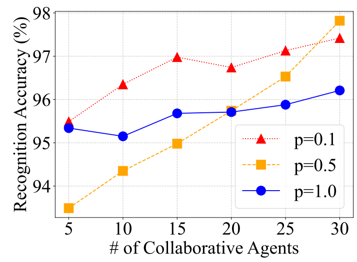


Fig. 3: Recognition accuracy of model trained with different number of collaborative agents under different non-i.i.d. conditions based on MNIST.

the channel state information (CSI) at a sampling rate of 1kHz over 3 antennas and 30 subcarriers when a person performs 6 types of gestures. The dataset consists of 12,000 labeled gesture data samples in total.

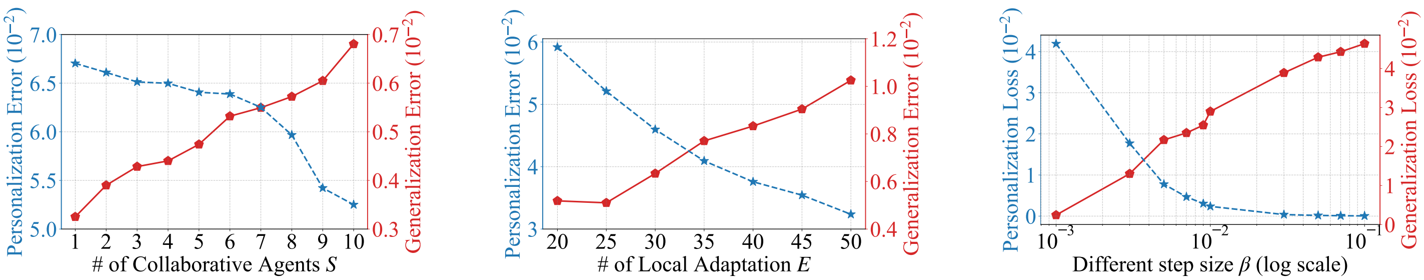
Model: For MNIST, we use a convolutional neural network (CNN) with 2 convolutional layers, 2 max-pooling layers, and 2 fully connected layers. For Widar3.0, we deploy a CNN with 2 convolutional layers, 2 max-pooling layers, 1 dropout layer, and 2 fully connected layers. Besides, we choose the optimizer as stochastic gradient descent.

Default Parameters: Unless otherwise specified, the default parameters are set as the overall number of agents $K = 10$, the number of collaboration round $T = 1,000$ for MNIST and $T = 200$ for Widar3.0, inner-layer learning rate $\beta = 0.001$ and outer-layer learning rate $\gamma = 0.005$, mini-batch size $|\mathcal{B}^{m,in}_{k,e}| = |\mathcal{B}^{m,out}_{k,e}| = 32$, and local adaptation steps $E = 30$. Besides, the Dirichlet distribution based on parameter p is used to control the heterogeneity degree of the data distribution of different agents, where p is set to 0.5 by default.

B. Experimental Result

In Fig. 2, we evaluate the personalization and generalization performance of MAN under different training iterations. As training progresses, both the personalization and generalization errors converge to stable levels, demonstrating that our proposed method effectively balances these objectives. In Fig. 3, we compare the converged recognition accuracy when the foundation model is trained with varying numbers of collaborative agents under different non-i.i.d. conditions. Here, $p = 1.0$ indicates that all agents operate in identical environments, while $p = 0.1$ represents extremely heterogeneous environments. We can observe that the accuracy always improves as the number of collaborative agents increases. In particular, this improvement is most pronounced in moderately heterogeneous scenarios, whereas the gains are more modest in both extremely non-i.i.d. and fully i.i.d. scenarios.

Table. I presents a comparative evaluation of different training approaches for the recognition task on MNIST and Widar3.0 datasets. Specifically, we benchmark our proposed MAN with two baseline methods: *(i) Local Training:* Each agent trains



(a) (b) (c)

Fig. 4: Personalization, generalization errors of MAN under different S, E, and β based on MNIST.

a local model based only on its local dataset and *(ii) Joint Training:* One of the most well-known methods is FedAvg [11] in which all agents collaboratively train a single shared model through periodic model aggregation. We report the best and worst performance for the cases where the agents are deployed in 10 known environments seen during the training phase and five previously unseen deployment environments, respectively, to exhaustively elaborate on the personalization and generalization capabilities of the model. Our proposed MAN demonstrates superior performance in both personalization and generalization, achieving a favorable tradeoff between them.

C. *Verification of Theoretical Results*

As illustrated in Fig. 4, we validate our theoretical results on the tradeoff between local personalization and global generalization by analyzing the effects of the number of collaborating agents S, the number of local adaptation rounds E, and the adaptation step size β based on MNIST. In Fig. 4(a), we fix the overall number of agents and vary the number of collaborating agents S. We observe that increasing S leads to a larger generalization error, which corroborates Theorem 1. However, more collaboration does not always improve personalization, as personalization performance depends heavily on the alignment between an agent's local data distribution and the collective distribution of the collaborating agents, verifying Theorem 2. Fig. 4(a) validates the tradeoff from a global collaboration perspective, while Fig. 4(b) and Fig. 4(c) explore it through local adaptation. Specifically, we observe that increasing either the number of local adaptation rounds E or the step size β reduces personalization error, aligned with Theorem 2, but increases generalization error, which supports Theorem 1. This occurs because, as observed in the previous discussion, larger E or β drives agents to prioritize local personalized knowledge of the specific environment over global shared learning.

VI. Conclusion

This paper investigated the fundamental tradeoff between generalization and personalization in AgentNet. We proposed a novel framework, named MAN, to efficiently balance these two competing objectives. In MAN, foundation meta-models are developed to capture shared knowledge across all agents. Each individual agent then fine-tunes the assigned meta-initialization by the agent controller to meet its personalized preferences. Moreover, we derived asymptotic theoretical bounds on generalization and personalization errors, shedding light on how these errors evolve under different conditions. Finally, extensive experiments based on real-world datasets demonstrated that MAN balances the tradeoff between generalization and personalization.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (NSFC) under grants 62301516 and 62525109, and the Mobile Information Network National Science and Technology Key Project under grant 2024ZD1300700.

References

[1] W. Saad, O. Hashash, C. K. Thomas, C. Chaccour, M. Debbah, N. Mandayam, and Z. Han, "Artificial general intelligence (AGI)-native wireless systems: A journey beyond 6G," *arXiv preprint arXiv:2405.02336*, 2024.

[2] Z. Durante *et al.*, "Agent AI: Surveying the horizons of multimodal interaction," *arXiv preprint arXiv:2401.03568*, 2024.

[3] Y. Xiao, G. Shi, and P. Zhang, "Towards agentic AI networking in 6G: A generative foundation model-as-agent approach," *IEEE Communications Magazine*, vol. 63, no. 9, Sep. 2025.

[4] Y. Xiao *et al.*, "Reasoning over the air: A reasoning-based implicit semantic-aware communication framework," *IEEE Trans. Wireless Commun.*, vol. 23, no. 4, pp. 3839–3855, Apr. 2024.

[5] Y. Xiao, G. Shi, Y. Li, W. Saad, and H. V. Poor, "Toward self-learning edge intelligence in 6G," *IEEE Communications Magazine*, vol. 58, no. 12, pp. 34–40, Dec. 2020.

[6] Y. Talebirad and A. Nadiri, "Multi-agent collaboration: Harnessing the power of intelligent LLM agents," *arXiv preprint arXiv:2306.03314*, 2023.

[7] W. Liang, J. Wang, W. Bao, X. Zhu, Q. Wang, and B. Han, "Continuous self-adaptive optimization to learn multi-task multi-agent," *Complex & Intelligent Systems*, vol. 8, no. 2, pp. 1355–1367, 2022.

[8] X. Meng and Y. Tan, "PMAC: Personalized multi-agent communication," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 16, 2024, pp. 17 505–17 513.

[9] Y. LeCun, "The mnist database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*, 1998.

[10] Y. Zheng, Y. Zhang, K. Qian, G. Zhang, Y. Liu, C. Wu, and Z. Yang, "Zero-effort cross-domain gesture recognition with wi-fi," in *ACM MobiSys*, New York, USA, Jun. 2019, pp. 313–325.

[11] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, Ft. Lauderdale, USA, Apr. 2017, pp. 1273–1282.