pFedNavi: Structure-Aware Personalized Federated Vision-Language Navigation for Embodied AI

Qingqian Yang, Hao Wang
Stevens Institute of Technology
Hoboken, NJ, USA
{qyang21,hwang9}@stevens.edu

Sai Qian Zhang
New York University
New York, NY, USA
sai.zhang@nyu.edu

Jian Li
Stony Brook University
Stony Brook, NY, USA
jian.li.3@stonybrook.edu

Yang Hua
Queen's University Belfast
Belfast, UK
Y.Hua@qub.ac.uk

Miao Pan
University of Houston
Houston, TX, USA
mpan2@central.uh.edu

Tao Song, Zhengwei Qi, Haibing Guan
Shanghai Jiao Tong University
Shanghai, China
{songt333,qizhwei,hbguan}@sjtu.edu.cn

***Abstract*—Vision-Language Navigation (VLN) requires large-scale trajectory–instruction data from private indoor environments, raising significant privacy concerns. While Federated Learning (FL) mitigates this by keeping data on-device, *vanilla* FL struggles under VLN's extreme cross-client heterogeneity in environments and instruction styles, rendering a single global model suboptimal. This paper proposes *pFedNavi*, a *structure-aware* and *dynamically adaptive* personalized federated learning framework tailored for VLN. Our key idea is to personalize *where it matters*: *pFedNavi* (i) adaptively identifies client-specific layers via layer-wise mixing coefficients, and (ii) performs fine-grained parameter fusion on the selected components (e.g., the encoder–decoder projection and environment-sensitive decoder layers) to balance global knowledge sharing with local specialization. We evaluate *pFedNavi* on two standard VLN benchmarks, R2R and RxR, using both ResNet and CLIP visual representations. Across all metrics, *pFedNavi* consistently outperforms the FedAvg-based VLN baseline, achieving up to 7.5% improvement in navigation success rate and up to 7.8% gain in trajectory fidelity, while converging 1.38× faster under non-IID conditions.**

***Index Terms*—Vision-Language Navigation, Personalized Federated Learning, Embodied AI**

I. Introduction

Vision-Language Navigation (VLN) has emerged as a popular and important task in embodied AI, where an agent must interpret natural language instructions and navigate through a visual environment [1, 2]. Solving VLN hinges on abundant, high-quality training data because the agent must perform complex multimodal reasoning, grounding ambiguous natural language in visual observations to reach the target [3]. VLN data often comes from *private homes or offices*—navigation trajectories and human instructions that can reveal *sensitive information* like house layouts, objects present, or personal habits. Fig. 1 illustrates typical VLN scenarios and the heterogeneity across different houses for embodied AI agents executing VLN tasks. Indeed, conventional VLN training assumes centralizing all user data on a server, which ignores privacy concerns. Recent studies have noted that most VLN research has overlooked these real-world privacy issues [2, 4], creating a gap between academic work and practical deployment in personal spaces.

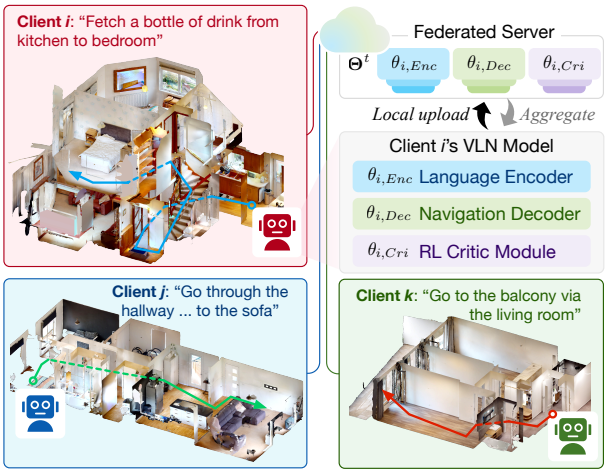

Fig. 1: Environmental heterogeneity across different houses for VLN task. Each client corresponds to a distinct house with substantially different spatial layout and structural characteristics.

Federated Learning (FL) offers a promising solution to this dilemma by enabling privacy-preserving collaborative training. In an FL setting, each embodied agent (or each environment) keeps its data locally and only shares model updates with a central server. For example, FedVLN [2] was recently proposed as the first federated VLN framework, showing that decentralized training on house-specific data can achieve navigation performance comparable to centralized training while keeping each user's data private. As Fig. 1 shows, each house environment in FedVLN is treated as a client that trains a local VLN agent on its private data and periodically shares only the model parameters for aggregation.

However, standard FL alone is not enough—a single global

This work was supported in part by the National Natural Science Foundation of China (NO. 62472284), Openmind (Wuhu) Intelligent Robot Co., Ltd., and Shanghai Key Laboratory of Scalable Computing and Systems. M. Pan's work was supported in part by the US National Science Foundation under grants CNS-2107057, CNS-2318664, CSR-2403249, and CNS-2431596.

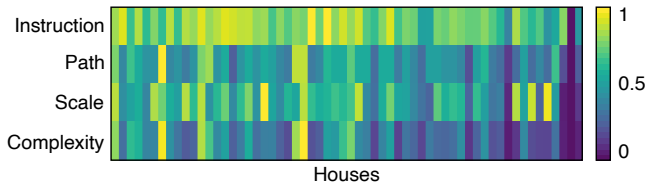

Fig. 2: Data heterogeneity analysis on RxR dataset [6]. We visualize house-level statistics along four dimensions: *Instruction*, measured by the average instruction length; *Path*, measured by the variance of navigation trajectory lengths within each house; *Scale*, measured by the number of rooms, indicating the house size; and *Complexity*, measured by the number of nodes in the induced navigation graph, indicating how many navigation states and decisions the agent needs to encounter. All statistics are normalized to [0,1] across houses. Darker colors indicate lower values, while lighter colors indicate higher values.

model cannot adequately serve every user when the data are highly heterogeneous. FedVLN relies on the basic FedAvg aggregation [5], which struggles under VLN's *severe non-IID conditions*. In practice, different clients' navigation data follow very different distributions, so blindly averaging their models can lead to an aggregated agent that *deviates from the optimal* for any individual environment. This limitation motivates the need for personalization on top of federated training.

Why personalized Federated Learning (pFL) for VLN? In real-world VLN deployments, the data heterogeneity is extreme, calling for pFL rather than a *one-size-fits-all* model. Specifically, enabling pFL for VLN is challenging because of several unique characteristics: **1) Heterogeneous Environments:** Each FL client corresponds to a distinct physical environment, e.g., different houses in Room-to-Room (R2R) and Room-across-Room (RxR). In particular, environments differ significantly in their spatial extent, layout organization, and navigation structure, resulting in induced navigation graphs with various sizes, connectivity patterns, and trajectory distributions. As illustrated in Fig. 2, different clients exhibit substantial heterogeneity in their underlying navigation environments along multiple complementary dimensions, including house scale, structural complexity, and path characteristics. **2) Personalized Instructions:** Language instructions are inherently personal and context-specific. Different users may describe routes in unique ways—with different vocabulary, levels of detail, or referring to custom landmarks. Fig. 2 further highlights such linguistic heterogeneity across clients. **3) Complex Multimodal Models:** VLN agents typically consist of multiple modules (vision encoder, language encoder, attention-based policy network, sometimes a value estimator), which interact sequentially to produce navigation decisions.

To address the above challenges, we propose *pFedNavi*, a model structure-aware personalized federated learning framework tailored for vision-language navigation. The key insight behind *pFedNavi* is that effective VLN personalization must be both structure-aware and dynamically adaptive. In other words, *pFedNavi* treats different parts of the VLN agent differently, allowing each client to adapt the model's components to its data in a fine-grained way rather than applying coarse global updates. Building on this insight, we make the following contributions:

- **Selective Module Personalization:** We develop a layer-wise adaptive personalization strategy that learns which parts of the VLN model to personalize for each client.
- **Fine-Grained Parameter Fusion:** For the model parts identified as needing personalization, we propose a fine-grained parameter fusion mechanism to combine global and local knowledge.
- Our extensive experiments on standard VLN benchmarks, R2R [1] and RxR [6], confirm that *pFedNavi* significantly outperforms conventional federated VLN models (which lack personalization) and even approaches the performance of centrally-trained models. Specifically, *pFedNavi* achieves up to 7.5% improvement in success rate, 7.8% in normalized dynamic time warping (nDTW), highlighting its effectiveness in improving task completion as well as trajectory fidelity in VLN. Moreover, *pFedNavi* achieves 1.38× faster loss convergence than baseline.

II. Background and Motivation

A. Vision-Language Navigation (VLN)

VLN [1] is a multimodal navigation task in which an embodied AI agent must follow natural-language instructions to reach a target location by making sequential decisions from egocentric visual observations. A defining property of VLN is its strong *environment heterogeneity*. In standard benchmarks such as R2R [1] and RxR [6], each environment (e.g., a building/scan) presents a distinct indoor layout, topology, object distribution, and navigation affordances, leading to substantially different visual and trajectory distributions across environments.

Meanwhile, VLN instructions are naturally *personalized*: different users (or annotators) describe the same route with different vocabulary, granularity, and landmark choices, and this variability is further amplified in multilingual settings such as RxR [6]. These two factors jointly induce severe non-IID behavior in VLN training and deployment.

To improve generalization, prior work has explored diverse modeling and training strategies, including imitation and reinforcement learning, structured reasoning via graph-based representations [7], long-term memory for accumulating navigation context [8], and data augmentation / fine-tuning with synthesized trajectory-instruction pairs [9]. Recent studies further consider pre-exploration for better adaptation to unseen environments [10] and leverage LLMs for explicit planning and reasoning (e.g., NavGPT [11]). However, these methods largely assume *centralized* access to training data, which is often unrealistic in practice because VLN trajectories and instructions are collected in private indoor spaces (e.g., homes, offices, and labs), making raw data sharing undesirable.

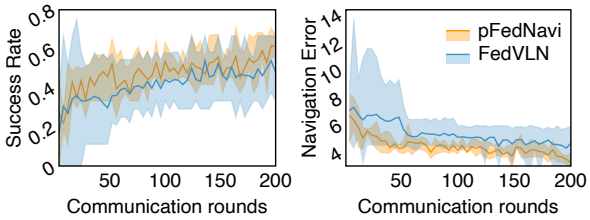

Fig. 3: Success rate and navigation error comparison on R2R dataset. The line and the shadow mean the average and variance of performance across clients.

B. Federated VLN Formulation

FedVLN [2], recently proposed as the first federated VLN framework, trains the trajectory model, language encoder, multimodal decision module, and speaker model across clients using FedAvg [5], and further introduces a federated pre-exploration phase for adaptation after deployment. A VLN agent deployed in a house environment i acts as a client in FL. In each local round, the agent follows a set of human linguistic instructions $\mathcal{I} = \{I_1, I_2, \ldots\}$, where each instruction $I \in \mathcal{I}$ is a sequence of tokens $I = \langle w_1, w_2, \ldots, w_m \rangle$ (m is the instruction length and w_j is a word token). For each instruction, the agent executes a navigation episode and produces a trajectory $\tau = \langle o_0, a_0, \ldots, o_T \rangle$ by interacting with the environment, terminating upon issuing a `STOP` action. Each local dataset $\mathcal{D}_i$ therefore consists of instruction–trajectory pairs (I, τ).

The local VLN model contains three modules: a language instruction encoder θ^t_{Enc}, a navigation decoder θ^t_{Dec}, and a critic module θ^t_{Cri} for RL learning. At communication round t, client i receives the global parameters $\mathbf{\Theta}^t = \{\theta^t_{Enc}, \theta^t_{Dec}, \theta^t_{Cri}\}$, and performs local optimization on its private dataset $\mathcal{D}_i$ (paired instruction–trajectory samples).

After local training, the updated parameters $\mathbf{\Theta}^{(t+1)}_i$ are uploaded to the server. The server aggregates updates from a subset of participating clients $\mathcal{S}_t \subseteq \{1, \ldots, N\}$ to construct the next-round global model. This federated optimization proceeds iteratively across rounds. The objective of vanilla federated VLN is to collaboratively learn a navigation policy that can operate across diverse embodied environments without sharing raw data.

C. Motivating pFL for VLN

While FL protects privacy and enables knowledge sharing, *vanilla* federated VLN (e.g., FedVLN [2]) that aggregates all client models into a single global model often *struggles* under VLN's severe data heterogeneity. This has motivated pFL, which aims to learn client-adaptive models rather than enforcing a *one-size-fits-all* solution. Fig. 3 shows that FedAvg-based VLN exhibits large performance variance across FL clients, demonstrating that a single model cannot sufficiently serve all environments.

Existing pFL methods include model-splitting approaches such as FedCP [12], which separate globally shared and client-specific components (e.g., backbone *vs.* head), as well as full-model adaptation approaches such as Per-FedAvg-style personalization [13]. However, VLN introduces *additional*

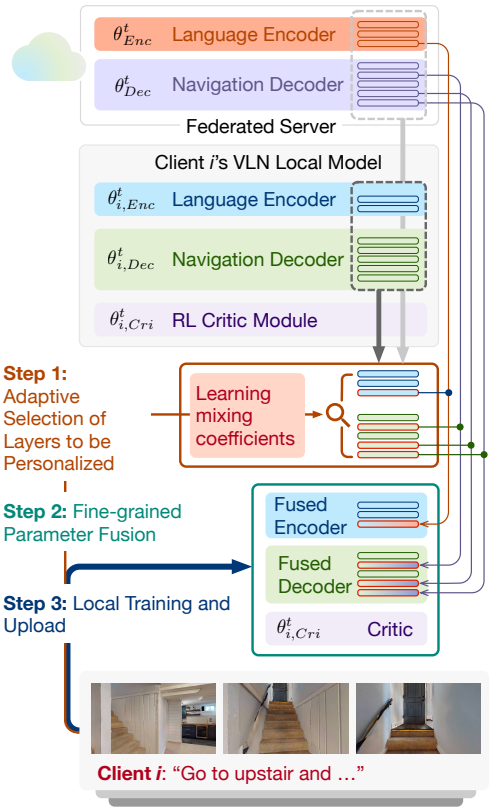

Fig. 4: *pFedNavi*'s workflow. *pFedNavi* operates in three stages: (1) adaptive personalized layer selection; (2) fine-grained parameter fusion; and (3) local training with federated aggregation.

difficulties due to multimodal grounding and tightly coupled encoder-decoder architectures, where indiscriminate personalization can degrade language grounding or action prediction.

III. The Design of *pFedNavi*

A. Overview

Enabling pFL for VLN is fundamentally challenging due to the joint effects of multimodal inputs, tightly coupled encoder-decoder structures, and strong environment-induced heterogeneity across clients. Building on the key insight that effective VLN personalization must be *structure-aware* and *dynamically adaptive*, we design *pFedNavi* in Fig. 4.

Design Objectives: 1) Client-adaptive navigation without sacrificing shared generalization; 2) Structure-aware personalization that preserves grounding; and 3) Stable and data-efficient personalization under severe non-IID.

Key Challenges: These objectives translate into three key challenges that directly motivate our design components:

- **Non-IID drift makes naive aggregation suboptimal.** With extreme cross-environment and cross-instruction heterogeneity, FedAvg-style averaging can produce a global model that is not optimal for any particular client, and local updates can be inconsistent across rounds.
- **VLN module coupling makes indiscriminate personalization harmful.** Unlike standard pFL settings, blindly

personalizing arbitrary layers can easily disrupt language grounding or action prediction because VLN decisions rely on tightly coupled encoder-decoder interactions.

- **"Which layers to personalize" is client- and round-dependent.** Different environments and instruction styles stress different model components; thus, a fixed model split (e.g., backbone and head) is often insufficient, and we need to adaptively *recognize* and *personalize* layers in the global model with clients' local information.

A Bird's Eye View of *pFedNavi*: Step ① Adaptive layer selection: We first identify model components that require personalization, including the encoder-decoder projection layer and selected decoder layers, using adaptive layer-wise mixing. **Step ② Fine-grained parameter fusion:** We apply fine-grained parameter fusion to the selected components to balance global knowledge and local specialization. **Step ③ Local training & upload:** We initialize the local model using the above steps and train on the private data.

B. Formulating pFL for VLN

Personalized federated objective. Let there be N clients, where Client i corresponds to a VLN agent deployed in environment i with private dataset $\mathcal{D}_i$ (paired instruction–trajectory samples). Unlike vanilla FL that optimizes a single shared model, pFL for VLN aims to learn a set of client-specific models $\{\mathbf{\Theta}_i\}_{i=1}^N$: $\min_{\{\mathbf{\Theta}_i\}} \sum_{i=1}^N \mathbb{E}_{(I,\tau)\sim\mathcal{D}_i}\Big[\mathcal{L}_{\text{VLN}}(I,\tau;\mathbf{\Theta}_i)\Big]$, where $\mathcal{L}_{\text{VLN}}$ denotes the standard VLN training objective (e.g., imitation loss and, when used, RL loss/critic terms). To encourage knowledge sharing, we maintain a *global reference* model $\mathbf{\Theta}$ on the server, constructed by aggregating client updates on globally shared parameters.

Why structure-aware personalization. VLN models differ from standard unimodal networks because language understanding and action prediction are tightly coupled through structured modules. Different components therefore exhibit different degrees of environment sensitivity. Accordingly, *pFedNavi* treats the VLN model as three modules and personalizes them differently:

- **Encoder.** We follow the standard VLN encoder [1], consisting of (i) an embedding layer, (ii) an instruction BiLSTM encoder, and (iii) an encoder-to-decoder projection. The embedding and BiLSTM mainly capture general linguistic regularities and are thus globally shared, while the projection directly mediates language grounding into navigation intent and is more environment-sensitive. Therefore, *pFedNavi* always includes the encoder-decoder projection layer in personalization.
- **Decoder.** We adopt the standard attention-based decoder used in prior VLN agents [1, 9], which contains multiple functional components (action embedding, visual attention, recurrent state update, instruction attention, candidate scoring). Since these components have heterogeneous sensitivity to local environments, *pFedNavi* does *not* predefine personalized decoder layers; instead, it adaptively selects them for each client and each round (Sec. III-C).
- **Critic.** The critic estimates value functions that depend heavily on environment-specific dynamics and reward landscapes. To avoid destabilizing training via cross-client averaging, *pFedNavi* keeps the critic *local* and does not force it to match the global reference.

C. Adaptive Layer Selection for Personalization

Given the received global model and the client's previous local model, we want to decide *which* decoder layers should be personalized for client i at round t.

Therefore, instead of predefining personalized layers, we adopt a layer-wise adaptive fusion mechanism to automatically identify decoder components that benefit from personalization. Specifically, we associate each decoder component (layer) $\ell \in \mathrm{L}_{\text{dec}}$ with a learnable mixing coefficient $\alpha_{i,\ell}^t \in [0,1]$ for client i at round t. The fused parameters are defined as $\hat{\theta}_{i,\ell}^{t+1,0} = \big(1-\alpha_{i,\ell}^t\big)\,\theta_\ell^t + \alpha_{i,\ell}^t\,\theta_{i,\ell}^t$, where θ_ℓ^t denotes the globally aggregated parameters of decoder layer ℓ at round t, and $\theta_{i,\ell}^t$ is the corresponding previous local parameters.

We learn $\boldsymbol{\alpha}_i^t = \{\alpha_{i,\ell}^t\}_{\ell\in\mathrm{L}_{\text{dec}}}$ by minimizing the teacher-forcing imitation loss on a small local batch set $\mathcal{B}_i \subset \mathcal{D}_i$: $\boldsymbol{\alpha}_i^t \leftarrow \arg\min_{\boldsymbol{\alpha}} \ \mathcal{L}_{\text{IL}}\Big(\mathcal{B}_i;\ \hat{\theta}_{i,\text{Dec}}^{t+1,0}(\boldsymbol{\alpha})\Big)$. Layers selected for personalization are then determined by a thresholding rule:

$$\text{Personalized\_layer}_i^t = \big\{\ell \in \mathrm{L}_{\text{dec}} \ \big|\ \alpha_{i,\ell}^t \geq \delta\big\}. \tag{1}$$

Layers with a small $\alpha_{i,\ell}^t$ are treated as globally shared and directly inherited from the global model. Here, δ controls how strongly a layer must prefer local parameters to be considered personalized. We set $\delta = 0.6$ rather than 0.5 to avoid marginal or ambiguous personalization decisions.

D. Fine-grained Parameter Fusion

After identifying personalized components, we perform *fine-grained parameter fusion* to initialize the client's personalized model before local training.

For client i at communication round t, let $\mathcal{K}_i^t$ denote the set of all layers selected for personalization, which includes: (i) encoder-decoder projection layer, and (ii) $\text{Personalized\_layer}_i^t$ identified in Equation (1). For each layer $l \in \mathcal{K}_i^t$, we construct the personalized initialization via element-wise interpolation between the global and previous local parameters:

$$\theta_{i,l}^{t+1,0} = \theta_{i,l}^t + W_{i,l}^t \odot \big(\theta_l^t - \theta_{i,l}^t\big), \tag{2}$$

where θ_l^t denotes the globally aggregated layer parameter, $\theta_{i,p}^t$ is the corresponding parameter from the previous local model, and $W_{i,l}^t \in [0,1]$ is a learnable fusion weight controlling the contribution of global knowledge. This design enables flexible personalization while constraining the additional cost to preserve scalability, as fusion weights are introduced only for a small subset of decoder components.

Parameters not selected for personalization are directly inherited from the global model:

$$\theta_{i,p}^{t+1,0} \leftarrow \theta_p^t, \quad \forall p \notin \mathcal{K}_i^t. \tag{3}$$

The critic module is kept local and initialized as

$$\theta_{i,\text{Cri}}^{t+1,0} \leftarrow \theta_{i,\text{Cri}}^t, \tag{4}$$

since it primarily estimates value functions conditioned on environment-specific dynamics. The fusion weights $\mathbf{W}_i^t = \{W_{i,l}^t\}_{l \in \mathcal{K}_i^t}$ are optimized by minimizing the supervised imitation loss under teacher forcing: $\min_{\mathbf{W}_i^t} \mathcal{L}_{\mathrm{IL}}\Big(\mathcal{D}_i; \boldsymbol{\Theta}_i^{t+1,0}(\mathbf{W}_i^t)\Big)$, which adaptively balances global generalization and local specialization for each personalized parameter.

E. Local Training and Upload

After learning the fusion weight, we calculate the initialized personalized model $\boldsymbol{\Theta}_i^{t+1,0}$ by Equations (2, 3, 4) and then train on the client's local dataset $\mathcal{D}_i$ using the standard VLN objective, which combines imitation loss (IL) and RL loss [9]. After local training, the updated model $\boldsymbol{\Theta}_i^{t+1}$ is uploaded to the server for aggregation: $\boldsymbol{\Theta}^{t+1} = \sum_{i \in \mathcal{S}_t} \frac{|\mathcal{D}_i|}{\sum_{j \in \mathcal{S}_t} |\mathcal{D}_j|} \boldsymbol{\Theta}_i^{t+1}$, where $\mathcal{S}_t$ denotes the set of participating clients at round t and $|\mathcal{D}_i|$ is the size of client i's local dataset. The aggregated global model $\boldsymbol{\Theta}^{t+1}$ is then broadcast to clients for the next communication round.

IV. Evaluation

A. Setup

Dataset and Model: We evaluate *pFedNavi* on two widely used VLN datasets: Room-to-Room (R2R) [1] and Room-across-Room (RxR) [6], which are both constructed from the Matterport3D dataset. R2R contains 7K navigation trajectories paired with human-written English instructions. RxR [6] is a larger multilingual dataset with more trajectories and denser linguistic grounding, where each word is time-aligned to annotator viewpoints. We follow the standard data preprocessing and environment configurations used in prior VLN work [2]. We adopt the VLN agent architecture consistent with prior studies [1] and evaluate it with two widely used alternative pretrained visual feature extractors, ResNet-152 and CLIP. The navigation instructions are encoded using an LSTM-based language encoder, and an attention-based LSTM decoder attends to both textual and visual features to predict actions at each step. The critic module is implemented to estimate state values during reinforcement learning.

FL Setup: We regard each agent in a building environment as an FL client. All experiments are conducted on a single machine with two NVIDIA RTX 3090 GPUs bridged with NVLink. We reuse the federated learning hyperparameters, such as the number of training rounds, the number of local epochs, and both the local and global learning rates, from FedVLN [2]. Specifically, at each communication round, a subset of clients is randomly sampled with a participation rate of $S_r = 0.2$. Each participating client performs local training for 5 epochs, and all models are trained until convergence. For personalized layer selection, the mixing coefficients $\boldsymbol{\alpha}$ are optimized using a learning rate of $\alpha_{\mathrm{lr}} = 0.1$ for $S_\alpha = 2$ steps. We select personalized decoder layers using a threshold $\delta = 0.6$. For fine-grained fusion, the fusion weights are updated with a learning rate of $\eta = 0.1$ for $S_W = 1$ step per round, and are fully optimized until convergence only in the second round, after which they are lightly fine-tuned.

TABLE I: Evaluation Results on R2R and RxR Datasets (all metrics except NE in %). ResNet and CLIP indicate the visual feature extractors used by VLN agent, with the model architecture remaining identical. ↑ indicates higher values correspond to better performance, whereas ↓ indicates lower values are better. *Please note that EnvDrop [9] is a centralized learning baseline.*

	Method on R2R	SR ↑	SPL ↑	OSR ↑	CLS ↑	nDTW ↑	NE ↓
ResNet	EnvDrop [9]	**56.7**	**54.3**	63.9	**67.2**	**56.0**	**4.55**
	FedVLN [2]	50.7	47.3	60.3	63.4	50.8	5.37
	pFedNavi	54.5	51.7	**65.2**	66.4	54.8	4.86
CLIP	EnvDrop [9]	**61.5**	56.5	**69.8**	67.3	55.4	3.94
	FedVLN [2]	59.5	56.3	65.1	67.5	55.1	4.24
	pFedNavi	60.7	**57.4**	67.3	**68.6**	**57.1**	**3.65**
	Method on RxR	**SR ↑**	**SPL ↑**	**OSR ↑**	**CLS ↑**	**nDTW ↑**	**NE ↓**
ResNet	EnvDrop [9]	**41.3**	36.6	50.4	57.2	**53.6**	**8.03**
	FedVLN [2]	39.9	33.1	47.8	55.5	50.7	8.62
	pFedNavi	40.1	**36.7**	**51.2**	**57.3**	52.9	8.10
CLIP	EnvDrop [9]	**47.7**	**43.4**	56.1	**61.2**	56.8	**6.53**
	FedVLN [2]	43.0	39.4	51.5	58.7	54.8	7.37
	pFedNavi	46.1	41.2	**56.3**	59.8	**56.9**	6.91

Baselines: We compare our method with both centralized and federated learning approaches. For centralized training, we use **EnvDrop** [9], which employs a BiLSTM-based language encoder, an attentive LSTM decoder for action prediction. We evaluate EnvDrop under two visual feature extraction settings, using either pretrained ResNet-based features or CLIP-based features. For federated learning, we evaluate **FedVLN** [2], the first work that applies FedAvg [5] to the VLN setting.

Metrics: To comprehensively assess navigation ability, we consider two categories of evaluation metrics: 1) goal-reaching performance and 2) trajectory fidelity. 1) **Success Rate (SR)** computes the proportion of episodes in which the agent issues a `stop` action within 3 meters of the target. **Success weighted by Path Length (SPL)** extends SR by incorporating path optimality, rewarding agents that reach the goal with shorter and more efficient paths. **Oracle Success Rate (OSR)** reflects the agent's ability to at least visit a promising region even if it fails to stop correctly. **Navigation Error (NE)** reports the average distance between the agent's final position and the goal location, providing a continuous measure of goal accuracy. 2) **Coverage weighted by Length Score (CLS)** jointly considers spatial coverage and trajectory length to evaluate how well the agent explores relevant regions. **Normalized Dynamic Time Warping (nDTW)** aligns the predicted and reference trajectories to quantify their similarity. Together, these metrics assess high-level task completion as well as low-level behavioral fidelity, both of which are essential for VLN evaluation. Notably, we use the average client-side value of each metric for our *pFedNavi* method and compare it with the corresponding global metric value.

B. The Effectiveness of pFedNavi

Table I summarizes the performance comparison among centralized training, FedAvg-based federated learning, and our personalized federated approach on both R2R and RxR datasets. Overall, *pFedNavi* consistently outperforms FedVLN

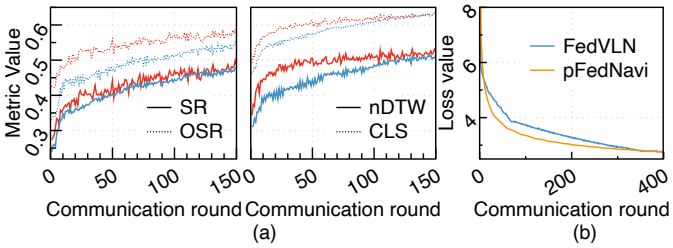

Fig. 5: Comparison between *pFedNavi* and FedVLN on R2R dataset using ResNet-152 visual features. (a) The performance curves of various metrics (e.g., SR, OSR, nDTW, and CLS), and (b) loss convergence over communication rounds (red: *pFedNavi*, blue: FedVLN).

across all metrics under both ResNet-152 and CLIP visual representations, demonstrating the effectiveness of personalization for VLN in the federated setting. Compared with FedVLN, *pFedNavi*'s improvement is particularly evident on metrics that reflect trajectory quality and goal grounding. These gains indicate that personalized federated learning enables each client to better adapt navigation policies to its local environment, alleviating the bias introduced by naive parameter averaging under heterogeneous VLN data.

Besides, we set the client participation rate to $S_r = 1$ for clearer visualization of learning dynamics. Fig. 5(a) compares the learning dynamics of *pFedNavi* and FedVLN on the R2R dataset with ResNet features. Across all metrics, *pFedNavi* achieves faster performance improvement. Notably, the more rapid gains on OSR and nDTW indicate that personalized parameter fusion helps preserve environment-specific navigation patterns and instruction grounding, whereas FedVLN improves more slowly due to averaging across heterogeneous clients.

C. Convergence and Overhead Analysis & Ablation Study

Fig. 5(b) shows the convergence performance compared with FedVLN, demonstrating that *pFedNavi* converges to the target loss $= 3.0$ in 208 communication rounds, whereas FedVLN requires 288 communication rounds, resulting in a 27.8% improvement in efficiency (1.38× faster convergence). Despite this short-term fluctuation, *pFedNavi* converges substantially faster thereafter. Since the critic module is always kept local and only the encoder and decoder modules are transmitted each round, our communication overhead is lower than FedVLN. In wall-clock time, our approach costs 3.6 min per round compared with 2.2 min for FedVLN.

We evaluate three variants to analyze the impact of personalized layer selection: 1) All layers, where all model parameters conduct parameter fusion for each client; 2) No layer, where all parameters are globally shared (FedVLN); and 3) *pFedNavi*. Table II shows that personalizing all layers leads to a clear performance degradation across all metrics. This indicates that fully localizing the entire VLN model is ineffective under federated settings, as each client typically has limited data and cannot reliably learn both general navigation knowledge and environment-specific behaviors. In addition, personalizing all layers incurs substantially higher computational and storage overhead. In contrast, *pFedNavi* selectively personalizes structure-sensitive components, achieving superior performance while significantly reducing training cost and model storage.

TABLE II: Evaluation results for various personalized layer selection strategies.

	SR ↑	SPL ↑	OSR ↑	CLS ↑	nDTW ↑	NE ↓
All layers	32.7	30.1	41.0	42.5	45.2	8.52
No layer	39.9	33.1	47.8	55.5	50.7	8.62
pFedNavi	**40.1**	**36.7**	**51.2**	**57.3**	**52.9**	**8.10**

V. Conclusion

This paper studies pFL for VLN under realistic heterogeneity, where each client corresponds to a distinct indoor environment with unique spatial structure and personalized instructions. We propose *pFedNavi*, a structure-aware personalized FL framework that adaptively selects components for personalization via layer-wise mixing and initializes client models through fine-grained parameter fusion, effectively balancing global generalization and local adaptation. Experiments on R2R and RxR demonstrate that *pFedNavi* consistently outperforms FedVLN, achieving up to 7.5% improvement in success rate and 7.8% improvement in normalized dynamic time warping, while converging 1.38× faster.

References

[1] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. D. Reid, S. Gould, and A. van den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *Proc. CVPR*, 2018.

[2] K. Zhou and X. E. Wang, "FedVLN: Privacy-preserving federated vision-and-language navigation," in *Proc. ECCV*, 2022.

[3] K. He, C. Si, Z. Lu, Y. Huang, L. Wang, and X. Wang, "Frequency-enhanced data augmentation for vision-and-language navigation," in *Proc. NeurIPS*, 2023.

[4] C. Miao, T. Chang, M. Wu, H. Xu, C. Li, M. Li, and X. Wang, "FedVLA: Federated vision-language-action learning with dual gating mixture-of-experts for robotic manipulation," in *Proc. ICCV*, 2025.

[5] 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*. PMLR, 2017, pp. 1273–1282.

[6] A. Ku, P. Anderson, R. Patel, E. Ie, and J. Baldridge, "Room-Across-Room: Multilingual vision-and-language navigation with dense spatiotemporal grounding," in *Proc. EMNLP*, 2020.

[7] S. Chen, P.-L. Guhur, M. Tapaswi, C. Schmid, and I. Laptev, "Think global, act local: Dual-scale graph transformer for vision-and-language navigation," in *Proc. CVPR*, 2022.

[8] J. Krantz, S. Banerjee, W. Zhu, J. Corso, P. Anderson, S. Lee, and J. Thomason, "Iterative vision-and-language navigation," in *Proc. CVPR*, 2023.

[9] H. Tan, L. Yu, and M. Bansal, "Learning to navigate unseen environments: Back translation with environmental dropout," in *Proc. NAACL*, 2019.

[10] Z. Wang, X. Li, J. Yang, Y. Liu, J. Hu, M. Jiang, and S. Jiang, "Lookahead exploration with neural radiance representation for continuous vision-language navigation," in *Proc. CVPR*, 2024.

[11] G. Zhou, Y. Hong, and Q. Wu, "Navgpt: Explicit reasoning in vision-and-language navigation with large language models," in *Proc. AAAI*, 2024.

[12] J. Zhang, Y. Hua, H. Wang, T. Song, Z. Xue, R. Ma, and H. Guan, "Fedcp: Separating feature information for personalized federated learning via conditional policy," in *Proc. SIGKDD*, 2023.

[13] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," in *Proc. NeurIPS*, 2020.