# FEDHVLN: BRIDGING HETEROGENEOUS MODELS FOR FEDERATED VISION-LANGUAGE NAVIGATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Federated vision-language navigation (VLN) trains agents from private environments, but averaging native models requires matching parameter shapes. We introduce FedHVLN, which shares compatible language-side modules while retaining each client's navigator. Four stages train a human-instruction encoder and transformation, a federated Speaker that turns local route context into pseudo-instructions, a feature generator aligned with human features, and pseudo-instruction navigation. Clients upload only the shared modules designated at each stage; inference uses the human-instruction path. On the Room-to-Room (R2R) validation split (1,021 held-out instructions from 56 previously seen environments), the mixed federation reaches 64.45% success rate (SR) and 61.69% success weighted by path length (SPL). At fixed round 991, aggregating the shared interface across both model sizes improves pooled SR and SPL by 3.14 and 3.23 percentage points over matched single-size controls. Both evaluated subsets have positive point differences on all five navigation metrics, although exploratory scene-resampling intervals for the compact-model subset include zero. In homogeneous federations, FedHVLN improves all five metrics over FedVLN, with SR/SPL gains of 1.37/1.49 and 2.15/2.74 points for the wide and compact models, respectively. These results support sharing a compatible language interface across the two tested navigator structures on R2R validation.

## 1 INTRODUCTION

Vision-language navigation (VLN) requires an embodied agent to follow natural-language instructions using visual observations of its surroundings (Anderson et al., 2018). Learning from navigation experience across different environments can help agents operate beyond a single location. However, observations and trajectories collected in private environments can contain sensitive information, limiting the sharing of these data for centralized training. Federated VLN enables clients to learn collaboratively by sharing model updates while keeping their navigation data local (Zhou & Wang, 2022).

Deployment introduces a further requirement: navigation models must fit the computational and memory capacities of the devices that run them (Qin et al., 2025). These capacities differ across devices, motivating compact models for constrained hardware and larger models for hardware with greater resources (Diao et al., 2021). Extending federated VLN to these devices therefore requires clients with different model sizes to learn collaboratively. Such models can have incompatible parameter shapes, preventing direct parameter averaging across clients. This raises the central question of how to share navigation knowledge across models adapted to different hardware budgets.

Existing federated VLN methods aggregate or fuse corresponding model parameters to share knowledge across clients. FedVLN shares language-encoder parameters when adapting agents to new environments, while pFedNavi personalizes client models through layer-wise parameter fusion (Zhou & Wang, 2022; Yang et al., 2026b). **These operations require compatible parameter shapes, preventing direct aggregation across models with different hidden or embedding dimensions.**



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

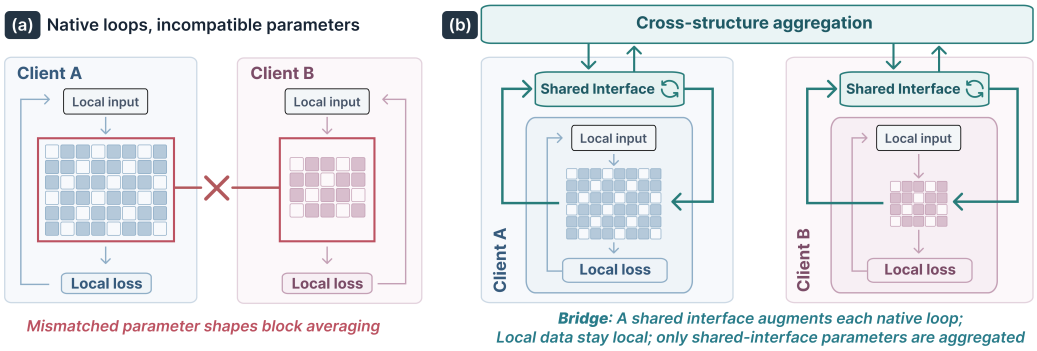


Figure 1: Motivation and design of FedHVLN. (a) Different native parameter shapes prevent direct averaging across model structures. (b) Each client keeps its local input–navigator–loss training loop and attaches a compatible language module to the input-side representation. The local navigation loss trains both the native navigator and this module; only the module's parameters are aggregated across structures. Local navigation data remain on the clients.

An alternative is to transfer knowledge by matching model predictions, which does not require matching parameter shapes (Lin et al., 2020). However, direct prediction matching requires common inputs and corresponding output meanings. **Independently collected trajectories in private environments do not by themselves provide common navigation inputs for comparing predictions.** In the VLN setting considered here, an agent selects its next movement from the viewpoints reachable from its current location. **Because candidate destinations vary across navigation states and environments, the same output position can refer to different movements across clients.**

FedMRL offers a close architectural precedent: it couples heterogeneous local models to an aggregated auxiliary model through client-private projectors (Yi et al., 2024). Its original image-classification task predicts a class for each input. In VLN, an action changes the agent's next observation and available viewpoints; FedHVLN therefore uses shared instruction features to condition each client's history-dependent navigation policy and trains them through local navigation rollouts.

Although clients navigate different environments, their models all need to interpret language instructions. We introduce **FedHVLN**, which integrates structurally compatible language-side modules with each client's native VLN model, as illustrated in Figure 1. The server aggregates only the modules designated at each stage, allowing clients to share language-side updates even when their native VLN parameters have incompatible shapes. Annotated clients obtain a pseudo-instruction by passing local scene–route context through the fixed Speaker; for generator alignment, trainable $G$ processes this generated text while fixed $E$ processes its paired human instruction, without requiring common navigation inputs across clients. The resulting instruction features enter the client's language module, while its visual module and native VLN decoder learn navigation decisions from local supervision, without matching action outputs across environments. Four chronological stages establish the human-instruction path $E \rightarrow D$, federate the Speaker, align $G$ with $E$, and use pseudo-instructions to update the local VLN, $G$, and $D$. At inference, human instructions follow $E \rightarrow D$; native VLN updates and navigation data remain local throughout.

We evaluate FedHVLN on the Room-to-Room (R2R) benchmark (Anderson et al., 2018) using two navigator configurations with different hidden and embedding dimensions. In a heterogeneous federation containing both configurations, FedHVLN achieves a success rate (SR) of 64.45% and success weighted by path length (SPL) of 61.69% over 1,021 validation instructions, demonstrating joint training and navigation with incompatible native parameter shapes. In homogeneous federations using either configuration, FedHVLN outperforms matched FedVLN baselines on all five reported navigation metrics. SPL increases by 1.49 and 2.74 percentage points for the larger and smaller configurations, respectively. These results show that FedHVLN supports heterogeneous model participation while also improving navigation in the evaluated homogeneous settings.

The main contributions of this work are as follows:

- **Federated learning across heterogeneous VLN models.** Building on the shared-auxiliary-model pattern, FedHVLN trains compatible instruction features through local navigation and integrates them into policies that act on changing observations and candidate viewpoints. Its stage-specific shared modules serve navigators with incompatible native parameter shapes, while each client's private projection adapts their features to its native language-module width.
- **Empirical validation in heterogeneous and homogeneous federations.** On R2R, we demonstrate joint training and navigation in a federation containing two structurally incompatible model configurations. In matched homogeneous comparisons, FedHVLN improves all five reported navigation metrics over FedVLN for both configurations, showing that its benefits extend beyond accommodating model heterogeneity.

The remainder of this paper is organized as follows. Section 2 reviews related work on federated navigation and heterogeneous knowledge sharing. Section 3 presents FedHVLN, including the problem formulation, shared representation learning, and federated training procedure. Section 4 presents the experimental evaluation and analysis. Section 5 concludes the paper and discusses its limitations.

## 2 Related Work

**Federated Vision-Language Navigation.** Existing federated VLN methods enable collaborative learning across private environments while adapting navigation models to local conditions. FedVLN aggregates locally trained model parameters and shares language-encoder parameters when adapting agents to new environments (Zhou & Wang, 2022). pFedNavi personalizes client models through layer-wise parameter fusion (Yang et al., 2026b), while FedNASP adjusts the degree of personalization at each navigation step using a shared backbone (Yang et al., 2026a). These approaches address differences in navigation environments and contexts, but their parameter-sharing mechanisms require the shared components to have compatible parameter shapes. When clients use navigation models with different hidden or embedding dimensions, the corresponding native parameters cannot be directly averaged.

**Heterogeneous Model Aggregation.** Several federated learning methods accommodate heterogeneous models by establishing correspondence between model components or transferring knowledge through predictions. FedMA matches hidden units before aggregating model parameters (Wang et al., 2020), while HeteroFL, FedRolex, ScaleFL, and FIARSE train differently sized submodels of a shared global network (Diao et al., 2021; Alam et al., 2022; Ilhan et al., 2023; Wu et al., 2024). FedDF instead transfers knowledge by matching ensemble predictions on shared unlabeled or generated server inputs (Lin et al., 2020). Applying prediction matching to VLN requires common navigation inputs and corresponding action meanings. Independently collected trajectories in private environments do not by themselves provide these common inputs, and candidate-action meanings vary across navigation states (Zhou & Wang, 2022). Another line of work connects heterogeneous models through shared representations or auxiliary components, which we discuss next.

**Representation-Based Knowledge Sharing.** FedProto and FedTGP share class prototypes (Tan et al., 2022; Zhang et al., 2024), while FedGH trains a shared prediction head (Yi et al., 2023). FedGen and FedCG use generators to transfer knowledge through synthetic features (Zhu et al., 2021; Wu et al., 2022), and GeFL-F combines a common feature extractor with feature generation (Kang et al., 2026). Closely related to our approach, FedMRL connects a shared auxiliary model to heterogeneous local models through private projectors and trains fused representations of different dimensions for classification (Yi et al., 2024). These methods organize exchanged knowledge around globally consistent class labels. In VLN, action indices refer to candidate destinations whose meanings vary across navigation states (Zhou & Wang, 2022). Their class-based sharing objectives therefore do not directly apply to VLN. To address this mismatch, FedHVLN shares language-module parameters rather than an action-classification head. Client-specific projections feed the resulting instruction features into each native navigator, which combines them with visual observations and navigation history to select actions. Local navigation losses train the shared language modules, allowing clients to contribute navigation knowledge without requiring their action indices to have the same meanings across environments.

# 3 FEDHVLN

FedHVLN lets clients with structurally different vision-language navigation (VLN) models learn through compatible language-side parameters. Figure 2 shows four training stages and the real-instruction inference path. We define the local navigation task and the interface between shared and native modules, then describe the data, optimization, and communication in each stage.

## 3.1 PROBLEM SETUP

Let $\mathcal{K}$ be the client set and $s(k)$ the native model configuration used by client $k$. Its parameters $\theta_k = (\theta_k^{\mathrm{vis}}, \theta_k^{\mathrm{lang}}, \theta_k^{\mathrm{dec}})$ comprise a visual module, a language module, and a VLN action decoder. At step $t$, the local policy scores the candidate viewpoints $\mathcal{A}_{k,t}$ using the instruction, observation history $o_{1:t}$, and previous actions $a_{<t}$. The candidate set and the meaning of an action index depend on the current state. Native hidden and embedding widths can differ across configurations, so the corresponding $\theta_k$ tensors cannot be averaged across clients.

Client $k$ retains local route and scene context $\mathcal{R}_k$. Annotated clients $\mathcal{K}_{\mathrm{hum}} \subseteq \mathcal{K}$ also have paired examples $\mathcal{D}_k^{\mathrm{hum}} = \{(c, \tau, x)\}$, where $c$ is the local scene-graph or route context, $\tau$ is the route, and $x$ is its human instruction. Other local routes need not have a human instruction. The objective is to improve each client's instruction-following policy using these local data while exchanging only compatible model parameters. Instructions, scene graphs, observations, routes, and action predictions remain on their clients.

## 3.2 LOCAL NAVIGATOR AND SHARED LANGUAGE INTERFACE

The shared modules are an auxiliary instruction encoder $E$, a language transformation $D$, a Speaker $S$, and a feature generator $G$. The transformation $D$ is distinct from the native VLN action decoder. $E$ and $G$ produce contextual instruction features with common width $d_c$, and $D$ transforms either feature sequence. For client $k$, a local projection $P_k$ maps the output of $D$ to the width required by its native language module. We include $P_k$ in $\theta_k^{\mathrm{lang}}$: it is trained with the local navigator and is never uploaded. Thus one common $D$ can provide auxiliary context to language modules with different native widths.

For a human instruction $x$, the auxiliary path is $x \to E(x) \to D(E(x)) \to P_k(D(E(x)))$. A pseudo-instruction $\tilde{x} = S(c)$ is text generated by the Speaker from local scene–route context $c$; its path is $\tilde{x} \to G(\tilde{x}) \to D(G(\tilde{x})) \to P_k(D(G(\tilde{x})))$. In either case, the projected features enter the native language module alongside the instruction. The local visual module processes observations, and the native VLN decoder combines visual and language states with navigation history to choose the next viewpoint. No cross-client action predictions are matched.

## 3.3 STAGE 1: HUMAN-INSTRUCTION NAVIGATION

For $(c, \tau, x) \in \mathcal{D}_k^{\mathrm{hum}}$, the human instruction enters both the native language module and $E$. The transformed, locally projected $D(E(x))$ supplies auxiliary language context while the client navigates along $\tau$. Teacher-forced imitation, sampled reinforcement learning, and non-padding token prediction jointly train $(\theta_k, E, D)$. We use the same loss form for the human and later pseudo-instruction navigation stages:

$$\mathcal{L}_{k,\mathrm{nav}}^{(q)} = \lambda_q \mathcal{L}_{k,\mathrm{ML}}^{(q)} + \mathcal{L}_{k,\mathrm{RL}}^{(q)} + 0.05\mathcal{L}_{k,\mathrm{tok}}^{(q)}, \qquad (\lambda_1, \lambda_3) = (0.2, 0.6), \tag{1}$$

where $\mathcal{L}_{\mathrm{ML}}$ is the negative log-likelihood of reference navigation actions, $\mathcal{L}_{\mathrm{RL}}$ is the sampled navigation policy loss, and $\mathcal{L}_{\mathrm{tok}}$ is cross-entropy on non-padding instruction tokens. The index $q = 1$ denotes human-instruction batches and $q = 3$ denotes pseudo-instruction–route batches. The local projection $P_k$, as part of $\theta_k$, receives navigation gradients. After local training, clients upload updates to $E$ and $D$ for aggregation, retaining every native update locally. The resulting $E$ supplies the fixed human-feature target in Stage 2.2, while $D$ initializes the later pseudo-instruction navigation stage.

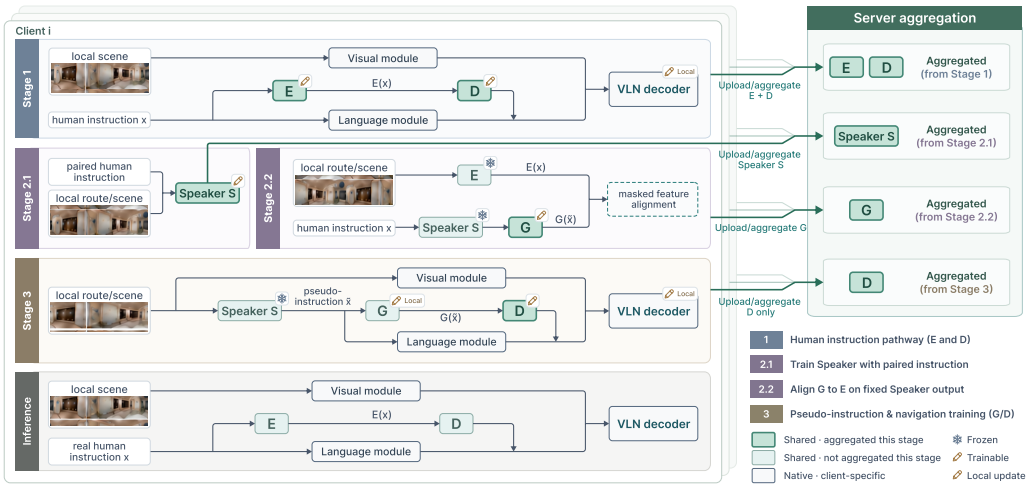


Figure 2: Four-stage FedHVLN training and inference. Stage 1 updates the local VLN, $E$, and $D$, uploading $E$ and $D$. Stage 2.1 trains and aggregates the Speaker on local scene–instruction pairs. In Stage 2.2, a fixed Speaker describes a local scene or route; $G$ processes its pseudo-instruction and is trained toward the fixed $E$ representation of the paired human instruction, with only $G$ uploaded. Stage 3 updates the local VLN, $G$, and $D$ on pseudo-instruction–route pairs, uploading only $D$. The output of $D$ is projected locally to the native language width. Arrows to the server mark uploads and aggregation; inference uses the human $E \to D$ path.

### 3.4 Stage 2.1: Federated Speaker Training

The Speaker learns to describe local route context before any pseudo-instruction supervises navigation. Each annotated client trains $S$ on $(c, x)$ from its own $\mathcal{D}_k^{\text{hum}}$, using the human instruction as the sequence-generation target. With $x_t$ denoting its $t$th token, the local Speaker objective is

$$\mathcal{L}_{k,\text{spk}} = -\sum_{(c,\tau,x)\in\mathcal{D}_k^{\text{hum}}} \sum_{t=1}^{|x|} \log p_S(x_t \mid x_{<t}, c). \tag{2}$$

Clients upload only $S$ for aggregation, then freeze the common Speaker. It supplies local route-conditioned pseudo-instructions for Stages 2.2 and 3; training pairs and generated text remain local.

### 3.5 Stage 2.2: Aligning Generated and Human Features

For an annotated local pair $(c, \tau, x)$, the fixed Speaker receives the scene or route context $c$ and produces a pseudo-instruction $\tilde{x} = S(c)$. The trainable generator processes $\tilde{x}$, whereas the fixed human-instruction encoder processes its paired human instruction $x$. Thus the target is $E(x)$, not $E(\tilde{x})$. Let $E(x) \in \mathbb{R}^{T_x \times d_c}$ and $G(\tilde{x}) \in \mathbb{R}^{T_{\tilde{x}} \times d_c}$ be their high-dimensional deep feature sequences, and let $M_{x,t}$ and $M_{\tilde{x},u}$ mark non-padding positions. We pool the two sequences separately:

$$z_E(x) = \frac{\sum_t M_{x,t} E(x)_t}{\sum_t M_{x,t}}, \qquad z_G(\tilde{x}) = \frac{\sum_u M_{\tilde{x},u} G(\tilde{x})_u}{\sum_u M_{\tilde{x},u}}. \tag{3}$$

The local alignment loss is

$$\mathcal{L}_{k,\text{align}} = \frac{1}{d_c} \left\| z_G(\tilde{x}) - \text{stopgrad}(z_E(x)) \right\|_2^2. \tag{4}$$

Pooling yields two fixed-width vectors even if the human and pseudo-instructions have different lengths; it does not assume token-by-token correspondence. $S$ and $E$ stay fixed, and only $G$ is optimized and uploaded for aggregation. The common $G$ initializes each client's generator in Stage 3, where navigation supervision can further adapt its token-level features locally.

### 3.6 Stage 3: Pseudo-Instruction Navigation

Each client samples routes from its own environment. For a local route context $c$ and route $\tau$, the frozen Speaker generates $\tilde{x} = S(c)$. The pseudo-instruction enters the native language module and $G$; $D(G(\tilde{x}))$ is projected by $P_k$ into the local language width. The route provides teacher-forced and sampled navigation supervision through Eq. 1. The client updates $\theta_k$, its locally retained $G_k$, and $D$, while $E$ and $S$ remain fixed. Only the $D$ update is uploaded and aggregated. Stage 3 therefore begins with the common $G$ learned in Stage 2.2, refines it for each local navigator, and trains $D$ on Speaker-conditioned navigation without exchanging the native VLN or the refined $G_k$.

### 3.7 Federated Updates and Inference

Let $\mathcal{C}_r^{(q)}$ be the clients participating in round $r$ of stage $q$, and let $n_{k,r}^{(q)}$ be the number of local training examples contributing to client $k$'s update in that round. For a module $M$ designated for upload in the stage, the server applies a local-example-weighted parameter average,

$$M^{r+1} = \sum_{k \in \mathcal{C}_r^{(q)}} \frac{n_{k,r}^{(q)}}{\sum_{j \in \mathcal{C}_r^{(q)}} n_{j,r}^{(q)}} M_k^{r+1}. \tag{5}$$

The uploaded set is $\{E, D\}$ in Stage 1, $\{S\}$ in Stage 2.1, $\{G\}$ in Stage 2.2, and $\{D\}$ in Stage 3. The server sends each aggregated module to participating clients for the next round or stage. Native $\theta_k$, including $P_k$, and the Stage 3-refined $G_k$ remain local. Algorithm 1 states the complete update and communication order.

**Algorithm 1: Four-stage federated training of FedHVLN**

**Input:** clients $\mathcal{K}$, annotated pairs $\{\mathcal{D}_k^{\text{hum}}\}$, local routes $\{\mathcal{R}_k\}$, native models $\{\theta_k\}$, shared $E, D, S, G$, and rounds $R_1, R_{2.1}, R_{2.2}, R_3$.
**Stage 1 – human-instruction navigation**
**for** $r = 1, \ldots, R_1$: broadcast $(E, D)$; each participating annotated client samples $(c, \tau, x)$, updates $(\theta_k, E, D)$ with Eq. 1, and uploads $(E, D)$; aggregate both using Eq. 5.
**Stage 2.1 – Speaker**
**for** $r = 1, \ldots, R_{2.1}$: broadcast $S$; each participating annotated client updates $S$ on $(c, x)$ with Eq. 2 and uploads $S$; aggregate $S$. Freeze the resulting Speaker.
**Stage 2.2 – feature alignment**
**for** $r = 1, \ldots, R_{2.2}$: broadcast $G$; for each participating annotated client, obtain $\tilde{x} = S(c)$, $z_E(x)$, and $z_G(\tilde{x})$ with fixed $(S, E)$, update only $G$ using Eq. 4, and upload $G$; aggregate $G$.
**Stage 3 – pseudo-instruction navigation**
Initialize each $G_k$ from aggregated $G$ and keep $(S, E)$ fixed.
**for** $r = 1, \ldots, R_3$: broadcast $D$; each participating client samples local $(c, \tau)$, generates $\tilde{x} = S(c)$, updates $(\theta_k, G_k, D)$ with Eq. 1, and uploads only $D$; aggregate $D$ and retain $(\theta_k, G_k)$ locally.
**Output/inference:** for real human $x$, use local $\theta_k$ and $P_k(D(E(x)))$; perform no upload.

At inference, the real instruction follows $x \to E(x) \to D(E(x)) \to P_k(D(E(x)))$ into client $k$'s adapted language module. Its visual module and native VLN decoder produce actions from local observations and navigation history. The Speaker and $G_k$ are training-only; inference requires neither route generation nor communication with the server.

## 4 Experiments

We organize our evaluation around three objectives. **1) Heterogeneous aggregation.** We test whether FedHVLN can aggregate knowledge across navigators with incompatible native structures. **2) Heterogeneous versus homogeneous performance.** We compare its navigation performance in mixed and homogeneous federations. **3) Homogeneous method comparison.** We compare FedHVLN with an existing federated VLN aggregation method when all clients use the same native structure.

### 4.1 Experimental Setup

We evaluate on R2R (Anderson et al., 2018), treating its 61 training environments (14,039 human instructions) as clients. The `val_seen` validation set has 1,021 held-out instructions from 56 of

these environments. Averaged over instructions, we report oracle success rate (OSR) over visited viewpoints, final-viewpoint success rate (SR), success weighted by path length (SPL), coverage weighted by length score (CLS), and normalized dynamic time warping (nDTW). Both models use the same Seq2Seq Listener topology, 512-dimensional visual features, and 128-dimensional angle features. Configuration A (wide) has recurrent and word-embedding widths of 512 and 256; configuration B (compact) uses 256 and 128, respectively, yielding incompatible native tensors. Before observing outcomes, we assign 41 clients to the wide model and 20 to the compact model. Their validation subsets contain 652 instructions in 37 environments and 369 in 19, respectively.

Each round samples 12 distinct clients without replacement under seed 51701, using the same schedule in all matched runs. A selected client uses batch size 64 for $\lceil 3n_k/(2 \times 64) \rceil$ paired human- and pseudo-instruction updates, where $n_k$ is its number of human training instructions. In the *cross-group* run, the shared interface aggregates across both model sizes while native parameters aggregate only within a size. The two *within-group controls* replay the first 991 rounds, retaining only selected clients of the evaluated size and aggregating their interface updates within that size. Approximately 8 wide-model and 4 compact-model clients participate per round: their controls contain 7,993/3,899 client visits and 44,146/25,864 paired updates, respectively. Together these recover all 12 sampled clients and 70,010 paired updates of the matched cross-group run. Controls use fixed terminal checkpoints. Full-population mixed and homogeneous runs use the same CLIP features, locally sampled augmentation routes, frozen Speaker, and training seed; among 15 scheduled checkpoints, they select by validation SPL, then SR, then the earlier update.

### 4.2 R2R Validation Results

Table 1 reports the mixed FedHVLN federation, the homogeneous wide-model and compact-model FedHVLN federations, and their FedVLN baselines.

**Table 1:** R2R validation results over all 1,021 instructions. Mixed combines wide-model and compact-model clients; each homogeneous row uses one model size. All metrics are percentages; higher is better.

| Config. | Method | OSR ↑ | SPL ↑ | SR ↑ | CLS ↑ | nDTW ↑ |
|---|---|---|---|---|---|---|
| Mixed | FedHVLN | 71.205 | 61.690 | 64.447 | 72.288 | 62.318 |
| Wide | FedHVLN | 74.437 | 63.019 | 66.895 | 71.816 | 61.378 |
| Wide | FedVLN | 71.988 | 61.527 | 65.524 | 69.235 | 59.021 |
| Compact | FedHVLN | 72.478 | 62.331 | 65.328 | 72.576 | 62.594 |
| Compact | FedVLN | 70.029 | 59.590 | 63.173 | 69.704 | 58.943 |

Table 2 compares cross-group and within-group training at fixed round 991 on identical validation subsets. Evaluated clients retain their instructions, initialization, sampling history, and paired-update budget; only the cross-group run receives interface updates from the other model size.

**Table 2:** Matched round-991 R2R validation. Cross-group uses 41 wide-model and 20 compact-model clients; each within-group control uses only the evaluated size. Differences are cross-group minus within-group in percentage points; higher is better.

| Training scope | Evaluated subset | OSR ↑ | SPL ↑ | SR ↑ | CLS ↑ | nDTW ↑ |
|---|---|---|---|---|---|---|
| Cross-group (61) | Wide (652) | 72.699 | 62.218 | 65.184 | 73.554 | 63.730 |
| Within-group (41) | Wide (652) | 70.552 | 58.974 | 62.117 | 72.113 | 61.382 |
| Difference (pp) | | +2.147 | +3.244 | +3.067 | +1.441 | +2.349 |
| Cross-group (61) | Compact (369) | 68.564 | 60.758 | 63.144 | 70.052 | 59.823 |
| Within-group (20) | Compact (369) | 65.854 | 57.567 | 59.892 | 69.229 | 58.961 |
| Difference (pp) | | +2.710 | +3.191 | +3.252 | +0.823 | +0.862 |

**Heterogeneous versus homogeneous performance.** On the same 1,021-instruction validation set, homogeneous wide-model FedHVLN exceeds the mixed federation by 2.45 percentage points in SR and 1.33 in SPL; homogeneous compact-model FedHVLN exceeds it by 0.88 and 0.64 points, respectively. The mixed run therefore does not show an accuracy advantage over either homogeneous run. Because the federation's model composition also changes, these differences do not isolate the effect of cross-structure aggregation.

**Homogeneous method comparison.** FedHVLN exceeds FedVLN on all five metrics with both model sizes (Table 1). Wide and compact SR/SPL gains are 1.37/1.49 and 2.15/2.74 points, respec-

tively; OSR, CLS, and nDTW also improve, indicating better endpoint success and reference-route agreement. Figure 3 shows these ten point differences with paired-scene intervals.

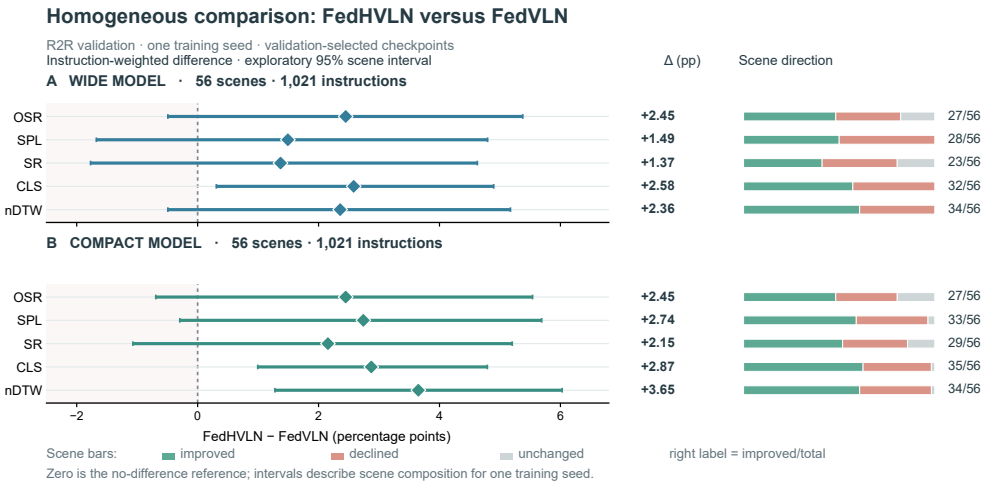


Figure 3: **Homogeneous FedHVLN versus FedVLN on the Table 1 validation set.** Panels A and B show the wide and compact models. Diamonds and whiskers are the instruction-weighted full-set difference and existing exploratory 95% interval from 20,000 paired resamples of 56 validation environments. Scene bars count environments with positive, negative, and tied differences; the printed fraction gives the positive count. Each run uses one seed and its validation-selected checkpoint; intervals do not adjust for checkpoint selection or multiple comparisons. The mixed row has no matched FedVLN comparator.

The wide-model CLS interval and the compact-model CLS and nDTW intervals lie above zero; the other seven include zero. Supplementary Figure S2 shows the scene-level SPL and SR changes.

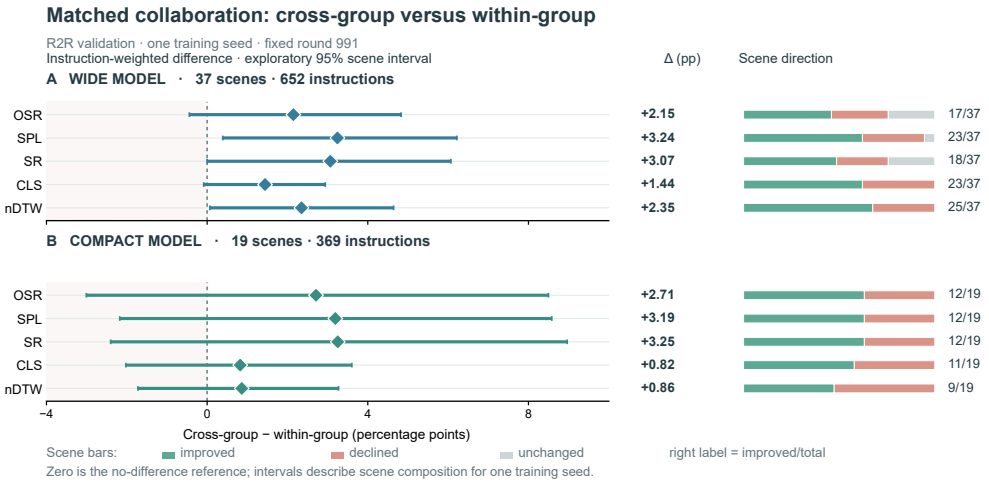


Figure 4: **Matched cross-group versus within-group comparison in Table 2.** Panels A and B show the wide (37 environments) and compact (19 environments) validation subsets. Diamonds and whiskers are full-subset instruction-weighted differences and existing exploratory 95% intervals from 20,000 paired environment resamples within each subset. Scene bars show positive, negative, and tied environment differences, with the positive count printed. Comparisons use one seed and fixed round-991 checkpoints; intervals are unadjusted across ten comparisons.

**Cross-structure collaboration.** Table 2 shows positive cross-group differences on all five metrics for both evaluated model sizes. The wide-model subset gains 3.07 SR and 3.24 SPL points, and the compact-model subset gains 3.25 SR and 3.19 SPL points; pooled over all 1,021 validation instructions, the gains are 3.14 SR and 3.23 SPL points. Each subset retains its own instructions, initialization, sampling history, and paired-update budget across the two rows. These matched validation comparisons show higher scores when the compatible language interface is aggregated across structures while native parameters remain within their structural groups.

Scene-level SPL/SR changes appear in Supplementary Figure S3.

Figure 4 gives exploratory scene-resampling intervals for the five differences in Table 2. The wide-model SPL and nDTW intervals lie above zero; all five compact-model intervals include zero despite positive point estimates.

Figure 5 shows checkpoint-binned training losses for the five runs in Table 1. All five traces decrease, but these within-run trends do not compare navigation accuracy.

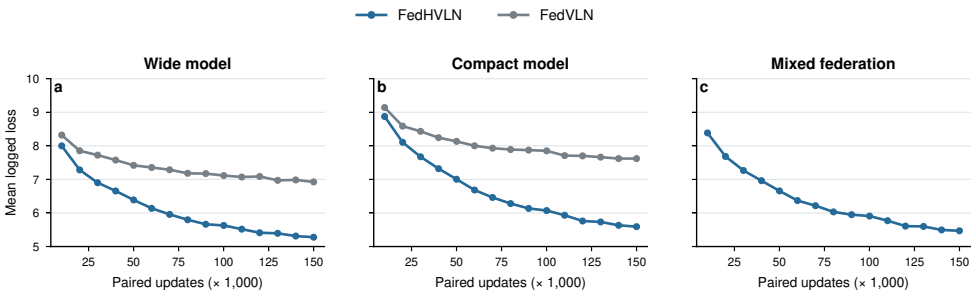


Figure 5: **Logged training loss for the five runs in Table 1.** Panels a–c show the wide-model, compact-model, and mixed federations. Blue: FedHVLN; gray: FedVLN. Each point averages client-completion training losses within the interval ending at one of 15 scheduled checkpoints. Loss is not a validation metric or a common-scale comparison between methods.

## 5 Limitations

Our heterogeneity evaluation uses two configurations of the same Seq2Seq Listener topology with identical visual features; only the recurrent and word-embedding widths differ. The results demonstrate collaboration across these two incompatible native parameter shapes, but do not establish that the approach extends to different navigator architectures or visual encoders. We have not measured communication volume, computational cost, or memory use, so the motivation of serving devices with different resource budgets remains unverified in deployment.

## 6 Conclusion

FedHVLN enables structurally different VLN models to participate in one federation without averaging incompatible native parameters. Its four training stages aggregate compatible language-side modules while each client's native navigator remains local. The human-instruction encoder and shared transformation are reused at inference after pseudo-instruction training. On the 1,021-instruction R2R validation set, the mixed federation reaches 64.45% SR and 61.69% SPL.

At the matched round-991 endpoint, cross-group aggregation yields pooled gains of 3.14 SR and 3.23 SPL points over the within-group controls while preserving each evaluated subset's local data and update budget. Point differences favor cross-group aggregation on all five metrics for both model sizes. In separate homogeneous comparisons, FedHVLN improves all five reported metrics over FedVLN, with SR/SPL gains of 1.37/1.49 points for the wide model and 2.15/2.74 for the compact model. These findings support cross-structure collaboration for the two tested configurations on the R2R validation set.

### REPRODUCIBILITY STATEMENT

Section 3 specifies the four training stages, the client-side updates, the shared-module aggregation rule, and the inference path. Section 4 reports the R2R data split, model widths, client sampling and update budgets, training seed, checkpoint selection, evaluation metrics, and matched-control comparisons. The separate supplementary material states the assumptions and proofs for the conditional shared-interface analysis.

### AI USE STATEMENT

We used generative AI tools to assist with polishing the manuscript, including improving wording and readability. The authors reviewed and revised the AI-assisted text, verified the final claims and references, and take responsibility for the entire submission.

## REFERENCES

Samiul Alam, Luyang Liu, Ming Yan, and Mi Zhang. FedRolex: Model-heterogeneous federated learning with rolling sub-model extraction. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-2152. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/bf5311df07f3efce97471921e6d2f159-Abstract-Conference.html.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3674–3683, 2018.

Enmao Diao, Jie Ding, and Vahid Tarokh. HeteroFL: Computation and communication efficient federated learning for heterogeneous clients. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2010.01264.

Fatih Ilhan, Gong Su, and Ling Liu. ScaleFL: Resource-adaptive federated learning with heterogeneous clients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24532–24541, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Ilhan_ScaleFL_Resource-Adaptive_Federated_Learning_With_Heterogeneous_Clients_CVPR_2023_paper.html.

Honggu Kang, Seohyeon Cha, and Joonhyuk Kang. GeFL: Model-agnostic federated learning with generative models. *IEEE Transactions on Mobile Computing*, 25(4):4493–4505, 2026. doi: 10.1109/TMC.2025.3592483. URL https://ieeexplore.ieee.org/document/11095422/.

Tao Lin, Lingjing Kong, Sebastian U. Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/18df51b97ccd68128e994804f3eccc87-Abstract.html.

Wenda Qin, Andrea Burns, Bryan A. Plummer, and Margrit Betke. Walk and read less: Improving the efficiency of vision-and-language navigation via tuning-free multimodal token pruning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 23556–23570. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1202. URL https://aclanthology.org/2025.emnlp-main.1202/.

Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. FedProto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8432–8440, 2022. doi: 10.1609/aaai.v36i8.20819. URL https://ojs.aaai.org/index.php/AAAI/article/view/20819.

Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BkluqlSFDS.

Feijie Wu, Xingchen Wang, Yaqing Wang, Tianci Liu, Lu Su, and Jing Gao. FIARSE: Model-heterogeneous federated learning via importance-aware submodel extraction. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-3672. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/d17477a5ff8b7ddb49e53f9c04305da5-Abstract-Conference.html.

Yuezhou Wu, Yan Kang, Jiahuan Luo, Yuanqin He, Lixin Fan, Rong Pan, and Qiang Yang. FedCG: Leverage conditional GAN for protecting privacy and maintaining competitive performance in federated learning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pp. 2334–2340, 2022. doi: 10.24963/ijcai.2022/324. URL https://www.ijcai.org/proceedings/2022/324.

Qingqian Yang, Hao Wang, Sai Qian Zhang, Jian Li, Yang Hua, Miao Pan, Tao Song, Zhengwei Qi, and Haibing Guan. FedNASP: Federated vision-language navigation with adaptive step-wise personalization. In *Computer Vision – ECCV 2026*, volume 17062 of *Lecture Notes in Computer Science*, pp. 261–277. Springer, 2026a. doi: 10.1007/978-3-032-37098-3_15. URL https://link.springer.com/chapter/10.1007/978-3-032-37098-3_15.

Qingqian Yang, Hao Wang, Sai Qian Zhang, Jian Li, Yang Hua, Miao Pan, Tao Song, Zhengwei Qi, and Haibing Guan. pFedNavi: Structure-aware personalized federated vision-language navigation for embodied AI. In *IEEE INFOCOM Workshop on Embodied Intelligence Networks (EIN)*, 2026b. URL https://intellisys.haow.us/assets/pdf/1571235278-final.pdf.

Liping Yi, Gang Wang, Xiaoguang Liu, Zhuan Shi, and Han Yu. FedGH: Heterogeneous federated learning with generalized global header. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 8686–8696, 2023. doi: 10.1145/3581783.3611781. URL https://arxiv.org/abs/2303.13137.

Liping Yi, Han Yu, Chao Ren, Gang Wang, Xiaoguang Liu, and Xiaoxiao Li. Federated model heterogeneous matryoshka representation learning. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2125. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/7a9ee756fb71d401bcb5230db02c11db-Abstract-Conference.html.

Jianqing Zhang, Yang Liu, Yang Hua, and Jian Cao. FedTGP: Trainable global prototypes with adaptive-margin-enhanced contrastive learning for data and model heterogeneity in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 16768–16776, 2024. doi: 10.1609/aaai.v38i15.29617. URL https://ojs.aaai.org/index.php/AAAI/article/view/29617.

Kaiwen Zhou and Xin Eric Wang. FedVLN: Privacy-preserving federated vision-and-language navigation. In *Computer Vision – ECCV 2022*, pp. 682–699. Springer, 2022. doi: 10.1007/978-3-031-20059-5_39.

Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12878–12889. PMLR, 2021. URL https://proceedings.mlr.press/v139/zhu21b.html.