Semantics-Aware Avatar Locomotion Adaption for Indoor Cross-Scene AR Telepresence

Yi-Jun Li, *Member, IEEE*, Hao-Zhong Yang, Wen-Tong Shu, and Miao Wang, *Member, IEEE*

***Abstract*—Geographically dispersed users often rely on virtual avatars as intermediaries to facilitate interactive communication and collaboration. However, existing methods for augmented reality (AR) telepresence applications exhibit limitations, including restricted movement within confined sub-areas, lack of smooth transitions, and the necessity for manually establishing object mapping between dissimilar environments. We present a novel interactive AR framework for virtual avatar locomotion adaption while preserving semantic coherence across dissimilar indoor scenes. Initially, we conduct a preliminary user study to identify key attributes influencing preferred avatar movement. These attributes are quantified as features, and a dataset of user annotations on avatar movements is created. Based on the user interaction and scene configurations, we employ a deep reinforcement learning neural network to guide the avatar to the ideal position while maximizing semantic coherence. We validate our proposed framework through simulations and user studies by implementing an AR-based 3D telepresence prototype, demonstrating the efficacy of our framework in conveying user intentions across dissimilar environments, enabling natural and immersive 3D telepresence interactions.**

***Index Terms*—Augmented reality, heterogeneous environments, reinforcement learning, telepresence.**

I. Introduction

AUGMENTED reality (AR) technology integrates computer-generated elements with the physical world, enhancing interactive experiences [1]. Recent advancements in AR have propelled the increasing popularity of AR applications across diverse domains, including education, storytelling, and remote telepresence [2], [3], [4]. For remote telepresence, geographically dispersed users can communicate or collaborate through virtual avatars overlaid on their local physical environments, fostering an immersive experience by creating the illusion of physical co-presence with remote participants [5], [6].

While directly mapping a user's movements onto the corresponding avatar is viable in identical room configurations [6], [7], a key challenge arises when room geometries and layouts differ between users. Such disparities can lead to issues such as avatars penetrating furniture or interacting with unintended environmental objects, degrading the sense of co-presence during multi-user remote telepresence [8], [9].

Existing solutions to address this challenge can be broadly categorized into two approaches. The first category aims to identify a mutual sub-region within different rooms, enabling users to freely interact within a small common space [10], [11], [12], [13], [14]. For instance, Keshavarzi et al. [10] proposed an optimization method based on the input semantic scene maps to find the maximal mutual space for multi-user immersive interaction. While facilitating multi-user telepresence, this approach confines user interaction to a limited common region, failing to utilize the full spatial extent of the rooms.

The other category of methods employs retargeting techniques to map user positions or movements to avatars, enabling remote users to navigate freely throughout the entire room [15], [16]. For instance, Yoon et al. [15] introduced an adaptive learning-based method that teleports avatars to optimal positions while preserving semantics of remote users and furniture. However, this placement method lacks movement consistency. Wang et al. [16] predict the user's target then guide the avatar towards the corresponding target using an established artificial potential field (APF) [17]. While this method preserves smooth movement transitions for the user, it requires manually establishing object mappings between rooms.

In this paper, we propose a novel AR framework to facilitate immersive remote telepresence in daily scenarios (shown in Fig. 1). Our locomotion adaption approach does not necessitate manual object mapping, and the movement of the full-body lifelike avatar automatically adapts to the remote user's interactions and the scene semantics. We first conduct a preliminary user study to gather annotated data and user feedback. Our analysis reveals that room layouts can be classified into several semantic parts. Based on the definition of this avatar locomotion adaption problem and established dataset, we then propose a framework

Received 26 July 2024; revised 5 December 2024; accepted 29 December 2024. Date of publication 3 January 2025; date of current version 5 September 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62372025 and Grant 62361146854 and in part by the Fundamental Research Funds for the Central Universities. Recommended for acceptance by S. Lee. *(Corresponding author: Miao Wang.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Ethics Committee of Beihang University under Application No. 2024-012, and performed in line with the Declaration of Helsinki.

Yi-Jun Li is with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China, and also with Beihang University, Beijing 100191, China (e-mail: liyijun@mail.tsinghua.edu.cn).

Hao-Zhong Yang and Wen-Tong Shu are with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing 100191, China (e-mail: haozhongYang@buaa.edu.cn; wentongshu@buaa.edu.cn).

Miao Wang is with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing 100191, China, and also with Zhongguancun Laboratory, Beijing 100094, China (e-mail: miaow@buaa.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TVCG.2025.3525697, provided by the authors.

Digital Object Identifier 10.1109/TVCG.2025.3525697

Fig. 1. Third-person perspectives of our cross-scene avatar locomotion adaption method in heterogeneous environments. Room A (Left) and Room B (Right) depict two remote, semantically different physical spaces. User A and User B, represented by their respective avatars (Avatar A and Avatar B), can move and interact across these spaces. The arrows indicate user movements and corresponding avatar transitions. For instance, as User A (left) moves within Room A, Avatar B adaptively transitions to the semantically equivalent position in Room B, maintaining semantic coherence despite the rooms' heterogeneity.

leveraging reinforcement learning to control real-time avatar movement that adhere to user interactions and coherence with scene semantics. While some works generate detailed human poses or motions by using data-driven models [18], [19], [20], our work plans symbolic character interactions before assigning predefined animation clips to different avatar behaviors.

Our main contributions include the following:

- Defining the avatar locomotion adaption problem in remote telepresence and analyzing its challenges based on our preliminary user study.
- Proposing a framework containing a user target predictor and a reinforcement learning mapper, which automatically controls avatar movements guided by user interactions and scene semantics from our annotated dataset.
- Evaluating accuracy and effectiveness of our approach through both virtual simulations and real user studies, and providing suggestions for future improvements.

II. Related Work

A. AR Telepresence in Dissimilar Spaces

The advent of consumer-grade AR and virtual reality (VR) head-mounted displays (HMDs) has brought telepresence experiences into daily life [4], [21]. Early AR telepresence systems enabled remote users to be teleported to fixed positions within the local host's physical space through real-time 3D video capturing and reconstruction, such as directly projecting the user's image onto a local sofa [22] or rendering the remote user on an optical see-through HMD [5]. However, the free movement of the remote user is not supported by these systems. Orts-Escolano et al. [6] developed a telepresence system to reconstruct realistic people, furniture and objects in real-time, but it could only capture a designated local space and required expensive hardware.

Besides relying on high-fidelity capturing and reconstruction, other systems such as Microsoft Mesh [23] and Meta's Horizon Workrooms [24] utilize virtual avatars to convey remote users' behaviors. Choi et al. [25] also proposed a framework that adapts full-body motions and interaction behaviors specifically for morphologically similar remote environments. However, due to the significant discrepancies of furniture layouts and space sizes, directly mapping a remote user's motion to their local avatar would cause penetration with the physical environments or interaction with unintended objects. These issues can conflict with the user's perception in the real world, diminishing the sense of presence and disrupting immersive collaboration.

To address these challenges, some methods aim to find common sub-regions across different scenes or constrain users to specific areas, allowing users to freely interact with each other within shared spaces [10], [11], [26]. For instance, Lehment et al. [11] proposed an automatic alignment scheme by maximizing the shared room common features to create a consensus reality. Keshavarzi et al. [10] converted scenes into topological scene graphs and provided advice on manipulation of ground objects to further maximize the mutual space. Sidenmark et al. [27] proposed an optimization-based method to merge dissimilar physical workspaces for remote collaboration. However, user movements are constrained within a sub mutual area, which could shrink or disappear as the dissimilarity between spaces increases. Moreover, user interactions with physical objects outside the mutual space are not supported during remote telepresence.

Other works share our goal of enabling full-space telepresence for users' daily activities [15], [16], [28], [29], [30]. Yoon et al. [15] trained a deep neural network to relocate avatars to positions that are most similar to the corresponding users' positions. Yang et al. [29] extended the previous method with visual guidance to preserve avatar gaze and pointing alignment. However, these approaches involve sampling the entire room to determine the optimal avatar pose, resulting in delays in real-time telepresence and a lack of natural avatar transitions. Jo et al. [28] utilized rigid transformations from off-line scene matching to identify avatar poses, but their construction of the scene object mapping solely relies on the distribution of furniture directions. Wang et al. [16] employed an APF-based controller to navigate the avatar towards the local target by utilizing a pre-constructed mapping of objects across different rooms. However, this mapping is static and necessitates manual

establishment beforehand. Moreover, their approach primarily focuses on room layouts and overlooks the impact of interactions between users.

B. Avatar Motion Adaption

Natural avatar motion adaption to match given activities and scene layouts is an active research area [19], [31], [32]. For instance, Hassan et al. [19] employed a conditional Variational Auto-Encoder (cVAE) to determine optimal avatar placement based on the 3D scene and human pose. Watkins et al. [32] introduced a pose-preserving optimization framework to adapt avatar motion sequences to different interaction scenarios. However, these approaches emphasize static positioning rather than real-time locomotion adjustment across diverse spaces, limiting their usability to dynamic AR telepresence scenarios.

Some existing methods [33], [34] commonly rely on designated task objectives or action descriptions for avatar motion adaption. For example, Tahara et al. [33] utilized scene graphs to align avatar actions with structured representations of the scene, generating semantically accurate avatar motions. Huang et al. [34] introduced a motion planning framework for generating life-like avatar movements, which considers the observer's viewpoint and the required demonstration task. While these approaches demonstrate promising results in adapting avatar motions to physical environments, they are not optimized for real-time avatar locomotion adaption.

Consequently, there has been a growing focus on addressing challenges where the avatar's motion must dynamically adapt to user interactions and the surrounding environment in real-time, such as in interactive storytelling [3], [35], [36], [37]. For instance, Li et al. [3] introduced a hierarchical story sampling technique to represent events, enabling virtual avatars to adapt behaviors based on user actions and scene context. Furthermore, they [35] also employed a sequential deep graph generative model to generate abstract virtual activity snippets, subsequently assigning appropriate 3D poses and symbolic interaction animations to avatars. Kim et al. [37] proposed a user-centered framework for adapting flying creature movements in outdoor AR storytelling. While these methods achieve dynamic adaption within predefined storylines or specific narratives, they rely on predefined story elements or script constraints, which are less suitable for spontaneous avatar locomotion adaption in AR telepresence scenarios.

C. Agent Control in Reinforcement Learning

Reinforcement learning (RL) has recently emerged as a promising data-driven approach for diverse agent control tasks [38], [39], [40]. It utilizes neural networks as function approximators, enabling policies to scale to high-dimensional state and action spaces and learn mappings from raw sensory inputs to actions without hand-engineered features. The generality across control tasks makes it a compelling and suitable framework for developing intelligent agent behavior via environmental interaction.

Based on given tasks and 3D scenes, methods combined with reinforcement learning have been applied to generate realistic avatar motion, which can be further categorized as kinematic-based methods and physical-based methods [41], [42], [43]. Kinematic-based methods rely on high-quality 3D human-scene data, which is scarce and challenging to acquire. On the other hand, physical-based approaches utilize physics simulators to ensure the physical plausibility of the generated motions. Nevertheless, these approaches necessitate extensive amounts of high-quality 3D human-scene data and require retraining their proposed frameworks with specific reward functions for each task. Moreover, it is essential to account for the complex semantics of indoor scenes compared to simulated environments.

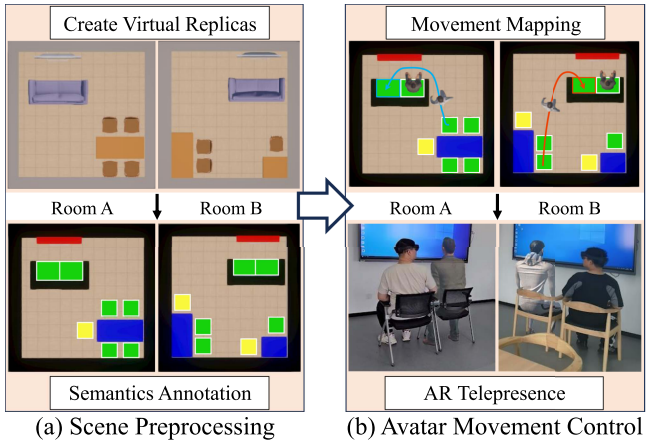


Fig. 2. Overview of our AR cross-scene avatar locomotion adaption method pipeline, which consists of two primary stages. (a) indicates the Scene Preprocessing stage, where virtual replicas of both physical rooms are created manually using 3D furniture models, and scene semantic information is represented by object categories and OBBs. (b) indicates the Avatar Movement Control stage. The predictor utilizes the user's historical trajectory and scene semantics to infer the intended destination. The mapper then determines the avatar's optimal location for real-time avatar navigation, maintaining semantic coherence across dissimilar rooms.

Several studies have explored using reinforcement learning to address the primary challenge of navigating mobile robots in indoor environments [44], [45], [46]. Yang et al. [45] employed a framework to map visual features to discrete motion actions for a single agent engaged in visual semantic navigation. By incorporating a graph convolutional network to extract abstract scene semantics, the agent achieves a higher success rate in locating objects with the given category label and even demonstrates the ability to discover unseen objects in unfamiliar environments. Based on this approach, Liu et al. [46] built a multi-agent framework, enabling the learning of a collaborative strategy among agents to locate multiple target objects. These works present an effective approach to the indoor navigation problem when target objects are provided. However, ambiguity arises when attempting to adapt virtual avatar movements across dissimilar indoor scenes, as the mapping of objects between different rooms remains undefined.

III. Overview

Our approach takes two dissimilar indoor scenes as initial inputs and dynamically maps the user's movements to the corresponding avatar's movements in the other scene, considering

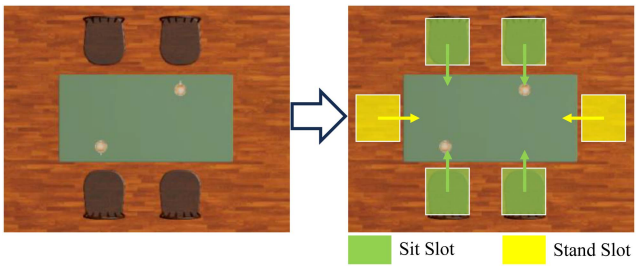


Fig. 3. Discretization of the supporting space into two categories of character slots (sit slots and stand slots).

the space geometry and scene semantics. Fig. 2 illustrates the pipeline, which primarily comprises the following two stages:

Scene Preprocessing: The scene semantics is represented by objects' category labels and their corresponding oriented bounding boxes (OBBs). We create virtual replicas of the physical indoor scenes using existing 3D furniture models and heuristically generate OBBs to represent the scene semantics with a manual refinement. This approach circumvents the need for computationally expensive automatic or semi-automatic methods [3], [33] for scene initialization and annotation.

Avatar Movement Control: The framework consists of two main components: a user target predictor and an avatar target mapper. The predictor utilizes the user's historical trajectory and scene semantics to predict the user's intended destination (see Section V). Based on this prediction and scene semantics, the target mapper determines the optimal location in real-time (see Section VI). The avatar then dynamically navigates to this optimal location.

IV. Scene Semantics and Annotation

A. Scene Semantics Representation

In our framework, we assume that when users are moving, they will finally arrive and stop at a meaningful position, such as a position in proximity to furniture or interactive objects. To simplify the problem, we introduce a discretization approach inspired by Li et al. [3], whereby the spatial regions capable of accommodating users or avatars (denoted as characters) are partitioned into discrete slots. Specifically, for furniture pieces capable of supporting multiple characters simultaneously, such as sofas, we partition the furniture into multiple slots, each assigned the same semantic label. Conversely, for furniture designed to accommodate a single character, such as chairs, no further subdivision is required. Besides, we discretize the spatial regions where a character can stand and interact with specific furniture or objects (such as standing near a table) to maintain consistency in the solution space. As illustrated in Fig. 3, two distinct categories of character slots are defined: stand slots and sit slots. Once a character occupies a slot, it is marked as occupied to prevent overlapping assignments. The partitioning of character slots for both categories is achieved through geometric heuristics that take into account an approximate anthropometric width of a human individual, and the partitioning results will be further refined by a manual process.

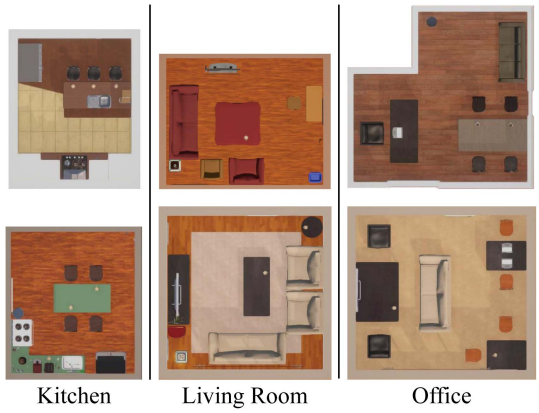


Fig. 4. Top-view illustrations of the room layouts in the dataset, representing three distinct categories: kitchen, living room, and office. Each category includes two unique configurations.

B. Dataset Annotation by User Survey

We conducted a user survey to understand how users prefer avatars to move in relation to their own movements across different scenarios. The methodology involved focusing on three indoor room categories (kitchen, living room, office) and employing 3D models sourced from AI2-THOR [47] to instantiate two indoor room layouts for each category, each containing a realistic assortment of furnishings and functional zones (shown in Fig. 4). We further combined the six room layouts into 12 pairs of scene configurations. For each pair of scene configuration, we constructed 40 questions by situating User A, User B, Avatar A, and Avatar B in different locations. The questions were intentionally designed to encompass a broad spectrum of distances, orientations, interactions (such as watching TV or talking face-to-face), and poses (sit or stand). This methodology culminated in a total of 480 questions (12 pairs $\times$ 40 questions).

For each question, participants selected the optimal target slot for Avatar A based on their preferences. Utilizing 2D floor plans and egocentric views of the user and avatar, we directed participants with the following instruction: "Select the optimal target slot for the avatar that accurately reflects the user's movement in a remote setting." The participants were exclusively presented with scene views and were not provided with any additional details regarding what activities that users were engaged in or observing. This process was aimed to develop an avatar movement control algorithm that did not require high-level contextual cues.

In total, we collected 10 responses from diverse participants for each question, resulting in $480 \times 10 = 4800$ user responses. The survey engaged 21 participants, and each participant responded to a minimum of 100 questions.

Fig. 5 shows two typical samples from the user survey, where the numbers over the slots indicate how many users selected that slot for the question. For example, when two users appeared to be looking at each other or attending to the same object together, participants would place the avatar in similar positions to perform the same action. Conversely, when the room layouts differed significantly due to the distance between the user and

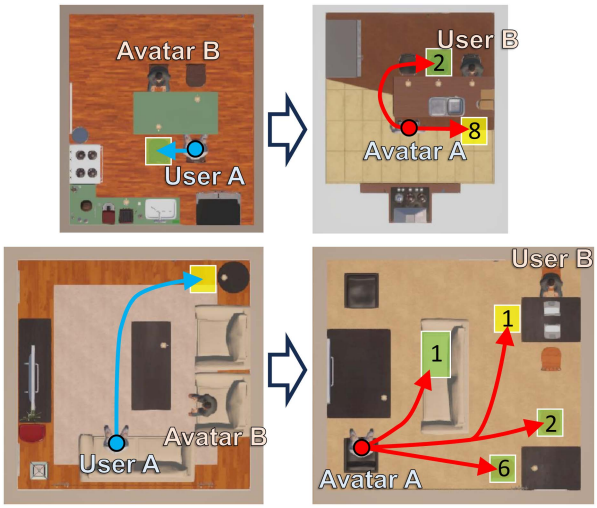


Fig. 5. Illustration of representative samples from the user annotation process. The number over each character slot indicates the vote count among 10 responses for the corresponding question.

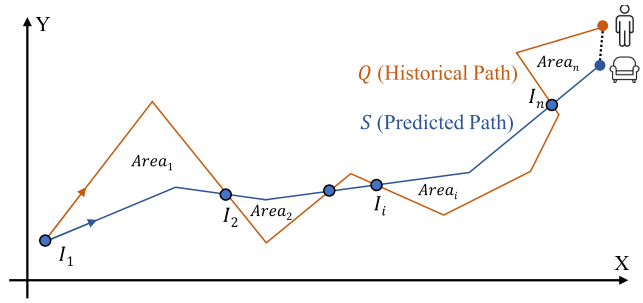


Fig. 6. Illustration of path dissimilarity calculation, which shows the geometric relationship between the historical path Q (blue line) and the predicted path S (orange line). The dissimilarity $D(Q,S)$ is measured based on the areas of polygons $(Area_1, Area_2, \ldots, Area_n)$ formed between the two paths at their intersections $(I_1, I_2, \ldots, I_n)$.

virtual avatar, distinct room layouts, or missing object categories, the participants' annotations were more diverse. Based on user annotation preferences and the semantic furniture in the rooms, OBBs inside the rooms were divided into the following 8 key categories: stand slot, sit slot, table, small item, door, window, screen, and wall. These OBBs represent the geometric semantics of the room for subsequent question solving.

V. User Target Prediction Method

We aim to dynamically predict the user's target location based on their current path and scene context. After obtaining the basic scene representation, including discrete OBBs of objects, we assume that users will avoid obstacles (e.g., tables, cabinets) and choose the path of least cost to reach a target location.

Given a set of candidate target slots, we calculate paths from the user's starting point to these slots. Based on our assumptions, we transform the user's target prediction problem into a path dissimilarity comparison problem. We use a method based on the area formed by the two paths [48] to quantify path dissimilarity.

Let $\mathbf{T}^u = \{T_1^u, T_2^u, \ldots, T_M^u\}$ denote the set of candidate slots for the user, where $P(T_i^u)$ represents the shortest path from the user's starting position to the candidate slot T_i^u. Let H_t^u represents the user's historical path from the starting position to the current position at time step t. The dissimilarity function $D(H_t^u, P(T_i^u))$ is used to measure the dissimilarity between the historical path and the predicted path to candidate slot T_i^u. For clarity, we use Q to represent the historical path H_t^u and S to represent the predicted path $P(T_i^u)$. The dissimilarity is then calculated as:

$$D(Q,S) = \sum_{j=1}^{n} w_j \times Area_j \tag{1}$$

where n is the total number of intersections, $Area_j$ is the area of jth polygon, and w_j is the weight of $Area_j$, which is defined as:

$$w_j = \frac{L_Q(I_j, I_{j+1}) + L_S(I_j, I_{j+1})}{L_Q + L_S} \tag{2}$$

where the numerator represents the perimeter of the polygon, the denominator represents the total length of paths Q and S, and $\{I_1, I_2, \ldots, I_n\}$ is the set of intersection points between paths Q and S, as illustrated in Fig. 6.

The candidate slot T_p^u with the lowest dissimilarity value is chosen as the predicted target slot for the user, where the slot label $p \in \{1, 2, \ldots, M\}$.

VI. Avatar Movement Control Method

After determining the user target, it is essential to navigate the virtual avatar within the room. However, room layouts can vary widely, including differences in the quantity and spatial distribution of furniture objects. To handle these variations and accommodate the user's movement, a dynamic mapping mechanism is required to translate the user's target slot into the optimal avatar target slot. This mapping mechanism consider both scene geometry and semantic information alongside real-time data about the user's movement. By integrating these factors, the avatar is guided to the most appropriate and semantically coherent position while avoiding obstacles.

Let $\mathbf{T}^a = \{T_1^a, T_2^a, \ldots, T_N^a\}$ denote the set of candidate slots for the virtual avatar, where the goal is to find the optimal mapping slot $T_q^a (1 \le q \le N)$. To address this problem, we proposes a reinforcement learning approach based on the Proximal Policy Optimization (PPO) framework. Our method employs a deep neural network as the target mapper for real-time finding of the optimal slot, which is trained on previous user annotation data.

During simulation, the virtual environment updates the user's and avatar's movements at a fixed frame rate. It then passes the user's target slot and the virtual avatar's candidate slot states as inputs to the target mapper. The target mapper returns the label of the current optimal avatar target slot, while the avatar's movement control mechanism adjusts the avatar's speed in real-time based on the distance differences. The avatar gradually approaches the optimal slot along the shortest path, and the virtual environment generates the next state.

The target mapper receives feedback from a reward function to update its model weights. The process continues iteratively until both the virtual avatar and the user reach their respective target slots. In the following sections, we detail the state representation, action space, reward function, and the neural network training process.

A. State and Action Representation

Real-time avatar movement control requires knowledge of the user's target slot and the corresponding slot for the avatar. Let N denote the number of candidate slots in the room where the avatar is located ($N = 20$ in our implementation), slots in the room are numbered from 1 to N. We define the state at each time step t as:

$$S_t = \left[\chi_{t-1}, f^u_{p,t}, \mathbf{F}^a_t\right] \tag{3}$$

$$\mathbf{F}^a_t = \left[f^a_{1,t}, f^a_{2,t}, \ldots, f^a_{N,t}\right] \tag{4}$$

where χ_t represents the mapping slot label feature of the avatar at time step t, encoded using one-hot encoding with N bits. $f^u_{p,t}$ denotes the feature vector of the user's target slot, and $\mathbf{F}^a_t$ represents the feature vectors for the avatar's candidate slots. Specifically, $f^a_{i,t} (1 \le i \le N)$ represents the feature vector of the avatar's ith candidate slot. All feature values are normalized to the range [0,1] for better fitting of the neural network. For cases where the number of slots in the room is less than N, we pad the feature vector with zeros.

For each slot, we define the feature vector from a two-dimensional top-down view to portray the semantic features of the slot at time step t. Based on findings from the previous work [15] and our user annotation process, we further refine these semantic features into lower-level features, as illustrated in Fig. 7.

1) Category Feature: Each slot has a category semantic label, typically represented by one-hot encoding. In our implementation, the category feature f_{ca} is a 2D one-hot encoding: sit slot [1,0] or stand slot [0,1]. This encoding determines the avatar's pose and orientation upon reaching the slot.

2) Relative Pose Feature: This feature $f_{ip,t} = [f^u_{ip,t}, f^a_{ip,t}]$ characterizes the spatial geometric relationships between the slot, the user, and the avatar. Based on Hall's spatial relation theory [49], distance reflects the intimacy between individuals. Besides, the distance relationship reflects the cost required to reach the slot, and the orientation directly impacts the user's visual content. Thus, we define:

$$f^u_{ip,t} = \left[d^u_t, \theta^u_{1,t}, \theta^u_{2,t}\right] \tag{5}$$

where d^u_t is the slot-user distance, $\theta^u_{1,t}$ is the non-negative angle difference between the slot's orientation and the direction to the user, and $\theta^u_{2,t}$ is the non-negative angle difference between the user's orientation and the direction to the slot. $f^a_{ip,t}$ is defined similarly for the avatar.

3) Visual Attention Feature: Users typically focus more on semantic information within their field of view (FoV). We categorize furniture objects that users mainly focus on into 7 categories: sit slot, table, small item, door, window, screen, and wall, assuming that users can only perceive furniture objects within a certain angle and distance range in their FoV. Specifically, the visual attention feature of a category depends on the number of its objects, the proximity of those objects to the user, and the position of the objects relative to the center of the FoV. The visual attention feature $f_{va,i}$ of the ith category in the visual attention feature set $\mathbf{F}_{va}$ is defined as:

$$f_{va,i} = \sum_j \left(\bar{d}_{va} - d_{i,j}\right)\left(\frac{1}{2}\bar{\theta}_{va} - \theta_{i,j}\right) \tag{6}$$

where $d_{i,j}$ and $\theta_{i,j}$ represent the distance and angle between the center of object j in the ith category and the slot, $\bar{d}_{va}$ and $\bar{\theta}_{va}$ represent the maximum distance and angle of the FoV (here $\bar{d}_{va} = 5m$, $\bar{\theta}_{va} = 40°$). For the contribution of objects outside the FoV, it is considered as 0 by default. For larger furniture (tables, windows, doors, screens, walls) where individual size differences can vary significantly, the proportion of the object bounding box in the FoV needs to be considered. To simplify calculations, if k_1 rays are uniformly shot out from the center of the slot at angles ($k_1 = 7$, not coinciding with the boundaries of the FoV), then:

$$f_{va,i} = \frac{1}{k_1}\sum_j \sum_k \left(\bar{d}_{va} - \hat{d}_{k,j}\right)\left(\frac{1}{2}\bar{\theta}_{va} - \hat{\theta}_k\right) \tag{7}$$

where $\hat{d}_{k,j}$ represents the distance between the intersection point of ray k and the bounding box of furniture j and the center of the slot. If there is no intersection point with furniture j or the distance exceeds the maximum distance of the FoV, then $\hat{d}_{k,j} = \bar{d}_{va}$, and $\hat{\theta}_k$ represents the angle between ray k and the orientation of the slot.

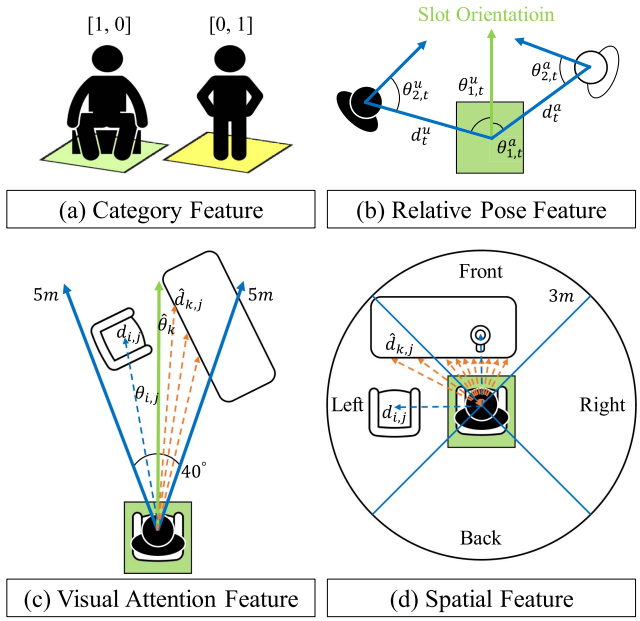


Fig. 7. Semantic features for character slot selection: (a) category feature encoding slot types (sitting [1,0] or standing [0,1]); (b) relative pose feature characterizing spatial relationships among user, avatar, and slot through distances and angles; (c) visual attention feature measuring object-focused attention within a 40° FoV at 5 m range; (d) spatial feature quantifying object distributions in four cardinal directions (front/back/left/right) within a 3 m radius.

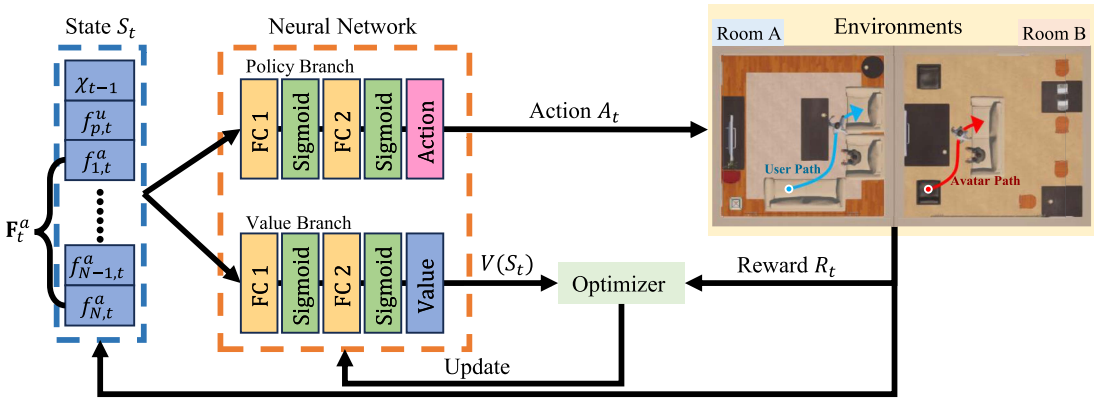


Fig. 8. Training pipeline of our deep reinforcement learning method. At each time step t, the state $S_t = [\chi_{t-1}, f^u_{p,t}, f^a_{1,t}, \ldots, f^a_{N,t}]$ includes the avatar's previous action label χ_{t-1}, the user's target slot feature $f^u_{p,t}$, and the avatar's candidate slot features $\mathbf{F}^a_t = [f^a_{1,t}, f^a_{2,t}, \ldots, f^a_{N,t}]$. This state is fed into the neural network, which comprises a policy branch and a value branch. The policy branch outputs the optimal avatar target slot label as the action A_t, while the value branch predicts the expected reward $V(S_t)$. The avatar navigates to the target slot via shortest path planning, transitioning to the next state S_{t+1} with reward R_t. The optimizer updates the network based on the accumulated rewards and $V(S_t)$. This process is iteratively executed during training.

4) Spatial Feature: Observations suggest that when users select a target slot, they not only consider the semantic information in front of the slot but also take into account the distribution of objects around the slot. Users typically consider whether certain categories of objects are around the slot and then consider the spatial distribution in the front, back, right, and left directions. Thus, the spatial feature $f_{sp,i}$ of the ith category in the spatial feature set $\mathbf{F}_{sp}$ is defined as:

$$f_{sp,i} = [f_{spF,i}, f_{spR,i}, f_{spB,i}, f_{spL,i}, f_{spMax,i}] \tag{8}$$

where $f_{spF,i}$, $f_{spR,i}$, $f_{spB,i}$, $f_{spL,i}$ measure the spatial distribution of objects in front, right, back, and left of the slot, and $f_{spMax,i}$ is the maximum value among them, representing the combined spatial distribution of four directions. Taking the calculation of $f_{spR,i}$ as an example, a 90° sector is established with the slot's right direction as the center, considering the inside object instances belonging to category i. For small-sized objects that can be treated as a whole (e.g., sit slot, small item), the spatial feature for this direction is defined as:

$$f_{spR,i} = \sum_j \left(\bar{d}_{sp} - d_{i,j}\right) \tag{9}$$

where $\bar{d}_{sp}$ is the maximum distance setting for the sector ($\bar{d}_{sp} = 3m$), and $d_{i,j}$ represents the distance of object j in the ith category of furniture relative to the center of the slot, defaults to $\bar{d}_{sp}$ if it exceeds the maximum distance. Similar to the visual attention features, for object categories with significant size variations, if k_2 rays are uniformly shot from the center of the slot at angles ($k_2 = 9$, not coinciding with the boundaries of the sector), then:

$$f_{spR,i} = \frac{1}{k_2} \sum_j \sum_k \left(\bar{d}_{sp} - \hat{d}_{k,j}\right) \tag{10}$$

where $\hat{d}_{k,j}$ represents the distance between the intersection point of ray k and the bounding box of furniture j and the center of the slot. If there is no intersection point with the bounding box of furniture j or the distance exceeds the maximum distance of the sector, then $\hat{d}_{k,j} = \bar{d}_{sp}$.

In summary, $f^u_{p,t} = [f_{ca}, f_{ip,t}, \mathbf{F}_{va}, \mathbf{F}_{sp}]$, and $f^a_{i,t}$ is defined in a similar manner. Since it is assumed that the scene is static except for the characters, the time-independent features f_{ca}, f_{va}, and f_{sp} can be precomputed.

5) Action and Network Architecture: The goal of our network is to map user behavior to avatar slot selection actions. To achieve this, our network (shown in Fig. 8) consists of two branches: the policy function and the value function, both implemented as Multi-Layer Perceptrons (MLPs). In each MLP, there are two fully connected layers with 512 hidden units each, followed by a sigmoid activation function. This structure avoids using complex structures like ResNet or DenseNet for real-time responsiveness.

The policy branch is responsible for selecting the avatar's target slot at each time step t. Given the current state S_t, it outputs an N-dimensional vector representing the probability distribution over candidate slots. The action A_t is determined by the index of the highest probability slot.

The value branch outputs a single scalar value $V(S_t)$ representing the expected cumulative discounted reward and is used as a baseline for decision-making. Both branches work together to refine the action-selection process, where the policy optimizes slot selection, and the value function ensures the actions are aligned with long-term goals. This enables the network to select the optimal target slot, ensuring its actions are semantically meaningful and spatially coherent with the user's behavior.

B. Reward

The reward function plays a crucial role in the training of reinforcement learning models by evaluating the quality of decisions made by the model in different states, guiding the model to gradually adjust its strategies during the learning process. Mapper models optimize decisions by maximizing the expected cumulative reward since time step t. In general, the process

can be described as follows: the agent receives an immediate reward R_t based on state S_t and decision A_t, then transitions to the next state S_{t+1} and optimizes its policy. Additionally, upon completion of optimization, an additional final reward is also provided to the agent. A well-designed reward function can help the agent accelerate the convergence of neural networks and make decisions to achieve the above objectives. The proposed reward function is based on scene semantics, behavior consistency, and temporal delay, and its specific details are as follows.

1) Scene Semantics: Scene semantics reward is used to measure the value of selecting a slot that conforms to the semantic mapping. Due to the different layouts among heterogeneous environments and the presence of one or more candidate slots that may meet the semantic requirements within a room, or even in extreme cases the absence of seemingly reasonable mapping slots, selecting the most user-desired virtual avatar mapping slot based on scene semantics poses a research challenge.

Based on previous user annotation data, a total of 4800 user feedback were obtained for 480 questions under 12 different room combinations, with annotations from 10 different users for each question. For the same question, as annotations from users may vary, using unprocessed data directly for training can lead to convergence issues in the model. Therefore, in order to meet the semantic expectations of the majority of users, a voting-based reward function is designed, where the higher the count of votes for a slot, the more suitable that slot is considered as the avatar target slot in the current scenario. Thus, the reward function term R_{sl} can be defined as follows:

$$R_{sl} = \begin{cases} 100 \times \frac{C_{A_{\text{final}}}}{C_{\max}} & (C_{A_{\text{final}}} \geq 1) \\ 100 \times \frac{C_{\min}}{C_{\max}} e^{-d_{\min}} & (C_{A_{\text{final}}} = 0) \end{cases} \tag{11}$$

Where C_{max} is the highest count of votes among all slots, A_{final} is the label of the slot finally selected by the target mapper, $C_{A_{final}}$ is the count of votes for the slot with the label A_{final}. For slots without any votes ($C_{A_{final}} = 0$), the reward gained by the agent for selecting that slot should not exceed the reward of any slot with votes, and the farther the distance from the optimal avatar target slot, the less the reward. Here, $C_{\min}$ is the minimum count of votes among slots with non-zero votes. Since there may be multiple slots with the highest count of votes, d_{min} denotes the minimum value from the euclidean distances between slots and the slot with the highest count of votes.

Directly using R_{sl} as the final reward and providing it to the agent upon reaching the target slot may lead to overly sparse rewards, resulting in learning difficulties. To enhance training efficiency, the final reward R_{sl} is transformed into an immediate reward $R_{sl,t}$, where the difference $R_{sl,t} - R_{sl,t-1}$ is passed as an immediate reward to the agent when the agent's behavior changes. The remaining reward of R_{sl} at the end is then provided as the final reward to the agent.

2) Behavior Consistency: The virtual avatar moves towards the target mapping slot along the computed shortest path. However, if the behavior of the target mapper changes too frequently, it will cause the virtual avatar to continuously switch to new targets, making it wander back and forth between multiple slots, leading to unstable movement of the virtual avatar, excessive path turning points, and long walking paths. Therefore, the reward function term R_{co} is defined to limit the generated path:

$$R_{co} = \begin{cases} 0 & (A_t = A_{t-1}) \\ -5 & (A_t \neq A_{t-1}) \end{cases} \tag{12}$$

This means that whenever the current behavior A_t of the target mapper differs from the behavior A_{t-1} at the previous time step, the agent will receive an immediate reward of -5 as the cost of behavior change. When the behavior remains consistent, the virtual avatar will move towards the target slot at a given speed along the shortest path until it reaches the target slot.

3) Temporal Delay: Due to the interruption caused by delays on the semantic synchronization of user interactions and the negative impact on user perception [16], the virtual agent should ideally arrive at their respective target slots simultaneously with the users. The reward function term R_{de} is defined to control the temporal delay of the virtual agent:

$$R_{de} = -100 \times \frac{\min(\tau, \tau_{\max})}{\tau_{\max}} \tag{13}$$

where τ is the delay between the virtual agent's arrival and the user's arrival at the target slot, and $\tau_{\max}$ is the maximum allowed delay (here, $\tau_{\max} = 3$). This reward function term controls the virtual avatar to arrive at the target slot as close to real-time as possible. For the virtual avatar's movement speed v_t^a, it can be dynamically adjusted as follows:

$$v_t^a = \min\left(\frac{d_t^a}{d_t^u} \max(v_{t-1}^a, 1), v_{\max}^a\right) \tag{14}$$

where v_{t-1}^a is the previous speed of the virtual avatar, $v_{\max}^a$ is the maximum allowed speed of the virtual avatar (here, $v_{\max}^a = 3$ m/s), and d_t^u and d_t^a represent the distances of the user and the virtual avatar to their respective target slots.

VII. Evaluation

A. Simulation Experiment

To evaluate the performance of our method, we conducted comprehensive simulations, including an ablation study and comparisons with other methods, as illustrated in Fig. 9. The experimental environment was developed using Unity3D 2021.3.5f1c1 and neural networks were implemented using ML-Agents toolkit (Release 20). Each network model was trained on a computer equipped with an Intel i7-9700 KF 3.60 GHz CPU and 32 GB RAM, and a GeForce RTX 2080Ti GPU for about 80 hours. The 4800 user-labeled data were randomly shuffled and divided into training and testing sets at a 3:1 ratio, ensuring that both sets included data from all 12 scene configuration pairs. The training set was used for model training, and all testing was performed on the test set.

1) Ablation Study: In our ablation study, the impact of different sub-terms in the reward function on the control of virtual agent behaviors was quantitatively tested. We compared our method (denoted as *Ours*) against the method without the behavior consistency reward term (denoted as *Ours w/o BC*), and

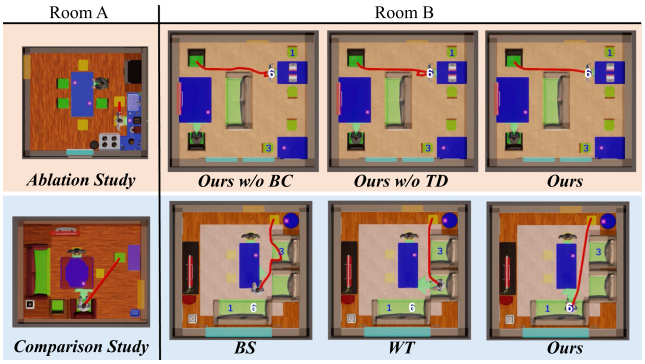


Fig. 9. Simulation experiment illustrating the performance of various methods. The top row presents the ablation study, comparing our method with variations excluding the behavior consistency term (*Ours w/o BC*) and the temporal delay term (*Ours w/o TD*). The bottom row shows the comparison study against the Baseline (*BS*) and Weighting Method (*WT*). The red path in Room B represents the movement trajectory of Avatar A under each method.

TABLE I
ABLATION AND COMPARISON STUDY RESULTS (MEAN ± STD)

Metric	*Ours w/o BC*	*Ours w/o TD*	*Ours*
DE (m)	0.54 ± 1.02	0.55 ± 0.95	0.57 ± 1.07
PLR	2.92 ± 3.49	1.64 ± 1.22	1.23 ± 1.09
BCT	44.79 ± 30.63	6.85 ± 6.83	1.83 ± 1.18
TD (s)	0.66 ± 0.85	0.36 ± 1.08	0.05 ± 0.18
Metric	***BS***	***WT***	***Ours***
DE (m)	1.61 ± 1.34	1.50 ± 1.33	0.57 ± 1.07
PLR	0.96 ± 0.28	1.17 ± 0.55	1.23 ± 1.09

the method without the temporal delay reward term (denoted as *Ours w/o TD*).

The evaluation metrics include Distance Error (*DE*, the distance in meters between the final avatar position and the optimal avatar target slot selected by most participants), Path Length Ratio (*PLR*, the ratio of the avatar's traveled path length to the user's), and Behavior Change Times (*BCT*, the number of behavior changes).

DE measures the semantic accuracy, *PLR* evaluates whether the avatar's movement is both semantically correct and spatially consistent by comparing the distances traveled by the user and avatar, and *BCT* assesses the consistency of the avatar's behavior throughout the movement. When evaluating the temporal delay reward term, *DE* and the Temporal Delay (abbreviated as *TD*) in seconds between the virtual agent reaching the goal and the user reaching the goal were used to assess whether this reward term could help synchronize the virtual agent's behavior with the user's.

Ablation study results were presented in the top of Table I. Regarding the behavior consistency reward term, Kolmogorov-Smirnov tests indicated that the data for *DE*, *PLR*, and *BCT* did not follow a normal distribution.

Post hoc analysis using Mann-Whitney U tests ($\alpha = 0.05$) showed that there was no significant difference in *DE* ($p = .948$) between *Ours* ($mean \pm std = 0.57 \pm 1.07$) and *Ours w/o BC* ($mean \pm std = 0.54 \pm 1.02$), but *PLR* ($-1.69, p < .001$) achieved with *Ours* ($mean \pm std = 1.23 \pm 1.09$) were significantly lower than *Ours w/o BC* ($mean \pm std = 2.92 \pm 3.49$). *BCT* ($-42.96, p < .001$) achieved with *Ours* ($mean \pm std = 1.83 \pm 1.18$) were significantly lower than *Ours w/o BC* ($mean \pm std = 44.79 \pm 30.63$).

For the temporal delay reward term, Kolmogorov-Smirnov test demonstrated that the *DE* and *TD* data did not conform to a normal distribution. Post hoc analysis using Mann-Whitney U tests ($\alpha = 0.05$) revealed that there was no significant difference in *DE* ($p = .782$) between *Ours* ($mean \pm std = 0.57 \pm 1.07$) and *Ours w/o TD* ($mean \pm std = 0.55 \pm 0.95$), but *TD* ($-0.31, p < .001$) obtained with *Ours* ($mean \pm std = 0.05 \pm 0.18$) was significantly lower than *Ours w/o TD* ($mean \pm std = 0.36 \pm 1.08$).

2) Comparison Study: In the comparison study, our method is quantitatively compared with two other contrasting methods. The descriptions of these three methods are as follows:

- *BS (Baseline):* The virtual avatar moves in real-time to the relative position of its corresponding user regarding the virtual avatar in the room and adjusts its orientation. If the movement collides with furniture, it will block the virtual avatar from continuing to move.
- *WT (Weighting Method):* This method evaluates candidate slots by considering category features, relative pose features, visual attention features, and spatial features. Each feature is assigned a weight based on its relative importance (manually refined). The slot with the highest weighted similarity to the user's target slot is selected as the optimal avatar target slot, and the avatar moves toward it along the shortest path.
- *Ours (Our Method):* The method based on reinforcement learning proposed in this paper, which controls the virtual avatar to gradually reach the mapping slot provided in real-time by the target mapper along the shortest path.

DE and *PLR* were reported, with the results shown in the bottom of Table I. Kolmogorov-Smirnov test indicated that the *DE* and *PLR* data did not follow a normal distribution. Thus we employed the non-parametric Kruskal-Wallis H test to analyze non-normally distributed data, which revealed a significant difference in *DE* among the methods ($H(2) = 53.365, p < .001$). Post-hoc analysis using Mann-Whitney U test ($\alpha = 0.05$) found that *DE* achieved by our method was significantly smaller than that of *BS* ($-1.04\,\text{m}, p < .001$) and *WT* ($-0.93\,\text{m}, p < .001$). However, there was no significant difference in *DE* between *BS* and *WT* ($p = .551$). Besides, the non-parametric Kruskal-Wallis H test revealed no significant difference in *PLR* among the methods ($H(2) = 3.434, p = .180$).

3) Discussion: We conducted an ablation study and a comparison study on the proposed method in a simulation platform to evaluate the rationality of the reward function design and the performance. In the ablation study, different sub-terms in the reward function were investigated for their effects on the control of the virtual avatar's behavior. The results demonstrated that under the influence of the behavior consistency reward term, the virtual avatar's movement distance decreased from nearly

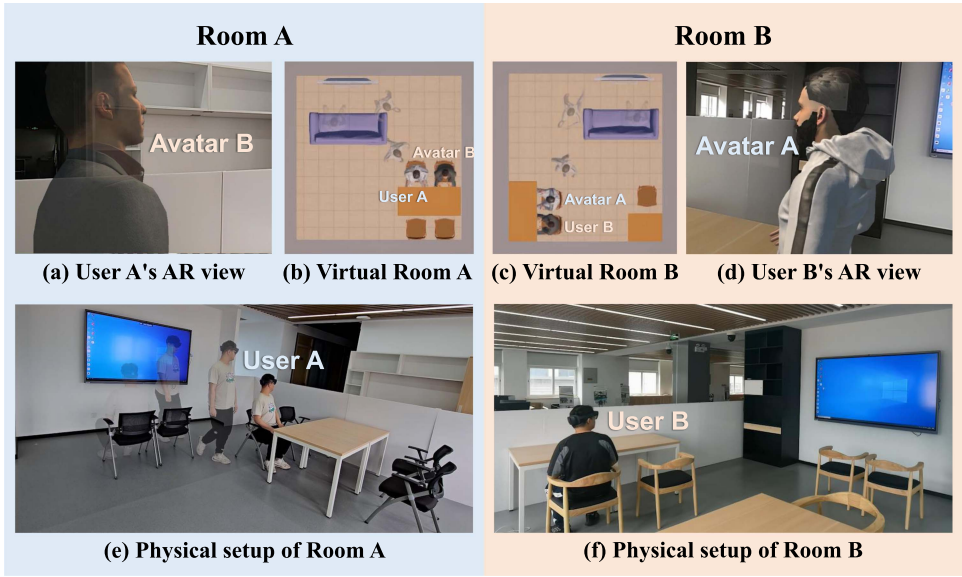


Fig. 10. Illustration of our AR multi-user experiment: (a) and (d) show the first-person AR views of User A and User B; (b) and (c) depict the top views of the virtual replicas (Room A and Room B); (e) and (f) present the physical room setups, demonstrating the spatial arrangement of furniture and users.

2.9 times the user's movement distance to approximately 1.2 times, while the number of behavior changes decreased from 44.8 times to 1.8 times, significantly reducing the redundant movement of the virtual avatar and enhancing behavior consistency. Furthermore, the temporal delay reward term facilitated a reduction in the temporal delay for the virtual avatar to reach the target from 0.36 s to 0.05 s, significantly improving the behavioral synchronization between the virtual avatar and the user. The ablation study results suggest that the incorporation of the behavior consistency and temporal delay reward terms optimizes the virtual avatar's movement path, significantly enhancing both behavior consistency and synchronization with the user.

In the comparison study, our method significantly reduced the distance error for the virtual avatar to reach the target compared to *BS* (-1.037 m) and *WT* (-0.929 m), indicating higher semantic accuracy and the ability to guide the virtual avatar to the desired locations. Although *WT* considered similar feature inputs as our reinforcement learning method, its distance error results showed no significant difference compared to *BS*, revealing that guiding the virtual avatar's movement through manually defined rules may not effectively meet the semantic requirements of indoor interaction and the demands for distributed multi-user collaboration. The powerful representational capacity of deep neural networks enables the learning of correct interactive semantic relationships from user-annotated data, thus controlling the virtual avatar's movement in a manner that satisfies the requirements of AR telepresence.

B. Multi-User AR Experiment

We conducted a multi-user AR experiment to assess the efficacy of our method and compare it with existing approaches in real scenarios. This experimental design necessitated the daily activities happen in telepresence collaboration.

1) Environments and Scenarios: We developed AR telepresence environments comprising two physical spaces measuring 4 m x 4 m. In our experimental setup, we outfitted these spaces with common indoor furniture items (TV, table, chair, etc.). We specifically adjusted the quantity and arrangement of objects to establish two distinct room layouts, which are illustrated in Fig. 10. The room layouts include different functional areas such as a lounge area and a study area, incorporating typical physical objects and interactions to facilitate various tasks.

In this study, we established three daily telepresence scenarios featuring real-time verbal communications and transitions between different locations. In Scenario 1, two users rest together, then one user leaves the lounge area to continue working. In contrast, Scenario 2 entails one user approaching the other from a distance. In scenario 3, two users navigate together to a shared destination, such as the lounge area. During each scenario, the user behaviors are mirrored in the corresponding avatar, creating a synchronized viewing experience for both users. We explain each scenario in detail below:

Scenario 1 (Leaving): The initial scene unfolds with two users watching TV together. Subsequently, User B says, "I have some homework, and I need to complete it first." Then User A responds, "Sure, go ahead and handle your homework," as User B departs from the lounge area and transitions into the study area to commence work.

Scenario 2 (Approaching): User B is doing homework as User A resides in the lounge area at a distance. Desiring to engage the remote user in viewing the homework content, User B vocally summons User A with, "Hey! Come check my homework." In response, User A acknowledges, "OK, I'm coming," and promptly starts moving towards Avatar B at the study area. Upon reaching and sitting, User A asks User B, stating, "All right, show me."

Scenario 3 (Moving together): Following the completion of homework, User B suggests watching TV, a proposal met with

agreement from User A. Then they simultaneously start transition from the study area to the lounge area and sit on a sofa for watching TV.

Illustration of the AR multi-user experiment. (a) and (d) show first-person AR views of User A and User B interacting with the environment. (b) and (e) depict top-down views of the virtual room replicas (Room A and Room B), highlighting furniture and user positions. (c) and (f) show the physical room setups, illustrating the actual spatial arrangement of furniture and users.

2) Methods: We compared our method with two alternatives, and the three methods are listed below:

- *BS (Baseline):* The user's avatar moves in real-time and adjusts its orientation according to the user's position relative to the other avatar in the scene. If the movement collides with furniture, the avatar is prevented from continuing the movement.
- *GT (Ground Truth):* Before the trial, participants need to understand task scenarios, and then annotate their ideal walking paths of themselves and their avatars with mouses according to the room top views shown on laptops. During the actual walking process, the annotated paths will be displayed in AR glasses to guide participants' walking, while their avatars walk along the annotated paths based on the current progress.
- *Ours (Our Method):* Our avatar control system containing a user target predictor and a reinforcement learning mapper, which controls the virtual avatar to gradually reach the mapping slot in real-time.

3) Metrics: According to previous studies [15], [16], [50], five metrics including similarity, semantics, preference, presence, and usability were adopted to evaluate method performances. The first metric is "similarity to the expectation", indicating the conformity between the avatar's transitions and participants' expectations. The second metric is "semantic correctness", which quantifies the alignment between the conversation context and the avatar's movements. The third metric is "overall preference", reflecting the subjective ranking of the method in this AR experience. The presence questionnaire is derived from the Networked Mind Measure of Social Presence [51] and the Temple Presence Inventory [52], while the questions for the method usability are chosen from the USE Questionnaire [53]. All questions of metrics are presented to participants using a seven-point Likert scale from 1 to 7, with 1 and 7 denoting the lowest and highest levels of agreement to these metrics.

Then we formulated five hypotheses:

H1: Our method achieves consistent similarity to the expectation with *GT* since both approaches consider user intent to guide avatar movements.

H2: The semantic correctness associated with our method is similar to *GT*, as the avatar movement will avoid obstacles and the avatar will finally arrive at the target location according to scene semantics or user requests.

H3: Participants' preference of our method is significantly higher than other methods, because it automates the control of virtual avatar movements to accurately mapped semantic positions, thereby reducing the need for manual user intervention.

H4: The presence with our method is equivalent to *GT*, as it provides more natural and realistic avatar movement, making users more aware of partner activities.

H5: The usability of our method is the highest as it is straightforward to operate, thereby reducing user workload while ensuring semantic accuracy.

4) Apparatus and Participants: The experiment was carried out in a laboratory having two physical tracking areas of $4\,\mathrm{m} \times 4\,\mathrm{m}$. Each participant wore a Microsoft Hololens2 HMD for visualizing virtual objects and avatars, and the system was implemented respectively on two laptops using Unity3D (v2021.3.5). Network communication is implemented through the Unity Netcode component, and the visuals are transmitted in real-time to AR glasses by Holographic Remoting application through wireless network. Avatar models and animation clips are from Mixamo[1] and Microsoft-Rocketbox.[2]

This multi-user study recruited 10 pairs of participants, comprising 10 males and 10 females aged between 21 to 30 years ($mean \pm std = 23.9 \pm 2.47$) from our university. The distribution of the pairs was as follows: 4 female-only, 4 male-only, and 2 mixed-gender pairs. All participants had normal or corrected-to-normal vision, and only 4 had no prior experience with AR/VR. The familiarity between teammates was assessed on a 7-point Likert scale ranging from 1 (never met before) to 7 (best friends or lovers), yielding $mean \pm std = 2.45 \pm 2.84$.

5) Task and Procedure: Our AR user study was approved by the Institutional Review Boards (IRB). Upon arrival at the laboratory in pairs, participants were instructed to complete a demographic questionnaire and we introduced the basic concept of our AR telepresence using a video tutorial and verbal instructions. Subsequently participants were led to separate rooms and they received training on operating the AR devices. During this preparation phase, participants were allowed to explore and got familiar with their room layouts as long as they want, and they could communicate with each other through AR devices.

With the completion of the preparation phase, the testing phase was initiated, which comprised a total of 9 trials (3 scenarios $\times$ 3 methods) for participant experience. The order of methods within each scenario was randomized and counterbalanced according to a Latin square design. After each trial, participants were required to fill out a seven-point Likert scale questionnaire, encompassing the previously mentioned questions. Additionally, participants also shared their feedback on each method during an interview. The experimental session lasted approximately 40 minutes.

6) Results: 90 valid trials (10 participant pairs $\times$ 3 scenarios $\times$ 3 methods) were collected. To analyze the study results, normality tests were conducted for all metrics via Kolmogorov-Smirnov tests, leading to the use of RM-ANOVA tests with Bonferroni-adjusted post-hoc testing if data is normally distributed. Greenhouse-Geisser corrections were applied when sphericity assumptions were violated. Non-parametric Friedman tests were employed for other non-normally distributed data. Wilcoxon signed-rank tests were conducted for post-hoc

[1]https://www.mixamo.com/

[2]https://github.com/microsoft/Microsoft-Rocketbox

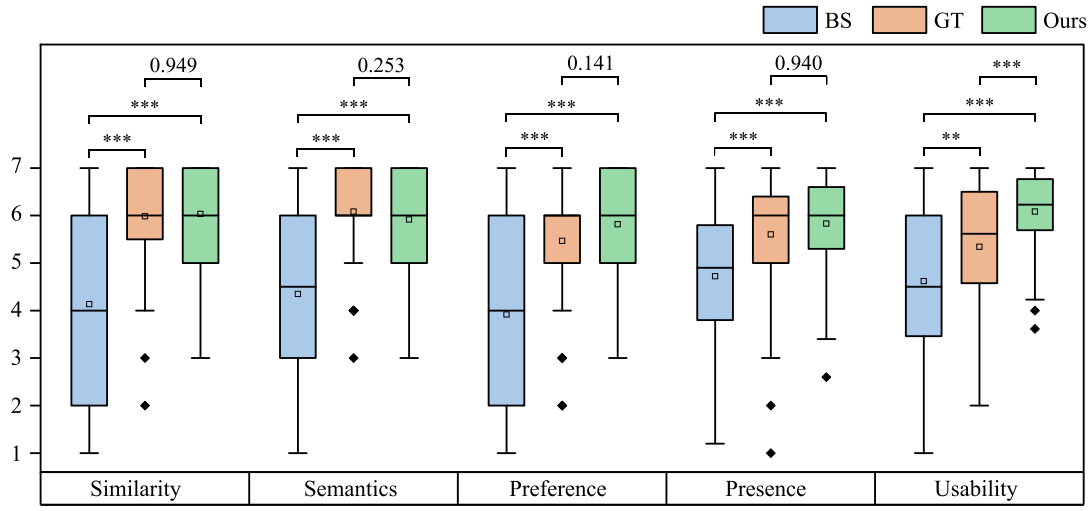


Fig. 11. Box plot results for the multi-user study, depicting means (hollow rectangles), median values (horizontal lines), and interquartile ranges (boxes), along with outliers (diamonds), for the metrics of similarity, semantics, preference, presence, and usability. The data is presented across three methods: *BS* (blue), *GT* (orange), and *Ours* (green). Statistically significant differences are marked with $*p < 0.05$, $**p < 0.01$, and $***p < 0.001$.

analysis upon discovering significant effects ($\alpha = 0.05$). Fig. 11 shows the median value (Mdn) and interquartile range (IQR) and the results of post-hoc tests on all metrics.

Similarity: Non-parametric Friedman test results revealed that METHOD had a significant effect on the similarity ($\chi^2(2) = 45.966, p < .001$). *BS* had significantly lower scores compared to *GT* ($-1.85, p < .001$) and *Ours* ($-1.90, p < .001$). Besides, there was no significant difference found between *GT* and *Ours* ($-0.05, p = .949$).

Semantics: Non-parametric Friedman test results revealed that METHOD had a significant effect on the Semantics ($\chi^2(2) = 44.103, p < .001$). *BS* performed significantly worse than *GT* ($-1.73, p < .001$) and *Ours* ($-1.57, p < .001$). Furthermore, there was no significant difference found between *GT* and *Ours* ($+0.16, p = .253$).

Preference: Non-parametric Friedman test results revealed that METHOD had a significant effect on the Preference ($\chi^2(2) = 47.813, p < .001$). *BS* was significantly less preferred compared to *GT* ($-1.55, p < .001$) and *Ours* ($-1.90, p < .001$). Meanwhile, *GT* scored marginally lower than *Ours* ($-0.35, p = .141$), though the difference was not statistically significant.

Presence: The presence indicator comprises 5 sub-questions, we evaluated it by calculating the mean score of these sub-questions for each participant (shown in Table II). Non-parametric Friedman test results revealed that METHOD had a significant effect on the presence ($\chi^2(2) = 24.524, p < .001$). *BS* had significantly lower scores compared to *GT* ($-0.88, p < .001$) and *Ours* ($-1.11, p < .001$). Besides, there was no significant difference found between *GT* and *Ours* ($-0.23, p = .094$).

Usability: For the usability indicator which comprises 13 sub-questions, we calculated the mean of each participant's responses to these sub-questions (shown in Table III). Non-parametric Friedman test results revealed that METHOD had a significant effect on the usability ($\chi^2(2) = 32.376, p < .001$). *BS* had significantly lower scores compared to *GT* ($-0.72, p =$

TABLE II
PRESENCE QUESTIONNAIRE RESULTS FOR EACH CONDITION (MEAN ± STD)

Question	BS	GT	Ours
1. I remained focused on my partner throughout our interaction.	5.53 (1.52)	5.82 (1.46)	6.18 (1.08)
2. My partner remained focused on me throughout our interaction.	4.87 (1.82)	5.32 (1.78)	5.50 (1.61)
3. My partner's thoughts were clear to me.	4.27 (1.89)	5.58 (1.47)	5.67 (1.32)
4. It was easy to understand my partner.	4.28 (1.96)	5.73 (1.45)	5.85 (1.35)
5. It seems that I and my partner was together in the same place.	4.67 (1.82)	5.57 (1.41)	5.95 (1.32)

TABLE III
USABILITY QUESTIONNAIRE RESULTS FOR EACH CONDITION (MEAN ± STD)

Question	BS	GT	Ours
1. It helps me be more effective.	4.22 (1.78)	5.07 (1.53)	5.83 (0.99)
2. It is useful.	4.10 (1.87)	5.32 (1.44)	5.93 (0.90)
3. It meets my needs.	3.95 (1.99)	5.42 (1.44)	5.97 (0.92)
4. It is easy to use.	5.23 (1.67)	5.28 (1.46)	6.23 (1.00)
5. It is simple to use.	5.13 (1.74)	5.33 (1.47)	6.30 (0.89)
6. It is user friendly.	4.78 (1.91)	5.28 (1.55)	6.20 (0.92)
7. I learned to use it quickly.	5.37 (1.68)	5.63 (1.37)	6.37 (0.78)
8. I easily remember how to use it.	5.50 (1.52)	5.60 (1.43)	6.40 (0.76)
9. It is easy to learn to use it.	5.50 (1.57)	5.63 (1.43)	6.35 (0.80)
10. I am satisfied with it.	4.02 (1.94)	5.25 (1.43)	5.90 (1.02)
11. I would recommend it to a friend.	3.90 (1.91)	5.03 (1.62)	5.82 (1.14)
12. It is fun to use.	4.43 (1.93)	5.40 (1.45)	5.97 (1.12)
13. It works the way I want it to work.	3.93 (2.06)	5.20 (1.46)	5.83 (1.03)

.004) and *Ours* ($-1.46, p < .001$). Besides, the scores of *GT* were significantly lower than those of *Ours* ($-0.74, p < .001$).

7) Interview Findings: In the gathered user comments, our method was shown to be the preferred choice of almost all participants (17 out of 20) for enhancing collaboration and communication in AR scenes. P8 commented, "With *Ours*, the avatar moved way more naturally to the right spot. It was easy to figure out what my partner was trying to do just by watching the avatar move around." P19 said, "*Ours* gave me a better sense of immersion. The avatar moved without any misalignment, and the interaction felt smooth and realistic, with the paths and destinations matching what my partner and I expected. It was like we were actually in the same space together." These comments indicated that our approach was more semantically accurate and reasonable, which in turn improved user immersion.

Furthermore, some participants perceived *Ours* as more efficient, having a lower learning cost, and allowing them to get started quickly. P17 noted: "*Ours* did not require me to learn the method and label the data in advance, which made it easier and more convenient to use." Similarly, P12 commented, "*Ours* had similar accuracy to *GT* but was more convenient since it did not require the pre-labelling process." These comments suggest that users valued *Ours* for striking a balance between quality and efficiency.

BS was generally perceived as ineffective by participants, with the avatar's position deviating significantly from expectations. As P13 noted, "With *BS*, the avatar failed to walk to the correct position and may have even sat on a non-existent chair, severely disrupting the sense of immersion." This observation suggests that *BS* could not effectively support collaboration and communication in AR environments.

Some participants felt that the avatar's movement had a great impact on method judgement. P20 explained, "In some scenarios, my partner's avatar remained still when I was walking, so I could not perceive a difference between the three methods." Specifically, *Ours* enhanced immersion through real-time and precise avatar movements, while *GT* facilitated understanding of the intended actions through auxiliary lines.

A few participants expressed a preference for *GT*. P5 explained, "*GT* allowed me to see auxiliary lines and help me understand each other's movements and intentions." P16 commented, "As a frequent user of electronic devices, I did not find the task of pre-drawing lines to be inconvenient, and I would only consider abandoning *GT* if a non-line drawing method offered a more immersive experience." Based on participant feedback, it was evident that *Ours* could be improved in terms of providing auxiliary prompts to enhance understanding.

8) Discussion: From the quantitative evaluation, *Ours* and *GT* significantly outperformed *BS* across all metrics. However, when comparing *Ours* against *GT*, no statistically significant differences were observed in similarity, semantics, presence, and preference scores. Regarding usability, *Ours* achieved a higher score than *GT*, exhibiting a significant improvement. In other words, our proposed method effectively addresses the challenge of controlling avatar movements while preserving semantic understanding in collaborative scenarios. For the remaining four metrics besides usability, our method's performance was on par with *GT*, showing no significant deviations. Furthermore, our method demonstrated a significant usability advantage over *GT*.

The experimental statistical results fully supported our four hypotheses, namely H1, H2, H4, and H5. Regarding H3, while *Ours* exhibited a higher average score compared to *GT*, the difference between these two methods did not reach statistical significance. Additionally, according to the collected participant interview data, the majority of participants (17 out of 20) expressed a preference for *Ours*. Consequently, there is a modest basis for entertaining this hypothesis.

Synthesizing the participant feedback from the interviews, *Ours* emerged as the preferred primary method. This preference stemmed from our method's ability to execute reasonable walking processes, resulting in more natural partner avatar movements and more easily interpretable intentions. However, some participants noted that due to their frequent use of electronic devices, the additional steps required by *GT* were manageable, and they did not feel compelled to choose *Ours* over *GT*. This observation may be related to the participants' individual habits.

VIII. Limitations

There are several limitations in our work. First, our method is currently designed on 2D planar maps and the performance is dependent on the quality of user annotations. Our method implicitly addresses contextual mismatches, such as differences in object presence or functionality between physical spaces (e.g., a TV in the living room versus none in the kitchen). While this reliance provides flexibility across diverse environments, it does not explicitly account for each potential mismatch. Future work could address this by developing systematic criteria to identify and describe mismatches between given scenes. Besides, enhancing the dataset with more diverse user annotations could enable the model to generalize better to a wider range of environments and behaviors. Exploring automatic or semi-automatic annotation techniques would further reduce dependency on manual labeling, enhancing both scalability and robustness.

Another limitation is the challenge posed by large and complex indoor environments, such as duplex rooms, where our method may struggle to accurately represent spatial relationships. Moreover, the approach's scalability to outdoor environments and its performance on entirely unseen spatial configurations warrant further investigation. Future work could explore multi-scale scene representations, dynamic scene analysis, and strategies for improving generalization to unseen configurations, especially in more open or complex environments.

Additionally, our method assumes that user movements are constrained by predefined slots, which simplifies the problem and improves computational efficiency. However, this discretization limits flexibility and may not be suitable for scenarios where users can remain idle at arbitrary positions within the environment. Therefore, future work could explore techniques for continuous position prediction and dynamic area mapping, which would enable the system to better accommodate diverse movement patterns.

Moreover, the scope of our work is restricted to avatar motion adaption from a locomotion perspective in heterogeneous environments. While our method enables realistic transition

mapping of user movements, it does not generate detailed avatar motions such as gazes, facial expressions, and interaction gestures. However, these are crucial for achieving a high degree of immersion and natural communication in AR telepresence meetings. To fully leverage the potential of our cross-scene avatar locomotion adaption method, future research could explore real-time animation generation that can seamlessly integrate these finer-grained motion details based on user interaction and environmental context, further enhancing the realism and interactivity of avatar motion adaption in heterogeneous environments.

Furthermore, our current work focuses on a typical AR telepresence scenario involving only two concurrent users situated in two heterogeneous physical environments. While this setup enables the investigation of core challenges in cross-reality embodiment and interaction, it does not explore the scalability and robustness of our approach when extended to more complex multi-user scenarios. As AR telepresence applications continue to evolve, it is crucial to explore how our method can be improved and adapted to scenarios involving multiple users simultaneously distributed across more than two different physical environments.

Finally, while our work does not directly address privacy and ethical concerns, these issues are critical for the development of AR telepresence systems. To enhance privacy protection, sensitive user data, such as local scene information and movement trajectories, could be processed locally on the user's device rather than transmitted to external servers, reducing the risk of data exposure. Additionally, it is important to consider the potential for avatar behaviors to unintentionally convey bias or cause miscommunication. Future work should explore solutions to mitigate these risks, ensuring that AR telepresence systems are deployed in a way that respects privacy and ethical standards.

IX. Conclusion and Future Work

In this paper, we propose a novel technique for controlling avatar movements for multi-user AR telepresence in heterogeneous environments. Our method framework includes a user target predictor and a reinforcement learning mapper, which learns from collected user-annotated data and scene semantic information, predicts user target positions in real-time, and maps them to corresponding locations in the room where the virtual avatar is located. According to both simulation and real user experiments, our method can generate natural avatar movements that adhere to physical environmental constraints, enabling more natural and intuitive interactions in distributed spatial collaborations.

Our approach offers a wide range of potential applications in AR and the metaverse. In team collaboration and remote meetings, it enables avatars to accurately reflect real-time spatial movements, enhancing collaboration across locations and providing an immersive experience. In virtual education, our method allows teacher avatars to adapt dynamically to students' environments, increasing interactivity and engagement. Additionally, in virtual healthcare, avatars can support remote patient monitoring and telemedicine, allowing doctors to observe physical behaviors for more personalized and empathetic consultations.

In future work, we aim to enhance the generalization capability of our method to support larger and more complex indoor environments, as well as outdoor scenarios. Furthermore, integrating our method with more detailed real-time avatar motion adaption methods, such as gaze control and gesture generation, is also worth investigating. As AR telepresence applications evolve, extending our approach to scenarios involving multiple users simultaneously distributed across more than two different physical environments would be valuable.

References

[1] J. Carmigniani, B. Furht, M. Anisetti, P. Ceravolo, E. Damiani, and M. Ivkovic, "Augmented reality technologies, systems and applications," *Multimedia Tools Appl.*, vol. 51, pp. 341–377, 2011.

[2] E. P. Yildiz, "Augmented reality research and applications in education," in *Proc. Augmented Reality Appl.*, 2021, pp. 1–2.

[3] C. Li, W. Li, H. Huang, and L.-F. Yu, "Interactive augmented reality storytelling guided by scene semantics," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 1–15, 2022.

[4] T. Rhee, S. Thompson, D. Medeiros, R. Dos Anjos, and A. Chalmers, "Augmented virtual teleportation for high-fidelity telecollaboration," *IEEE Trans. Vis. Comput. Graph.*, vol. 26, no. 5, pp. 1923–1933, May 2020.

[5] A. Maimone, X. Yang, N. Dierk, A. State, M. Dou, and H. Fuchs, "General-purpose telepresence with head-worn optical see-through displays and projector-based lighting," in *Proc. IEEE Virtual Reality*, 2013, pp. 23–26.

[6] S. Orts-Escolano et al., "Holoportation: Virtual 3D teleportation in real-time," in *Proc. 29th Annu. Symp. User Interface Softw. Technol.*, 2016, pp. 741–754.

[7] N. Yamashita, K. Hirata, T. Takada, Y. Harada, Y. Shirai, and S. Aoyagi, "Effects of room-sized sharing on remote collaboration on physical tasks," *IPSJ Digit. Courier*, vol. 3, pp. 788–799, 2007.

[8] I. Podkosova and H. Kaufmann, "Co-presence and proxemics in shared walkable virtual environments with mixed colocation," in *Proc. 24th ACM Symp. Virtual Reality Softw. Technol.*, 2018, pp. 1–11.

[9] J. S. Casaneuva, "Presence and co-presence in collaborative virtual environments," Master's thesis, University of Cape Town, 2001.

[10] M. Keshavarzi, A. Y. Yang, W. Ko, and L. Caldas, "Optimization and manipulation of contextual mutual spaces for multi-user virtual and augmented reality interaction," in *Proc. 2020 IEEE Conf. Virtual Reality 3D User Interfaces (VR)*, 2020, pp. 353–362.

[11] N. H. Lehment, D. Merget, and G. Rigoll, "Creating automatically aligned consensus realities for ar videoconferencing," in *Proc. 2014 IEEE Int. Symp. Mixed Augmented Reality*, 2014, pp. 201–206.

[12] J. E. S. Grønbæk et al., "Partially blended realities: Aligning dissimilar spaces for distributed mixed reality meetings," in *Proc. 2023 CHI Conf. Hum. Factors Comput. Syst.*, 2023, pp. 1–16.

[13] M. Keshavarzi, M. Zollhoefer, A. Y. Yang, P. Peluse, and L. Caldas, "Mutual scene synthesis for mixed reality telepresence," 2022, *arXiv:2204.00161*.

[14] D. Kim and W. Woo, "Edge-centric space rescaling with redirected walking for dissimilar physical-virtual space registration," 2023, *arXiv:2308.11210*.

[15] L. Yoon, D. Yang, J. Kim, C. Chung, and S.-H. Lee, "Placement retargeting of virtual avatars to dissimilar indoor environments," *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 3, pp. 1619–1633, Mar. 2022.

[16] X. Wang, H. Ye, C. Sandor, W. Zhang, and H. Fu, "Predict-and-drive: Avatar motion adaption in room-scale augmented reality telepresence with heterogeneous spaces," *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 11, pp. 3705–3714, Nov. 2022.

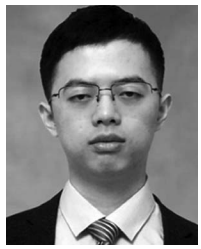
[17] J. Thomas and E. S. Rosenberg, "A general reactive algorithm for redirected walking using artificial potential functions," in *Proc. 2019 IEEE Conf. Virtual Reality 3D User Interfaces (VR)*, 2019, pp. 56–62.

[18] M. Hassan et al., "Stochastic scene-aware motion prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 11374–11384.

[19] M. Hassan, P. Ghosh, J. Tesch, D. Tzionas, and M. J. Black, "Populating 3 D scenes by learning human-scene interaction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 14708–14718.

[20] S. Huang et al., "Diffusion-based generation, optimization, and planning in 3 d scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 16750–16761.

[21] M. Wang, Y.-J. Li, J. Shi, and F. Steinicke, "Scenefusion: Room-scale environmental fusion for efficient traveling between separate virtual environments," *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 8, pp. 4615–4630, Aug. 2024.
[22] T. Pejsa, J. Kantor, H. Benko, E. Ofek, and A. Wilson, "Room2Room: Enabling life-size telepresence in a projected augmented reality environment," in *Proc. 19th ACM Conf. Comput.-Supported Cooperative Work Social Comput.*, 2016, pp. 1716–1725.
[23] Microsoft, "Microsoft mesh," Microsoft Teams. Accessed: Jan. 12, 2024. [Online]. Available: https://www.microsoft.com/en-us/microsoft-teams/microsoft-mesh
[24] I. M. Platforms, "Meta horizon workrooms," Meta. Accessed: Dec. 01, 2024. [Online]. Available: https://forwork.meta.com/horizon-workrooms
[25] S. Choi, S. Hong, K. Cho, C. Kim, and J. Noh, "Online avatar motion adaptation to morphologically-similar spaces," in *Computer Graphics Forum*, vol. 42, Hoboken, NJ, USA: Wiley, 2023, pp. 13–24.
[26] D. Kim, J.-E. Shin, J. Lee, and W. Woo, "Adjusting relative translation gains according to space size in redirected walking for mixed reality mutual space generation," in *Proc. 2021 IEEE Virtual Reality 3D User Interfaces (VR)*, 2021, pp. 653–660.
[27] L. Sidenmark, T. Zhang, L. Al Lababidi, J. Li, and T. Grossman, "Desk2Desk: Optimization-based mixed reality workspace integration for remote side-by-side collaboration," in *Proc. 37th Annu. ACM Symp. User Interface Softw. Technol.*, 2024, pp. 1–15.
[28] D. Jo, K.-H. Kim, and G. J. Kim, "Spacetime: Adaptive control of the teleported avatar for improved ar tele-conference experience," *Comput. Animation Virtual Worlds*, vol. 26, no. 3/4, pp. 259–269, 2015.
[29] D. Yang, J. Kang, T. Kim, and S.-H. Lee, "Visual guidance for user placement in avatar-mediated telepresence between dissimilar spaces," *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 12, pp. 7558–7570, Dec. 2024.
[30] J. Kang, D. Yang, T. Kim, Y. Lee, and S.-H. Lee, "Real-time retargeting of deictic motion to virtual avatars for augmented reality telepresence," in *Proc. 2023 IEEE Int. Symp. Mixed Augmented Reality*, 2023, pp. 885–893.
[31] Y. Lang, W. Liang, and L.-F. Yu, "Virtual agent positioning driven by scene semantics in mixed reality," in *Proc. 2019 IEEE Conf. Virtual Reality 3D User Interfaces (VR)*, 2019, pp. 767–775.
[32] A. Watkins, A. Ullal, and N. Sarkar, "Every 'body' gets a say: An augmented optimization metric to preserve body pose during avatar adaptation in mixed/augmented reality," *IEEE Trans. Vis. Comput. Graph.*, early access, Apr. 17, 2024, doi: 10.1109/TVCG.2024.3388376.
[33] T. Tahara, T. Seno, G. Narita, and T. Ishikawa, "Retargetable AR: Context-aware augmented reality in indoor scenes based on 3 D scene graph," in *Proc. 2020 IEEE Int. Symp. Mixed Augmented Reality Adjunct*, 2020, pp. 249–255.
[34] Y. Huang and M. Kallmann, "Planning motions and placements for virtual demonstrators," *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 5, pp. 1568–1579, May 2015.
[35] C. Li and L.-F. Yu, "Generating activity snippets by learning human-scene interactions," *ACM Trans. Graph.*, vol. 42, no. 4, pp. 1–15, 2023.
[36] W. Li, C. Li, M. Kim, H. Huang, and L.-F. Yu, "Location-aware adaptation of augmented reality narratives," in *Proc. 2023 CHI Conf. Hum. Factors Comput. Syst.*, 2023, pp. 1–15.
[37] M. Kim, R. Alghofaili, C. Li, and L.-F. Yu, "Dragon's path: Synthesizing user-centered flying creature animation paths for outdoor augmented reality experiences," in *Proc. ACM Siggraph 2024 Conf. Papers*, 2024, pp. 1–11.
[38] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *Proc. Auton. Agents Multiagent Syst.: AAMAS 2017 Workshops*, Best Papers, São Paulo, Brazil, Springer, 2017, pp. 66–83.
[39] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Proc. 10th Int. Conf. Mach. Learn.*, 1993, pp. 330–337.
[40] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
[41] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa, "Amp: Adversarial motion priors for stylized physics-based character control," *ACM Trans. Graph.*, vol. 40, no. 4, pp. 1–20, 2021.
[42] M. Hassan, Y. Guo, T. Wang, M. Black, S. Fidler, and X. B. Peng, "Synthesizing physical character-scene interactions," in *Proc. ACM SIGGRAPH 2023 Conf.*, New York, NY, USA, 2023, pp. 1–9. [Online]. Available: https://doi.org/10.1145/3588432.3591525
[43] X. B. Peng, P. Abbeel, S. Levine, and M. Van de Panne, "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–14, 2018.
[44] K. Zhu and T. Zhang, "Deep reinforcement learning based mobile robot navigation: A review," *Tsinghua Sci. Technol.*, vol. 26, no. 5, pp. 674–691, 2021.
[45] W. Yang, X. Wang, A. Farhadi, A. Gupta, and R. Mottaghi, "Visual semantic navigation using scene priors," 2018, *arXiv: 1810.06543*.
[46] X. Liu, D. Guo, H. Liu, and F. Sun, "Multi-agent embodied visual semantic navigation with scene prior knowledge," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 3154–3161, Apr. 2022.
[47] E. Kolve et al., "AI2-THOR: An interactive 3D environment for visual AI," 2017, *arXiv: 1712.05474*.
[48] N. Pelekis, I. Kopanakis, G. Marketos, I. Ntoutsi, G. Andrienko, and Y. Theodoridis, "Similarity search in trajectory databases," in *Proc. IEEE 14th Int. Symp. Temporal Representation Reasoning*, 2007, pp. 129–140.
[49] E. T. Hall and E. T. Hall, *The Hidden Dimension*, vol. 609. Garden City, NY, USA: Anchor, 1966.
[50] Y. Choi, J. Lee, and S.-H. Lee, "Effects of locomotion style and body visibility of a telepresence avatar," in *2020 IEEE Conf. Virtual Reality 3D User Interfaces*, 2020, pp. 1–9.
[51] C. Harms and F. Biocca, "Internal consistency and reliability of the networked minds measure of social presence," in *Proc. 7th Annu. Int. Workshop: Presence*, Universidad Politecnica de Valencia Valencia, Spain, 2004, pp. 1–7.
[52] M. Lombard, T. B. Ditton, and L. Weinstein, "Measuring presence: The temple presence inventory," in *Proc. 12th Annu. Int. Workshop Presence*, 2009, pp. 1–15.
[53] A. M. Lund, "Measuring usability with the use questionnaire12," *Usability Interface*, vol. 8, no. 2, pp. 3–6, 2001.



Yi-Jun Li (Member, IEEE) received the PhD degree from Beihang University, in 2024. He is currently a postdoc researcher in the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His research interests include virtual reality, redirected walking, and virtual collaboration.

Hao-Zhong Yang received the bachelor degree from China University of Geosciences Beijing, in 2023. He is currently working toward the master's degree with Beihang University, Beijing, China. His major research interest is in virtual reality and computer graphics. Now he is working on research topics related to automated design and perception in virtual reality.

Wen-Tong Shu .received the bachelor's degree from the Renmin University of China, in 2023. He is currently working toward the master's degree with Beihang University, Beijing, China. His major research interest is in virtual reality and computer graphics. Now he is working on research topics related to virtual reality teleportation and perception.

Miao Wang (Member, IEEE) received the PhD degree from Tsinghua University, in 2016. He is an associate professor with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, China. He is also an adjunct researcher with Zhongguancun Laboratory, Beijing, China. His research interests include Virtual Reality, Computer Graphics and Visual Computing. During 2016-2018, he did postdoc research in Visual Computing with Tsinghua University. He serves as a program committee member of IEEE VR and ISMAR conferences. He is a member of ACM and Asiagraphics.