Real-time Translation of Upper-body Gestures to Virtual Avatars in Dissimilar Telepresence Environments

Jiho Kang, Taehei Kim, Hyeshim Kim, and Sung-Hee Lee

***Abstract*—In mixed reality (MR) avatar-mediated telepresence, avatar movement must be adjusted to convey the user's intent in a dissimilar space. This paper presents a novel neural network-based framework designed for translating upper-body gestures, which adjusts virtual avatar movements in dissimilar environments to accurately reflect the user's intended gestures in real-time. Our framework translates a wide range of upper-body gestures, including eye gaze, deictic gestures, free-form gestures, and the transitions between them. A key feature of our framework is its ability to generate natural upper-body gestures for users of different sizes, irrespective of handedness and eye dominance, even though the training is based on data from a single person. Unlike previous methods that require paired motion between users and avatars for training, our framework uses an unpaired approach, significantly reducing training time and allowing for generating a wider variety of motion types. These advantages were made possible by designing two separate networks: the Motion Progression Network, which interprets sparse tracking signals from the user to determine motion progression, and the Upper-body Gesture Network, which autoregressively generates the avatar's pose based on these progressions. We demonstrate the effectiveness of our framework through quantitative comparisons with state-of-the-art methods, qualitative animation results, and a user evaluation in MR telepresence scenarios.**

***Index Terms*—Telepresence, virtual avatar, motion retargeting, human animation, mixed reality.**

I. Introduction

Mixed reality (MR) telepresence enables remote participants to feel like they are in the same space by transferring virtual avatars. These avatars' natural and lifelike movements can make interactions immersive and realistic, even when the users are in different locations. A key factor is how accurately virtual avatars can convey the user's intent, despite the spatial dissimilarities between the telepresence environments. Researchers have explored methods such as creating mutual spaces [1], optimally positioning virtual avatars [2]–[4], or finding a partial alignment between spaces [5], [6]. However, if the furniture arrangements differ significantly between the two spaces, identifying equivalent spatial relationships may not be possible. Additionally, cluttered environments can drastically reduce the space available to remote participants, leading to poor spatial utilization. In such cases, simply copying and pasting user motions onto avatars can result in ineffective interactions.

Jiho Kang, Taehei Kim, Hyeshim Kim, and Sung-Hee Lee (corresponding author) are with KAIST. E-mail: {jhkang0408, sunghee.lee}@kaist.ac.kr

Fig. 1. User egocentric views (top) and room perspectives (bottom) of our MR telepresence spaces A (left) and B (right). In space B, avatar X′ represents user X from space A. In space A, avatar Y′ represents user Y from space B. Virtual avatars and objects are augmented at different locations and scales in each space. Both users focus their eye gaze on the avatar of the other party. They point at a particular object (Jupiter) with one hand (user X using the left hand and user Y using the right hand) while using the other hand to perform explanatory gestures. Our system accurately translates these upper-body gestures into the respective avatar motions in the remote space in real-time, enabling effective bi-directional interaction between users in remote locations.

Figure 1 illustrates the key challenges in telepresence scenarios where spatial arrangements of users, avatars, and points of interest differ between spaces. When users primarily communicate through upper-body gestures, these gestures must be accurately translated to the avatar to suit the remote space. This process should encompass a wide range of dynamically changing gestures, including eye gaze, deictic gestures (such as pointing at and touching objects), free-form gestures, and transitions between them. Since this translation must occur in real-time, the system should instantly reflect the user's continuous movements on the avatar's pose using only current and past motion data, without access to future information. Additionally, the solution must be universally applicable across avatars with diverse body dimensions.

To address these challenges, we propose a novel two-stage approach that decouples upper-body gesture translation into two modular sub-tasks: understanding the user's gesture and generating the avatar's corresponding motion in a dissimilar environment. This strategy is implemented through two independent deep neural networks, the Motion Progression Network (MPNet) and the Upper-body Gesture Network (UGNet).

The MPNet takes user-invariant behavioral features between

the user's joints and interaction targets to determine the degree of progression for each gesture category (simply referred to as *motion progression* hereafter), operating on a time window of current and past tracking signals. These features are designed to be independent of individual user characteristics. Given the output of the MPNet, the UGNet determines the avatar's gaze direction, head and right-hand transformations, and right index fingertip position. These avatar-invariant outputs applicable across different avatar structures, are used as end-effectors for inverse kinematics (IK) to compute the final avatar upper-body motions. During training, MPNet and UGNet are trained separately using the same captured motion sequences for their specific tasks, while at runtime, these networks connect to form an end-to-end pipeline.

This two-network structure, trained independently, overcomes the limitations of the state-of-the-art (SOTA) method [7]. Unlike [7], which learns to directly map user motions to avatar motions using paired motion datasets, our unpaired approach indirectly connects the user and avatar motions through the motion progression as an intermediate stage. This is not tied to specific user-avatar motion pairs, resulting in improved training efficiency and broader motion variety. Despite being trained on a single individual's motion data, our approach ensures robust performance across user attributes such as size, handedness, and eye dominance. Furthermore, our work enhances avatar realism through eye animation and handles more challenging scenarios where the head and hand move independently (Figure 1).

To the best of our knowledge, this work is the first data-driven approach to interpret users' general upper-body gestures and translate them to corresponding virtual avatars in dissimilar telepresence environments. We conducted a comprehensive quantitative evaluation to validate our method. This includes the comparison with the SOTA method [7] and alternative network architectures. We also demonstrate our approach's qualitative effectiveness with animation results. Finally, we implemented an MR telepresence application prototype and performed a user evaluation to examine how our framework supports actual collaboration and social interaction.

In this study, we limited spatial dissimilarity between spaces with objects of the same type, but in different arrangements and shapes. The target objects include avatars, TVs, and virtual objects. In real-world applications, recognizing the correspondence between objects in two different spaces, and identifying the target object of upper-body gestures is essential. Recent research has advanced in this direction, with several works demonstrating methods for predicting users' pointing targets [8], [9] and gaze targets [10] in MR environments. While target prediction continues to evolve, we focused on high-quality gesture translation by manually predefining this information for the scope of this study.

In summary, the principal contributions of this paper are as follows:

- The first learning-based real-time translation of general upper-body gesture, including eye gaze, deictic gesture (pointing at and touching objects), free-form gesture, and the transitions between these gestures to virtual avatars in dissimilar telepresence environments.
- A novel unpaired dataset approach that eliminates the need for motion pairing, significantly reducing training time and enabling a wide range of action categories.
- A lightweight, real-time framework for MR telepresence applications that generalizes to various users, despite being trained on single-person data.
- For follow-up research, we release the source code and dataset at https://github.com/jhkang0408/RTUGVADTE.

II. Related Work

A. MR Telepresence

Realistically reconstructing objects from the local space into the remote space in real-time is a critical element that directly influences user immersion in the MR telepresence experience. To achieve this, researchers have utilized methods such as projection, display, and capture. Maimone et al. [11] pioneered a proof-of-concept telepresence system with real-time 3D scene capture and continuous-viewpoint head-tracked stereo 3D display. Expanding on this, Maimone et al. [12] designed a general-purpose telepresence framework with optical see-through displays and projector-based lighting for a natural experience. Beck et al. [13] presented an immersive telepresence system that allows distributed user groups to meet in a shared virtual 3D world, continuously capturing users and their local interaction spaces at each site. Pan et al. [14] emphasized the importance of gaze, attention, and eye contact in communication by proposing a low-cost cylindrical videoconferencing system for multiple viewpoints. Pejsa et al. [15] invented the Room2Room, which enables lifelike, collaborative interaction between two remote participants, capturing virtual copies of people and objects from one space and projecting them into another. Orts et al. [16] introduced Holoportation, an end-to-end MR telepresence system, demonstrating high-fidelity, real-time 3D reconstruction of distant individuals and objects. In a similar approach, Loki leverages video, audio, and spatial capture to connect local and remote space [17], creating a bi-directional MR system.

Another important challenge in MR telepresence is to overcome the spatial dissimilarity between the spaces. Several researchers have tackled this issue by creating mutual spaces. For example, Lehment et al. [18] developed a method for aligning two MR Teleconference rooms into a shared workspace. Extending this concept, Keshavarzi et al. [1] introduced a framework that accommodates different room layouts and sizes to create an optimal mutual space for multiple user interactions. Grønbæk et al. [5] presented the concept of Partially Blended Realities to support remote collaborators in partially aligning their physical spaces. Kim et al. [19] proposed a space-rescaling technique using redirected walking to register dissimilar physical-virtual spaces. Fink et al. [6] introduced a concept of Relocations that relocate remote user representations to local space with spatial awareness and referencing.

Optimizing avatar placement and behavior is another promising approach for limited space in shared areas. Jo et al. [20] suggested a method focused on positioning the avatar. They match spatial and object-level characteristics between

remote and local sites, adapting the avatar's position and motion to fit the local AR environment. Yoon et al. [2] developed a deep-learning framework that learns to position an avatar in a distinct space while preserving the user's intended spatial context. Yang et al. [4] proposed visual guidance to help users move to positions that allow their avatars to be placed at positions that preserve the user's motion context. On the other hand, Wang et al. [3] and Choi et al. [21] introduced a smooth translation of the entire motion sequence to address dissimilarities. The former evaluated the most preferred transition style, while the latter addressed online full-body motion adaptation of avatars in similar spaces, but with slightly different furniture.

In a dissimilar telepresence environment, the user's gestures should also be modified to their avatar to preserve the motion context. A recent study by Kang et al. [7] developed a deep learning framework to retarget users' deictic motions to avatars in different environments. They generated pairs of deictic motion sequences targeting various locations, using these as inputs for user motions and the corresponding ground truth for avatar motions for training. While this target-based motion pairing approach is effective for deictic motions, it has limitations in training efficiency and action category scalability. First, the wide variety of combinations for motion pairs, dominated by target locations, significantly increases the training time. Second, unlike deictic motions with predictable head and hand orientations, free-form gestures, such as descriptive actions and natural motions during speech, do not follow consistent patterns. This inconsistency makes creating comparable motion pairs highly impractical. Our main focus is thus to translate general upper-body gestures without using a motion pairing procedure.

B. Motion Retargeting

Our motion translation problem is closely related to the motion retargeting problem, which modifies the original motion to match a new character or environment. Early studies framed motion retargeting as a spacetime optimization problem [22]–[24]. Recently, deep learning-based motion retargeting methods were developed to autonomously learn constraints, reduce the need for manual specification for each motion instance, and generalize across different skeletal structures [25]–[28]. Most recently, researchers [29] utilize the vision-language model to approach the motion retargeting process.

Closely related to our work are the studies that adapt the source characters' movements to different environmental contexts. Early works [30]–[36] aimed to avoid violations of physical laws and prevent collisions by formulating spatial relationship descriptors to model interactions between characters and objects. For example, Al-Asqhar et al. [31] designed a spatial relationship-based representation to adapt motion on large mesh structures. Tonneau et al. [33] further refined this approach to manage large deformations by repositioning contacts to preserve physical properties. Kim et al. [34] created a spatial map to translate human motions to virtual avatars, aligning them with similarly shaped objects. Recognizing that a human character comprises not just a skeleton but also skin, Jin et al. [35] introduced the aura mesh layer, which surrounds a character's surface, to maintain the original spatial relationships of the motion between skinned characters. Recent studies [37]–[39] adopted machine learning techniques to automatically generate spatial descriptors, building spatial correspondence between agents and objects and utilizing these neural fields for interactions with new objects.

Our work explores motion retargeting for environment variations. While prior research has addressed contact-based interactions and the adaptation of entire sequences or predefined sequence units of the original motion, we tackle a different aspect of motion retargeting: the real-time translation of upper-body gestures. We process motion frame-by-frame in real-time and model the coordinated movements between eyes, head, and hands to preserve upper-body gesture intention across different environments. This approach differs from contact-based motion retargeting, which primarily focuses on maintaining physical constraints such as foot placement and surface contact; instead, our method prioritizes the preservation of gestural semantics and spatial relationships critical for non-verbal communication, even when no explicit physical contacts are involved. This problem poses unique challenges as it requires interpreting and translating ongoing user movements into avatar poses, relying solely on current and past motion data without information about future movements.

C. Gesture Recognition

Real-time gesture recognition is crucial in designing natural human-computer interfaces, where accurately predicting gestures before completion enables responsive interaction. Recent deep learning approaches have advanced from offline classification [40]–[43] to online detection and early prediction [9], [44]–[46]. Molchanov et al. [44] demonstrated online gesture detection using 3D CNNs with connectionist temporal classification. Gupta et al. [45] and Long et al. [46] further explored this direction by explicitly modeling frame-level gesture progression to enable early prediction. Expanding the focus to broader topics on avatar gesture, notable works include gaze-mediated interaction [47], avatar retargeting effects on communication [48], and tunable gesture dynamics [49].

In the context of progression modeling, while these works have focused on hand gestures using dense input data such as video or hand skeleton sequences, our work instead models coordinated motion progression patterns between eyes, head, and hands using sparse tracking signals. Additionally, we aim to achieve user-invariant performance through training on single-user data, generalizing well across variations in handedness, eye dominance, and size while significantly reducing data collection efforts.

III. Dataset

We first introduce the dataset prepared to train and test our framework.

A. Motion Capture

To enable our framework to comprehensively deal with diverse gestures taken by general users, we carefully designed

the data capture stage to collect motions for pointing, gazing, touching, and free-form gestures along with their combinations with transitions. Specifically, we collected motions in the below categories:

- Pointing and gazing at a single target,
- Pointing at a single target with gaze shift (pointing and gazing at the first target, shifting only the gaze to a second target briefly, and then returning gaze to the original target),
- Explaining with pointing (considering a target as a person, making an explanatory gesture towards them, pointing at a different target, and returning to continue the explanation)
- Pointing and gazing at two targets in sequence,
- Transitioning between a pointing gesture and a free-form gesture (alternating between pointing at a single target and engaging in free-form gestures),
- Touching and gazing at a single target,
- Touching a single target with gaze shift,
- Explaining with touching,
- Touching and gazing at two targets in sequence,
- Transitioning between a touching gesture and a free-form gesture.

Training data was captured from a single right-handed, right-eye dominant male individual (height: 170cm, arm length: 74cm). Test data came from five individuals with diverse handedness, eye dominance, gender, heights, and arm lengths. Detailed information on the capture setup, target configurations, test subject characteristics, and data lengths are provided in the supplementary material.

Considering the significant variability in deictic gestures among individuals [50], we followed [7] to define a user's precise pointing or touching pose as the completion state (CS) for data collection purposes. In the context of pointing, CS is when the user accurately positions the tip of their right index finger at the target within their field of view, assuming the eye-finger ray (EFR) intersects with the target. For touching, CS is achieved when the tip of the index finger directly contacts the target.

During data acquisition, subjects were instructed to accurately perform the CS of the deictic gesture. The subject for the training data was asked to perform motions naturally, emphasizing a neutral style to ensure data neutrality. Other subjects were not given any guidance on their movements. All original motion data, initially at 60 fps, was downsampled to 30 fps. We then divided the sequences into multiple motion clips, each containing 45 frames and an overlap of 7 between consecutive clips.

B. Data Labeling

We defined three types of motion progression: Gaze, Deictic Gesture, and Idle. Each type ranges from 0 to 1, indicating the progression level. The Gaze increases from 0 when the subject begins moving toward a target and reaches 1 when both eye and head movements are stabilized on the target. Similarly, the Deictic Gesture is initially set at 0 and increases when the subject begins to point at or touch the target, reaching 1 when CS is achieved. The Idle is assigned a value of 1 when the subject maintains a standard upright posture and a value of 0 otherwise. We used Idle progression to differentiate movements from the standard upright posture, instead of categorizing it as a free-form gesture. Modeling the Idle progression reduces error accumulation in UGNet's auto-regression operations. It transitions from 0 to 1 after a 0.5-second delay when the subject returns to the upright posture. Intermediate values for each progression level are calculated using linear interpolation. Each sequence was then manually labeled according to these guidelines.

When viewed from a finite state machine perspective, we can more intuitively classify the progressive states in terms of Gaze, Deictic Gesture, and Idle. The default state, the standard upright posture, can be classified as (0,0,1). Free-form head and hand movements can be considered as (0,0,0). The completion of a deictic gesture can be classified as (1,1,0). Lastly, gazing accompanied by free-form hand movement can be classified as (1,0,0).

Note that our approach distinguishes from [7] by considering scenarios where head and hand movements are independent by assigning unique labels to each target. In contrast, [7] assumed that head and hand targets are aligned or that there is only a single target.

C. Data Augmentation

To enrich our training dataset, we manipulated the speed of the original motion sequences to generate additional data. Using monotone piecewise cubic interpolation [51], we adjusted the speed to 0.5 and 2 times their original values. We applied this adjustment to the categories of pointing and gazing at a single target, pointing and gazing at two targets in sequence, touching and gazing at a single target, and touching and gazing at two targets in sequence. Furthermore, we generated training data for left-handed subjects by mirroring the original motion sequences.

IV. Method

In this section, we will assume that the user performs deictic motions with their right hand. Figure 2 provides an overview of a framework composed of two distinct models: the MPNet and the UGNet. We initially extract the user's behavioral features from their gaze, head, right hand, index fingertip, and target position transformation history. The MPNet then extracts user-invariant features unaffected by the user's size, movement style, handedness, and eye dominance to predict motion progressions. Following this, the UGNet autoregressively predicts the transformations of the avatar's gaze, head, right hand, and index fingertip based on these progressions. Finally, an IK solver [52] is used to calculate the avatar's upper-body pose in the current frame.

A. Motion Progression Network (MPNet)

Input and Output. The MPNet is designed to predict the motion progression for diverse users by leveraging only a limited amount of training motion data. This generalization of

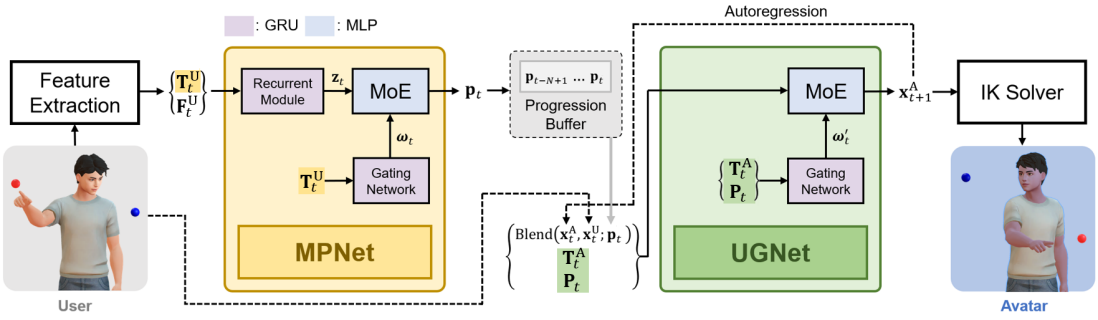


Fig. 2. Our neural network-based framework translates upper-body gestures to virtual avatars in dissimilar environments. The process is initiated by extracting behavioral features from the user, which are then analyzed by the MPNet to identify the user's motion progressions. The UGNet then regresses the avatar's next-frame end-effectors based on these progressions and a linear blend of the user's and avatar's current end-effectors, weighted by the progression values. Finally, an IK solver computes the avatar's upper-body pose for the next frame.

Fig. 3. (Left) The MPNet receives angle-based input features calculated using direction vectors defined at the root (O), head (H), right index fingertip (I), head target (T^H), and right-hand target (T^R). (Right) The UGNet outputs the gaze direction ($\mathbf{x}_t^E$), the transformations of the head ($\mathbf{x}_t^H$) and the right hand ($\mathbf{x}_t^R$), and the position of the right index fingertip ($\mathbf{x}_t^I$).

predicting motion progression is achieved through an angle-based and distance-based input representation that is user-invariant.

Before detailing input representations, we define reference frames and key directional vectors as shown in Figure 3 (Left). The head (H) joint is positioned at the midpoint between the eyes. The root (O) joint transformation is also positioned at the origin of the head joint transformation but in the standard upright posture. The base directions defining the input feature are as follows:

- $\overrightarrow{\mathrm{HT^H}}$: direction from head to head target,
- $\overrightarrow{\mathrm{HT^R}}$: direction from head to right-hand target,
- $\overrightarrow{\mathrm{HI}}$: direction from head to right-hand index fingertip,
- $\overrightarrow{\mathrm{EF}}$: forward vector of eye,
- $\overrightarrow{\mathrm{HF}}$: forward vector of head.

The MPNet's input, denoted as $\{\mathbf{f}_t, \mathbf{t}_t\} \in \mathbb{R}^{11}$ at the current time frame t consists of a behavioral feature $\mathbf{f}_t \in \mathbb{R}^6$ and a target feature $\mathbf{t}_t \in \mathbb{R}^5$. The behavioral feature, $\mathbf{f}_t = \{\mathbf{a}_t, \mathrm{d}_t\}$, includes an angle feature $\mathbf{a}_t \in \mathbb{R}^5$ and a distance feature $\mathrm{d}_t \in \mathbb{R}$. The angle feature, $\mathbf{a}_t = \{\mathrm{ET^H}, \mathrm{FT^H}, \mathrm{IT^R}, \mathrm{FI}, \mathrm{I}\}$, is derived from the base directions:

- $\mathrm{ET^H}$: angle between $\overrightarrow{\mathrm{EF}}$ and $\overrightarrow{\mathrm{HT^H}}$,
- $\mathrm{FT^H}$: angle between $\overrightarrow{\mathrm{HF}}$ and $\overrightarrow{\mathrm{HT^H}}$,
- $\mathrm{IT^R}$: angle between $\overrightarrow{\mathrm{HI}}$ and $\overrightarrow{\mathrm{HT^R}}$,
- FI: angle between $\overrightarrow{\mathrm{HF}}$ and $\overrightarrow{\mathrm{HI}}$,
- I: angle between $\overrightarrow{\mathrm{HI}}$ and world down vector.

The distance feature, d_t, indicates how far the right hand has moved from its initial position in the upright standing posture. We set $\mathrm{d}_t = 1$ when it has moved less than 0.25m and 0 otherwise.

The target feature $\mathbf{t}_t = \{\mathbf{t}_t^H, \mathbf{t}_t^R\}$ consists of the head target feature $\mathbf{t}_t^H = \{\theta_t^H, \phi_t^H\} \in \mathbb{R}^2$ and the right-hand target feature $\mathbf{t}_t^R = \{\theta_t^R, \phi_t^R, \mathrm{r}_t^R\} \in \mathbb{R}^3$: horizontal θ_t and vertical ϕ_t angles and a distance r_t to the position of the predefined target relative to the root. Following the approach of [7], we set $\mathrm{r}_t = 1$ when $\mathrm{r}_t > 1$ to ensure our model processes pointing gestures across all distances. This is based on the observation that pointing gestures exhibit distance-invariant characteristics [53]. Such an approach enables the model to recognize and generate pointing gestures across various ranges of distances.

The MPNet receives the sequence $\{\mathbf{F}_t, \mathbf{T}_t\} = \{\mathbf{f}_i, \mathbf{t}_i\}_{i=t-N+1}^{t} \in \mathbb{R}^{N\times 11}$ with a time window from $t-N+1$ to t. The window size N is empirically set to 30 ($= 1$ second for 30 fps) to capture temporal features of the input. The output $\hat{\mathbf{p}}_t = \{\hat{\mathrm{p}}_t^G, \hat{\mathrm{p}}_t^D, \hat{\mathrm{p}}_t^I\} \in \mathbb{R}^3$ of the MPNet comprises motion progressions for gaze $\hat{\mathrm{p}}_t^G$, deictic gesture $\hat{\mathrm{p}}_t^D$, and idle $\hat{\mathrm{p}}_t^I$ states.

Network Architecture. We adopt the Mixture of Expert (MoE) [54] model for the MPNet. A gating network dynamically adjusts the network's weights based on the given head and right-hand target position. The gating network, a two-layer gated recurrent unit (GRU), receives the target feature $\mathbf{t}$ and determines the blending coefficient $\omega \in \mathbb{R}^9$ ($\omega = \{\omega^{(i)}\}_{i=1}^{9}$) of MoE in the MPNet. The operation of a gating network is defined as:

$$\omega_t = \text{gating network}(\mathbf{t}_t) = \sigma(\text{GRU}(\mathbf{h}_{t-1}^{gat}, \mathbf{t}_t)) \quad (1)$$

where a softmax function σ makes the sum of blending coefficients 1, and $\mathbf{h}_{t-1}^{gat} \in \mathbb{R}^9$ is the hidden state of the second layer in the previous frame.

We feed $\{\mathbf{F}_t, \mathbf{T}_t\}$ to a single-layer GRU recurrent module. This module captures temporal features to compute a latent vector $\mathbf{z}_t$ for MoE to predict the motion progressions. The operation of the recurrent module is defined as:

$$\mathbf{z}_t = \text{recurrent module}(\{\mathbf{f}_t, \mathbf{t}_t\}) = \text{GRU}(\mathbf{h}_{t-1}^{rec}, \{\mathbf{f}_t, \mathbf{t}_t\}) \quad (2)$$

where $\mathbf{h}_{t-1}^{rec} \in \mathbb{R}^{24}$ is the hidden state of the layer in the previous frame.

The MPNet has 9 experts; with each expert $\alpha^{(i)}$ being a three-layer multilayer perceptron (MLP). Given $\mathbf{z}_t$, the MoE outputs the desired motion progression $\hat{\mathbf{p}}_t$. The operation of MoE is denoted as:

$$\hat{\mathbf{p}}_t = \text{MoE}(\mathbf{z}_t; \alpha(\omega_t)) \tag{3}$$

where the weights $\alpha(\omega_t) = \sum_{i=1}^{9} \omega_t^{(i)} \alpha^{(i)}$ are calculated as a linear combination of expert weights.

Training. For each motion clip, we updated the MPNet's weights from the reconstruction loss $\|\mathbf{p}_t - \hat{\mathbf{p}}_t\|_1$ each time and the time window advanced by one frame. The training process runs for 80 epochs. We used the AdamW [55] optimizer with a training batch size of 16. The initial learning rate was set at 1×10^{-4} and decreased exponentially by 0.975 after each epoch.

B. Upper-body Gesture Network (UGNet)

Input and Output. The UGNet is designed to generate upper-body gestures for avatars with different upper-body shapes. It uses an avatar-invariant output representation, defined as an end-effector transformation. This approach allows a straightforward generation of upper-body gestures for avatars with diverse skeletal structures with an IK solver. For network training and experimentation, a widely used 6D rotation representation [56] is employed.

The UGNet's input $\{\mathbf{x}_t, \mathbf{T}_t, \mathbf{P}_t\}$ at current time frame t consists of the end-effectors $\mathbf{x}_t \in \mathbb{R}^{24}$, a sequence of target feature $\mathbf{T}_t = \{\mathbf{t}_i\}_{i=t-N+1}^{t} \in \mathbb{R}^{N\times 5}$, and a sequence of motion progressions $\mathbf{P}_t = \{\mathbf{p}_i\}_{i=t-N+1}^{t} \in \mathbb{R}^{N\times 3}$. The end-effectors $\mathbf{x}_t = \{\mathbf{x}_t^{\text{E}}, \mathbf{x}_t^{\text{H}}, \mathbf{x}_t^{\text{R}}, \mathbf{x}_t^{\text{I}}\}$ include avatar's eye gaze direction $\mathbf{x}_t^{\text{E}} \in \mathbb{R}^3$, transformations for head $\mathbf{x}_t^{\text{H}} \in \mathbb{R}^9$ and the right hand $\mathbf{x}_t^{\text{R}} \in \mathbb{R}^9$, and position of right hand's index fingertip $\mathbf{x}_t^{\text{I}} \in \mathbb{R}^3$ (Figure 3). Here, $\mathbf{x}_t^{\text{H}}$ is defined relative to the root (O) joint transformation, while $\mathbf{x}_t^{\text{E}}$ and $\mathbf{x}_t^{\text{R}}$ are defined relative to the head joint transformation, and $\mathbf{x}_t^{\text{I}}$ is defined relative to the right-hand joint transformation. The UGNet's output $\hat{\mathbf{x}}_{t+1}$ represents the end-effectors at the next time frame.

Network Architecture. We also adopt the MoE model for the UGNet. However, unlike the MPNet's gating network which only uses the target feature, the UGNet's gating network includes motion progressions $\mathbf{p}$ to determine the blending coefficient $\omega' \in \mathbb{R}^9$ ($\omega' = \{\omega'^{(i)}\}_{i=1}^{9}$) of the MoE within the UGNet. The operation of the UGNet's gating network, a two-layer GRU, is defined as follows:

$$\omega_t' = \text{gating network}(\{\mathbf{t}_t, \mathbf{p}_t\}) = \sigma(\text{GRU}(\mathbf{h}_{t-1}^{gat}, \{\mathbf{t}_t, \mathbf{p}_t\})) \tag{4}$$

where $\mathbf{h}_{t-1}^{gat} \in \mathbb{R}^9$ is the hidden state of the second layer in the previous frame.

The UGNet has 9 experts with each expert $\beta^{(i)}$ being a three-layer MLP. Given end-effectors $\mathbf{x}_t$ in the current frame with $\mathbf{T}_t$ and $\mathbf{P}_t$, the UGNet's MoE outputs the desired end-effectors $\hat{\mathbf{x}}_{t+1}$ of the next frame. The operation of UGNet's MoE is denoted as:

$$\hat{\mathbf{x}}_{t+1} = \text{MoE}(\{\mathbf{x}_t, \mathbf{T}_t, \mathbf{P}_t\}; \beta(\omega_t')) \tag{5}$$

where the weights $\beta(\omega_t') = \sum_{i=1}^{9} \omega_t'^{(i)} \beta^{(i)}$ are blended by expert weights $\{\beta^{(i)}\}_{i=1}^{9}$.

Training. We utilize scheduled sampling [57] to alleviate the common issue of error accumulation in autoregressive prediction. A sampling probability p is defined for each training epoch. The output pose is used instead of directly inputting the ground truth pose data for the next timestep with a probability of $1-p$ once the output pose is generated. The training procedure is divided into three separate phases: supervised learning ($p = 1$), scheduled sampling (with p decaying), and autoregressive prediction ($p = 0$), with respective epochs of 30, 60, and 120. During the scheduled sampling phase, the sampling probability linearly diminishes to 0 with each successive training iteration.

For each motion clip, the weights of UGNet are updated frame-by-frame solely from the reconstruction loss $\|\mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1}\|_1$. For action categories including free-form movement, such as transitioning between a deictic gesture and a free-form gesture, and explaining with pointing or touching, we set $(\mathbf{x}_{t+1}^{\text{E}}, \mathbf{x}_{t+1}^{\text{H}}) \leftarrow (\mathbf{x}_t^{\text{E}}, \mathbf{x}_t^{\text{H}})$ if $\text{p}_t^{\text{G}} = 0$ and $(\mathbf{x}_{t+1}^{\text{R}}, \mathbf{x}_{t+1}^{\text{I}}) \leftarrow (\mathbf{x}_t^{\text{R}}, \mathbf{x}_t^{\text{I}})$ if $\text{p}_t^{\text{D}} = 0$ in the loss function. This facilitates the avatar to better follow the user's motion at runtime when the progression is zero.

We did not use velocity-based augmentation (Section III.C) to prevent the temporal continuity degradation which is crucial to UGNet's autoregression. We employed the AdamW optimizer with a training batch size of 32. The initial learning rate was set at 1×10^{-4} and exponentially decreased by a factor of 0.99 after each epoch.

C. Runtime

This section describes the procedure of real-time upper-body gesture translation using MPNet and UGNet. Although trained separately, two networks are utilized together during runtime.

Initially, the MPNet predicts the current frame's user motion progression $\mathbf{p}_t$ based on the user's behavioral feature sequence $\mathbf{F}_t^{\text{U}}$ and target feature sequence $\mathbf{T}_t^{\text{U}}$. The operation of the MPNet is defined as:

$$\mathbf{p}_t = \text{MPNet}(\{\mathbf{F}_t^{\text{U}}, \mathbf{T}_t^{\text{U}}\}). \tag{6}$$

Next, the UGNet generates the avatar's end-effectors $\mathbf{x}_{t+1}^{\text{A}}$ for the next frame by referencing the user's motion progression sequence $\mathbf{P}_t$ and the avatar's target sequence $\mathbf{T}_t^{\text{A}}$. We note that the UGNet does not directly use the avatar's end-effector pose at the current frame as its autoregressive input. Instead, it employs a blended value that combines this pose with the user's end-effector pose, based on the motion progression. The operation of the UGNet is defined as:

$$\mathbf{x}_{t+1}^{\text{A}} = \text{UGNet}(\{\text{Blend}(\mathbf{x}_t^{\text{A}}, \mathbf{x}_t^{\text{U}}; \mathbf{p}_t), \mathbf{T}_t^{\text{A}}, \mathbf{P}_t\}). \tag{7}$$

The key role of this motion blending is to seamlessly transmit the user's pose to the avatar when there is no observed motion progression. We can think of an example situation where the user's right hand is performing a free-form gesture. The progression value for a deictic gesture would be zero. UGNet should accurately reconstruct the user's hand motion, which is free-form. As the progression value increases

above zero, the UGNet then starts to produce the avatar's corresponding motion. Similarly, the avatar's head pose is set to be identical to the user's when the gaze progression is zero and starts to deviate from it as the progression increases. The motion blending is defined as:

$$\text{Blend}(\mathbf{x}_t^{\text{A}}, \mathbf{x}_t^{\text{U}}; \mathbf{p}_t) = \begin{bmatrix} (1-\text{b}^{\text{H}})\mathbf{x}_t^{\text{A,E}} + \text{b}^{\text{H}}\mathbf{x}_t^{\text{U,E}} \\ (1-\text{b}^{\text{H}})\mathbf{x}_t^{\text{A,H}} + \text{b}^{\text{H}}\mathbf{x}_t^{\text{U,H}} \\ (1-\text{b}^{\text{R}})\mathbf{x}_t^{\text{A,R}} + \text{b}^{\text{R}}\mathbf{x}_t^{\text{U,R}} \\ (1-\text{b}^{\text{R}})\mathbf{x}_t^{\text{A,I}} + \text{b}^{\text{R}}\mathbf{x}_t^{\text{U,I}} \end{bmatrix} \tag{8}$$

where b^{H}, representing the blending weights for the head joint, are adjusted linearly as follows:

$$\text{b}^{\text{H}} = \begin{cases} \min(\text{b}^{\text{H}} + \delta, 1), & \text{if } \text{p}^{\text{I}} \text{ and } \text{p}^{\text{G}} \text{ are under thresholds} \\ \max(\text{b}^{\text{H}} - \delta, 0), & \text{otherwise.} \end{cases} \tag{9}$$

The blending weight for the arm b^{R} is determined in the same manner with respect to p^{I} and p^{D}. We set $\delta = 0.05$.

For the avatar's left hand, its transformation matrix is determined to maintain the azimuth, altitude, and distance between the user's hand and head joint. This method allows natural-looking left-hand animations but does not prevent the hand from penetrating the torso. To resolve this, a collision geometry of an elliptical cylinder was created around the chest and the hand joint was projected to the collision boundary if penetration occurred.

Our framework shows fast inference times (about 6.7ms on an Intel i9 processor; MPNet: 2.8ms, UGNet: 2.4ms, IK: 1.5ms) and low memory requirements (2.24MB total; MPNet: 0.04MB, UGNet: 2.20MB), appropriate for real-time telepresence applications.

V. Technical Evaluation

We evaluate our framework quantitatively and qualitatively using our in-house dataset. The training utilized the motion data of a single subject who is 170cm tall, right-handed, and has right-eye dominance. All evaluations were performed on test subjects. For left-handed subjects, we used a network trained with the training subject's motion data with the symmetric transformation applied. All experiments were conducted on an Nvidia RTX 4090 GPU, using the PyTorch framework [58].

A. Evaluation Metric

Deictic Intention Preservation. The criteria for determining if an avatar has accurately maintained a user's deictic gesture is based on the error at the moment when the user reaches the CS. To assess the pointing accuracy, we used an angular metric which is invariant to the target distance. We calculated the horizontal error (HE [°]) and the vertical error (VE [°]) between the EFR vector and the vector extending from the head to the hand target. To evaluate touch accuracy, we measured the positional error (PE [cm]) between the hand target and the right index fingertip.

Movement Naturalness. As a metric for movement naturalness, we employed the Fréchet motion distance (FMD) proposed by [59], which measures the distance between the latent vectors of real and created motions. A lower FMD value indicates less difference between real and generated motions, suggesting a greater sense of naturalness in the generated motions.

To construct a motion latent space, we trained a two-layer convolutional autoencoder that reconstructs the movements of a subject's head and right-hand joints with 170cm height for one second (30 frames). To calculate FMD for right- and left-handed subjects, we conducted separate training for left-handed subjects by mirroring the original motion data. We used the movements of other subjects (161cm, 172cm, 173cm, 174cm, 180cm) as validation data. For right-handed subjects, the average per-frame position and rotation errors were 0.38cm and 0.97° in the training data, and 1.29cm and 3.90° in the validation data (161cm, 172cm, 180cm). For left-handed subjects, they were 0.38cm and 0.95° in the training data, and 1.57cm and 3.96° in the validation data (173cm, 174cm).

To evaluate the FMD, we created a test dataset by using a subset of the validation data. Specifically, we used each motion sequence in the validation data as user input for our test. The target positions from a randomly selected sequence within the same action category were assigned as the avatar's targets. We set the avatar's initial pose to an upright standing pose. We segmented the motion sequences, generated by each method, into segments of 30 frames. Each segment overlapped with the next by 15 frames (0.5s). Sequences utilized as user input were segmented in the same way as real motion. We input these segmented sequences into the encoder of a pre-trained convolutional autoencoder. This process allowed us to extract feature vectors, which we then used to calculate the FMD. We report the mean (± standard deviation) FMD scores over 10 trials.

B. Comparison with State-of-the-art Method

We compared our framework to the current SOTA method, AMNet (AngleNet and MotionNet) [7]. All categories (AC) enumerated in Section III were used to train our framework. For comparison purposes, we also retrained AMNet using the same settings described in [7], but only with data for deictic gestures (DG), where motion pairing was feasible. This includes pointing (or touching) and gazing at a single target, pointing (or touching) and gazing at two targets in sequence. We trained our framework on DG data as well.

The average training times were 260 minutes for the MPNet and 100 minutes for the UGNet in our framework using AC data; 170 minutes for the MPNet and 30 minutes for the UGNet in our framework using DG data; and 300 minutes for AMNet using DG data. The metric values averaged over test subjects are summarized in Table I.

Regarding deictic intention preservation (DIP), Ours_DG outperformed AMNet in VE and PE, while AMNet is slightly better at HE. This comparison is noteworthy as both models were trained on the same deictic gesture (DG) dataset. To assess the impact of dataset composition on performance, we compared Ours_DG with Ours_AC. Despite Ours_AC being trained on a significantly expanded dataset that includes free-form gestures, it maintained a deictic accuracy comparable to

TABLE I
COMPARISON OF OUR APPROACH WITH THE SOTA METHOD ON TEST SUBJECTS.

Method	DIP			FMD↓
	HE↓	VE↓	PE↓	
Ours_AC	2.37	1.57	3.08	**6.15 ± 0.86**
Ours_DG	2.72	**1.24**	**2.51**	21.37 ± 3.52
AMNet_DG	**1.77**	3.43	4.20	30.29 ± 2.40

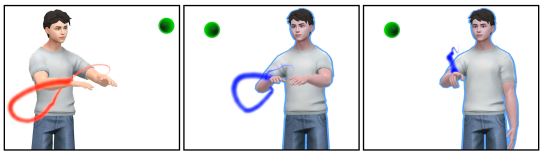


Fig. 4. The results of translating the user's upper-body gesture (free-form gesture) through our framework and AMNet, respectively. When the user makes a gesture toward the target (green sphere), our framework accurately translates the gesture toward the avatar's target, whereas AMNet produces a wiggling deictic gesture.

Ours_DG. This suggests that our framework design is robust and not constrained by action category scalability.

Concerning movement naturalness, Ours_AC exhibited a lower FMD than AMNet. As illustrated in Figure 4, AMNet, which was trained on DG data, struggles to accurately translate free-form gestures and different transitional movements that test subjects perform. In contrast, Ours_AC executes these tasks effectively. However, the significant underperformance of Ours_DG, compared to its performance with Ours_AC, indicates that DG data alone does not adequately support the generation of diverse and natural upper-body gestures. This highlights the importance of employing an unpaired dataset approach, which can accommodate complex movements such as free-form and transitional gestures where motion pairing is impractical. Intriguingly, Ours_DG still shows lower FMD values than AMNet. This result aligns with the avatar motions for deictic gestures (refer to the supplementary video), where our framework generates sharper and more natural-looking motions than AMNet.

C. Comparison with Alternative Network Architectures

We examine the performance of our two distinct networks, MPNet (GRU-MoE) and UGNet (MoE), with different network configurations to assess how various architectural components influence each model's effectiveness. Given the limited prior research on deep learning-based real-time upper-body gesture translation to virtual avatars in dissimilar environments, except for [7], we carefully established comparison models for both MPNet and UGNet.

We selected several network architectures for our comparative study, including MLP, Transformer [60], and diffusion models [61]. This selection is motivated by their proven efficacy and prominence. MLP serves as a fundamental yet powerful baseline, representing traditional feed-forward architectures. Transformer has gained significant traction in processing time-series data, which is crucial for our motion translation task. We incorporated diffusion models into UGNet

TABLE II
COMPARISON OF MODEL SIZE, TRAINING TIME, AND INFERENCE TIME FOR ALTERNATIVE NETWORK ARCHITECTURES.

Model	Network Architecture	Size	Training	Inference
MPNet	MLP	0.04MB	110m	0.2ms
	GRU-MoE (Ours)	0.04MB	260m	2.8ms
	Transformer	0.05MB	620m	2.6ms
UGNet	MLP	2.21MB	50m	0.3ms
	MoE (Ours)	2.20MB	100m	2.4ms
	Diffusion	2.62MB	150m	24.4ms
	Transformer	2.48MB	240m	2.9ms

to explore their potential in motion synthesis and upper-body gesture translation, given their success in generative tasks.

We ensured similar network sizes and utilized identical training protocols for fair comparison. Despite employing diverse architectures for MPNet and UGNet, we maintained consistency in the input and output specifications. Each architecture's size, training time, and inference time are reported in Table II. As the networks are trained at 30 fps, all proposed architectural combinations meet real-time operation without delay. Implementation specifics for each variant of MPNet and UGNet are as follows:

1) MPNet Architectures

- **MLP** model uses three linear layers and ELU as the activation function between the layers.
- **Transformer** model employs a Transformer encoder with one linear layer at both the input and output stages. We configured the encoder with four self-attention layers and four heads.

2) UGNet Architectures

- **MLP** model uses three linear layers and ELU as the activation function between the layers.
- **Diffusion** model follows the recently proposed autoregressive motion diffusion model (AMDM) [62]. Unlike sequence-based motion generation, AMDM generates motion on a frame-by-frame basis, which has been demonstrated to achieve sufficient performance with fewer denoising steps (10-50 steps). For real-time operation, we set the number of denoising steps to 30. While maintaining AMDM's architecture, we adjusted the hidden layer dimensions to match the size of other models in our comparison. Instead of following the conventional diffusion approach of gradually removing noise, we adopted the direct pose regression strategy commonly used in recent motion research [63]–[65] to improve training stability.
- **Transformer** model uses a Transformer encoder whose attention captures the correlations between the current end-effectors $\mathbf{x}_t$, the target feature $\mathbf{t}$, and motion progression $\mathbf{p}$ to predict the next frame's end-effector $\mathbf{x}_{t+1}$. There are two linear layers at the beginning to embed $\mathbf{x}$ and $\{\mathbf{t}, \mathbf{p}\}$ respectively, and one linear layer at the end of the architecture. Transformer encoder is configured with four self-attention layers and four heads.

As UGNet's performance heavily depends on the accuracy of motion progression, we first assessed which MPNet model most accurately predicts motion progression before evaluat-

TABLE III
COMPARATIVE STUDY OF ALTERNATIVE NETWORK ARCHITECTURES FOR MPNET ON TEST SUBJECTS.

MPNet	MPE↓	SS↓
MLP	0.056	0.16
GRU-MoE (Ours)	**0.054**	**0.05**
Transformer	0.057	0.10

TABLE IV
COMPARATIVE STUDY OF ALTERNATIVE NETWORK ARCHITECTURES FOR UGNET ON TEST SUBJECTS.

UGNet	DIP			DR			FMD↓
	HE↓	VE↓	PE↓	HL↓	VL↓	PL↓	
MLP	2.16	2.17	4.91	18.44	9.78	20.92	5.92 ± 0.55
MoE (Ours)	2.37	**1.57**	**3.08**	20.33	10.70	20.69	6.15 ± 0.86
Diffusion	3.54	2.07	3.77	**16.61**	**8.53**	**19.11**	7.19 ± 0.39
Transformer	**2.00**	2.62	4.46	30.25	17.15	28.31	$\mathbf{4.06 \pm 0.19}$

ing the entire framework's performance. We employed two evaluation metrics: motion progression error (MPE) and state stability (SS). MPE represents the average per-frame error between ground truth and predicted values, evaluating the model's overall accuracy. SS, on the other hand, measures how stably the motion progression (0,0,1) is maintained when the target changes from the default state. Specifically, SS is measured by calculating the average per-frame error between the predicted motion progression and (0,0,1) over one second after changing the target position from the default state. Ideally, the predicted motion progression should remain as (0,0,1) despite the target change. This measurement serves as a crucial indicator for assessing MPNet's stability and responsiveness in unpredictable situations not included in the training and test datasets. The SS measurement process and its significance are detailed in the supplementary video.

Table III provides metric values averaged over test subjects for different MPNet architectures. GRU-MoE model shows a slight improvement in MPE compared to other architectures. However, it demonstrates a substantial advantage in SS, indicating that our proposed model is much more robust to unexpected changes in target position. This enhanced stability can be attributed to two factors: the modular nature of MoE, where specialized experts handle different target scenarios effectively, and the lightweight sequential processing of GRU that enables quick adaptation to target changes. Based on these results, we selected GRU-MoE model for MPNet in our final framework.

We evaluated the performance of different UGNet configurations within our end-to-end framework with the MPNet fixed as GRU-MoE model. Table IV presents metric values averaged over test subjects. Regarding DIP, while MLP, MoE, Transformer models showed comparable performance in HE, Diffusion notably underperformed in this metric. MoE model particularly excelled in VE and PE. This suggests that accurately producing deictic gestures on height and distance is more challenging than azimuth, possibly due to a narrower range of values. MoE model's ability to factorize weights based on target features allowed it to adapt more effectively to various height and distance configurations, resulting in superior accuracy.

Concerning FMD, Transformer model outperformed other architectures. MLP and MoE models showed similar performance levels, while Diffusion model performed the poorest. As observed in the supplementary video, Diffusion model exhibits noticeable motion jittering. This instability can be attributed to two fundamental challenges. First, there is an architectural mismatch between our task and the Diffusion model's generative nature. Our framework requires precise and consistent motion translation while the Diffusion model learns probability distributions, leading to undesirable variations in poses. Second, our training data provides only single mappings per condition, limiting the model's ability to learn meaningful distributions and leading to poor temporal consistency and generalization. To effectively utilize Diffusion model in our context, we need to restructure the training data to include multiple pose variations and modify the architecture for stronger temporal consistency.

Visual inspection of generated animations revealed that Transformer model produced smoother motions than other models. However, despite its motion smoothness, we observed that Transformer model responded slowly to scenarios not encountered during training, such as sudden changes in target position (refer to the supplementary video).

To quantify this behavior, we introduce a new metric: dynamic responsiveness (DR). DR measures how swiftly different UGNet architectures adapt to abrupt target changes. We initialize the avatar in a CS state posture. We then abruptly change the target position and measure how quickly the motion achieves the CS state for the new target position over 1 second (30 frames). For pointing gestures, we calculate the average per-frame horizontal latency (HL [°/frame]) and vertical latency (VL [°/frame]) using the angle between the EFR vector and the vector extending from the head to the new hand target. For touching gestures, we calculate the average per-frame positional latency (PL [cm/frame]) based on the distance between the new target and the index fingertip. We emphasize that DR is specifically designed to evaluate the performance of UGNet alone, rather than the entire framework.

The results show that Diffusion and MLP models performed similarly, followed by MoE model, while Transformer model exhibited the lowest performance. Upon visually inspecting the animations, Diffusion, MLP, and MoE models showed negligible differences and produced human-like motions. In contrast, Transformer model not only displayed slower responsiveness but also generated motions that appeared robotic.

After a comprehensive evaluation considering DIP, FMD, and DR, we found that each architecture exhibited distinct strengths and weaknesses. Transformer model excelled in FMD but showed poor DR, a significant drawback when real-time usage in a telepresence scenario. MLP model performed well in DR and provided FMD comparable to MoE model, but fell short in DIP. Diffusion model showed good DR but underperformed in both FMD and DIP. MoE model, while not the top performer in every metric, consistently demonstrated satisfactory performance across all measures. It exhibited superior DIP, which might be important during real user-included scenarios to convey intended direction and gestures, particularly in VE and PE. It maintained competitive

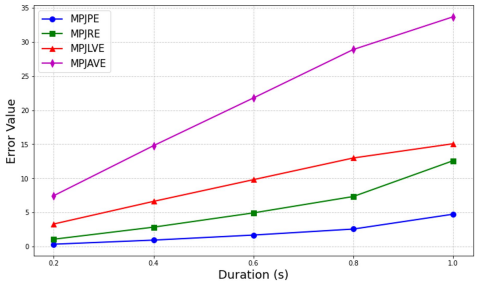


Fig. 5. Impact of missed tracking duration on motion quality. All metrics linearly increase with duration, indicating that tracking errors accumulate proportionally to the missed tracking time.

performance in FMD and DR.

Considering our application's accuracy, naturalness, and responsiveness requirements, we conclude that MoE model offers the most suitable balance for UGNet. Its consistent performance across all metrics, without significant weaknesses in any area, makes it the most versatile choice for our diverse requirements, especially from the system's perspective where real-time translation of motions within dissimilar environments should be possible.

D. Robustness to Noisy Tracking

In real-world MR telepresence scenarios, user motion tracking can be subject to various types of noise. We conducted experiments with three distinct noise cases to evaluate our framework's resilience to such perturbations: missed tracking (frame skip), occlusion (jitter), and continuous noise.

Missed tracking represents complete tracking failure, during which the system maintains the last tracked frame's values throughout the failure duration. Occlusion (jitter) represents occasional, high-magnitude changes in tracking data, typically resulting from sudden tracking errors due to occlusion. Continuous noise represents persistent background noise, often stemming from equipment limitations. Visual examples are available in the supplementary video.

We examined the framework's response to missed tracking by randomly introducing tracking failures (with a 30% probability per second) for various durations. During these periods, the system substituted the last available frame's values. We tested all possible target pairing cases within the same action category for each input motion sequence.

To quantify the impact of missed tracking, we computed the difference between motions generated with and without tracking failure. We employed four metrics calculated for the head and hand joints: mean per joint position error (MPJPE [cm]), mean per joint rotation error (MPJRE [°]), mean per joint linear velocity error (MPJLVE [cm/s]), and mean per joint angular velocity error (MPJAVE [°/s]). Higher error values indicate greater deviation from motions without tracking failure, suggesting decreased motion quality.

Figure 5 shows metric values averaged over test subjects for different durations of missed tracking. Error metrics increase with longer tracking failure durations. When tracking failure exceeded 0.8s, the generated animations showed notable motion artifacts as the system attempted to catch up with restored tracking data, a behavior attributable to MPNet's 1s input sequence length. However, for durations of 0.6s or less, the framework maintained acceptable animation quality.

We also evaluated the framework's performance under occlusion (jitter) and continuous noise conditions. As shown in the supplementary video, our experiments demonstrated that the framework maintained animation quality even under significant noise levels.

Our framework achieves robust performance under noisy conditions without incorporating additional training strategies such as noisy data augmentation or noise-specific loss terms. This inherent resilience can be attributed to three key architectural design choices. First, our two-stage approach provides effective noise buffering through motion progression, which serves as an intermediate representation that reduces end-to-end noise propagation. Second, we leverage user-invariant features, particularly the angle-based input representation trained into MPNet, which provides inherent robustness to noise through its scale and position-independent characteristics. Finally, our sequence-based approach utilizing MPNet's 30-frame window mitigates the impact of short-term noise.

E. Effectiveness of Input and Output Representation

In our experiments, subjects naturally had various physical characteristics. Figure 6 shows that MPNet can reliably predict motion progressions for test subjects, despite being trained with data from only one individual. This is possible thanks to MPNet's user-invariant input representation. Furthermore, as seen in Figure 7, the end-effectors of the avatar generated by UGNet can be adjusted to match differences in arm lengths between the user and the avatar by scaling the positions. For qualitative evaluation of animation results, we encourage readers to check our supplementary video.

VI. User Evaluation

We conducted an MR user evaluation to assess how our upper-body gesture translation framework facilitates remote interaction between distant users in dissimilar telepresence environments. Our study examined how different translation methods impact user experience and communication quality.

A. Methods

Our experiment included three conditions: baseline (**BS**) method, where the user's upper-body motion was copy-and-pasted to the avatar, SOTA method (**AMNet** [7]), and our method (**Ours**).

B. Measures

We measured two components using questionnaires adopted from a previous study [2]: Temple Presence Inventory (TPI) [66] and Message Understanding (MU) [67]. TPI analyzes the social presence between participants. MU assesses how well users understand messages received from their counterparts.

We included five custom questions that asked participants to rate the upper-body gestures of the counterpart's avatar:

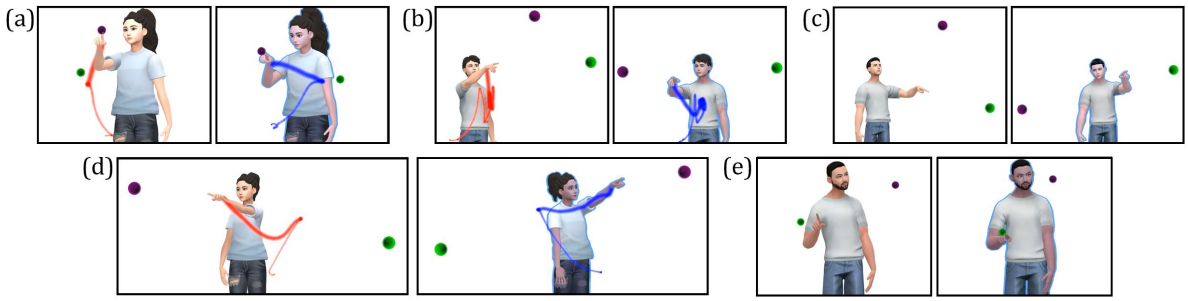
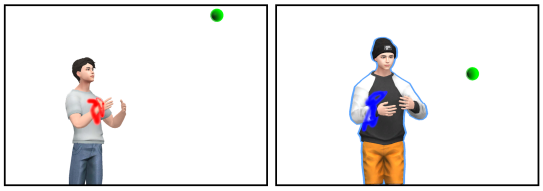
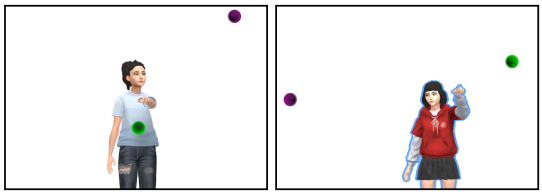


Fig. 6. Upper-body gestures (left) from various test subjects and avatar motions (right) translated in real-time by our framework. Two targets (green and purple spheres) have different positions for users and their corresponding avatars. (a) 161cm subject, touching and gazing at two targets in sequence. (b) 172cm subject, explaining with pointing. (c) 173cm subject, pointing at a single target with gaze shift. (d) 174cm subject, pointing and gazing at two targets in sequence. (e) 180cm subject, touching at a single target with gaze shift.



(a) User: 172cm, Avatar: 189cm (10% larger than user).



(b) User: 174cm, Avatar: 157cm (10% smaller than user).

Fig. 7. UGNet's end-effector output can be adjusted to accommodate size differences between the user and the avatar.

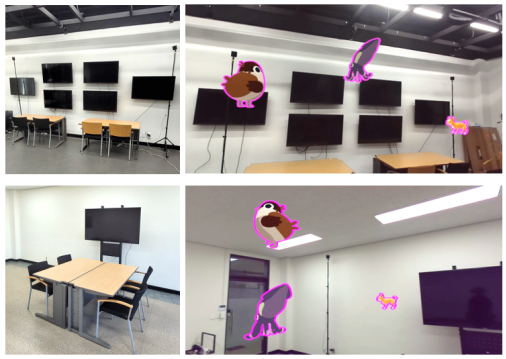


Fig. 8. Room perspective (left) and user egocentric (right) views of two remote spaces with differently placed virtual animals.

- Q1: the avatar's upper-body gestures helped me understand their intentions.
- Q2: the avatar's upper-body gestures aligned well with the spatial relationships of objects in my space.
- Q3: the avatar's upper-body gestures fit naturally within the context of our conversation.
- Q4: the avatar's upper-body gestures helped me pay attention to important subjects.
- Q5: the avatar's upper-body gestures helped me to understand the targets being explained better.

C. Task

We selected Charades as the experimental task based on three criteria: 1) it requires diverse upper-body gestures, including both deictic and free-form expressive movements; 2) it allows assessment of gesture translation between dissimilar spatial environments; and 3) it promotes natural interaction and communication between participants.

D. Procedure

Upon arrival at their respective spaces (shown in Figure 8), participants first signed consent forms and provided demographic information. After receiving instructions about the study structure and tasks, they were equipped with the HMD and hand gloves.

For the experiment, we prepared three sets of virtual animals, each containing three. These virtual animals were augmented at different locations in each participant's space to create dissimilar telepresence environments. Each condition used a different animal set, and both conditions and sets were randomized across participants. During each condition, participants followed this sequence: 1) one participant (the describer) selected and described an animal using upper-body gestures; 2) the other participant (the guesser) identified the animal by pointing at it; 3) upon correct identification, participants switched roles; and 4) this sequence was performed twice per condition.

The order of conditions was randomized for each participant. After completing each condition, participants filled out questionnaires evaluating their experience. Following all conditions, we conducted semi-structured interviews focusing on: 1) noticed differences between conditions, and 2) how these differences affected task performance. The total experiment duration was approximately 60 minutes. For additional details, please refer to the supplementary video.

E. Implementation

Our configuration comprises two identical setups deployed in separate remote locations. Each setup incorporates a ZED mini RGB-D camera paired with an HTC Vive Pro Eye headset, supporting MR visualization and enabling real-time occlusion interactions between physical and virtual objects.

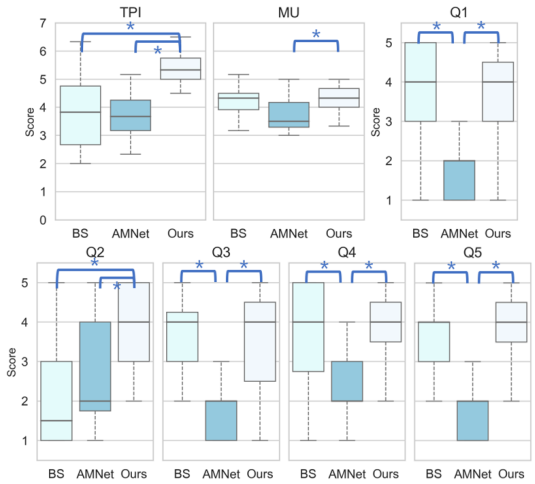


Fig. 9. Statistical results for measures between methods. Significance is marked with an asterisk (*).

To accurately capture the user's hand movements, the user is equipped with Vive trackers and Noitom Hi5 VR Gloves. We developed the system using the Unity3D engine (version 2020.3.9f1) and the SteamVR platform. The Photon Unity Network framework facilitates communication between systems at different locations. During experiments, operators at each location monitored their respective participants and changed target points in real-time.

F. Participants

An Institutional Review Board approved our study. We recruited 24 participants (12 dyads, 11 female, 13 male). Their age ranged from 23 to 34 years (M=28.270, SD=3.764). We collected height (156 to 185cm), arm length (63 to 74cm), handedness (23 right, 1 left), and dominant eye (16 right, 8 left). Regarding VR and MR experience, participants reported the following: 22 participants had prior experience with VR, while two participants had no experience. 13 participants had used MR before while 11 participants had no experience.

G. Results

We ran the Shapiro-Wilk test on the data in all conditions. For those that passed the assumption of normality, we chose repeated measures ANOVA. For those that violated it, we chose a non-parametric Friedman test to analyze the differences between conditions. Figure 9 presents the distribution of user responses.

For the TPI, the data passed the normality assumption, so we ran the test using repeated measures ANOVA. The results showed a significant effect by methods ($F(2,44)$=14.161, $p < 0.001$). The post-hoc analysis with Bonferroni correction revealed that Ours (M=5.240, SD=0.806) had significantly higher scores compared to both BS (M=3.934, SD=1.295) and AMNet (M=3.884, SD=0.998).

For the MU, the difference between methods showed significance under the non-parametric Friedman test ($\chi^2(2)$=7.724, $p < 0.021$, W=0.168). Post-hoc analysis using Conover's test with Bonferroni correction revealed that AMNet (M=3.689, SD=0.624) was significantly lower than Ours (M=4.283, SD=0.488), p=0.027. However, no significant differences were found between BS (M=4.203, SD=0.600) and AMNet (p=0.226) or between BS and Ours (p=1.000).

For the custom questions, the data did not meet normality assumptions across all measures, so we employed a non-parametric Friedman test. We then conducted post-hoc analysis using Conover's test with Bonferroni correction:

- Q1: $\chi^2(2)$=21.947, $p < 0.001$, W=0.477. Ours (M=3.652, SD=1.152) and BS (M=3.652, SD=1.301) had higher scores compared to AMNet (M=1.826, SD=0.717).
- Q2: $\chi^2(2)$=14.519, $p < 0.001$, W=0.316. There were differences between Ours (M=3.739, SD=1.264) and BS (M=2.130, SD=1.359), as well as between Ours and AMNet (M=2.652, SD=1.265).
- Q3: $\chi^2(2)$=18.779, $p < 0.001$, W=0.408. Ours (M=3.522, SD=1.238) and BS (M=3.565, SD=1.343) had higher scores compared to AMNet (M=1.826, SD=0.650).
- Q4: $\chi^2(2)$=16.971, $p < 0.001$, W=0.369. Ours (M=3.976, SD=0.968) and BS (M=3.565, SD=1.409) had higher scores compared to AMNet (M=2.304, SD=1.020).
- Q5: $\chi^2(2)$=26.000, $p < 0.001$, W=0.565. Ours (M=3.826, SD=1.072) and BS (M=3.609, SD=1.076) had higher scores compared to AMNet (M=1.870, SD=0.920).

H. Discussion

Our experimental results demonstrate that effective avatar-mediated telepresence requires two key capabilities: supporting a wide range of upper-body gestures and accurately translating these gestures in dissimilar environments. Our analysis examines how different methods address these requirements and their impact on user experience.

First, the ability to translate diverse upper-body gestures significantly impacts communication quality. The MU scores showed that both Ours and BS methods, which support free-form and explanatory gestures, enabled significantly better message understanding than AMNet, which is limited to only deictic gestures. This limitation was particularly evident in participants' feedback, with multiple participants explicitly stating "I cannot understand what my partner was doing" (P10, P14, P18). Other participants noted the restrictive nature of AMNet's gesture translation, commenting "I was describing animals differently but my partner kept on saying that I was doing the same gesture" (P21, P22). During AMNet conditions, participants observed that their avatars repeatedly made similar pointing motions, with one arm often remaining static. These observations align with AMNet's consistently lower scores across Q1, Q3, Q4, and Q5, which evaluate gesture effectiveness in conveying intentions, naturalness, attention direction, and explanation quality.

Second, our framework's ability to adjust virtual avatar movements in dissimilar environments proved crucial for maintaining effective communication. This capability to translate gestures while preserving their intended meaning across different spatial contexts was reflected in the highest TPI scores for our method, indicating superior social presence

Fig. 10. Screenshots of commerce scenario in our MR telepresence.

compared to both BS and AMNet. The effectiveness of our gesture translation approach was particularly evident in Q2 responses, where our method significantly outperformed both alternatives in terms of spatial relationship perception. As P17 noted, "the sense of communication felt stronger when what I was seeing matched what the other person was seeing." At the same time, P23's experience with BS highlighted the problems of poor spatial adjustment: "I had to question myself whether I am explaining correctly since my collaborating partner was pointing at different directions all the time."

Based on these findings, we identify clear trade-offs in existing approaches. The BS method supports free-form gestures but fails to provide appropriate spatial adjustment, leading to confusion in spatial relationships. AMNet attempts to handle spatial differences but significantly restricts users' gesture expression by limiting it to only deictic gestures. Our method addresses these limitations by successfully combining both capabilities, supporting diverse upper-body gestures while appropriately translating them across dissimilar spaces. These results indicate that considering gesture expressiveness and spatial translation is crucial for designing effective MR avatar-mediated telepresence systems.

VII. Application

We envision that our translation framework will be widely used in MR avatar-mediated telepresence. To illustrate its potential, we present two scenarios in educational and commercial settings. Live demonstrations of these scenarios are included in the supplementary video.

A. Scenarios

Education. User X learns from user Y about two planets in the solar system, Earth and Jupiter. In each space, these planets are virtually augmented to the sizes and locations desired by both users. As they take turns using a variety of upper-body gestures to explain about the planets while viewing each other's avatars, our system generates movements for the avatars that are suitable for the placement of users and planets within the remote space (refer to Figure 1).

Commerce. User Y wants to buy clothing and seeks User X's opinion for this purpose. User Y has the physical item, while User X possesses a virtual 3D replica of it. User X has augmented this product to be larger than its actual size for closer inspection. Figure 10 shows that when users touch specific clothing parts (the hat), their avatars take the appropriate motions. This enables users to convey the focus of their discussion accurately.

VIII. Limitation

In this section, we discuss several limitations of our method. A primary limitation stems from the presumption of predetermined targets and established object correspondences. In real-world MR telepresence scenarios, the number of target objects may vary dynamically, requiring accurate real-time identification of the user's intended targets and their correlations.

Another limitation concerns the scope of gesture translation. Our approach primarily handles gaze-based deictic gestures with clear targets, limiting its applicability to implicit targets or complex non-verbal gestures (e.g., hand-switching, hand-with-hand interactions). It also lacks support for physical object manipulation and contact-based interactions. Future work could explore more generalized gesture representations and incorporate physics-based approaches to enhance motion realism by preserving physical properties like tempo and speed, rather than relying solely on IK solvers.

The third limitation relates to labeling motion progressions in the training data. We applied heuristic rules for manual labeling, but automatic extraction of motion progressions would greatly improve development efficiency and enable more nuanced motion translation.

Finally, incorporating lower-body movements for full-body avatars could enable more dynamic telepresence beyond stationary interactions. This extension presents challenges in avatar locomotion and interactions with real-world furniture (e.g., desks, chairs), requiring motion planning that complies with physical constraints.

IX. Conclusion

We presented a neural network-based framework that translates upper-body gestures into virtual avatars for MR telepresence between dissimilar spaces. We validated that the MPNet can extract user-invariant characteristics of upper-body gestures to infer the progressions of user movements. The UGNet then reliably generates the corresponding avatar's motions using these progressions in an autoregressive manner. Compared to the SOTA method that requires paired motion datasets, our unpaired approach reduces training time, broadens the range of action categories, and results in more lifelike avatars' movements. Our framework is designed specifically for real-time MR telepresence and has been demonstrated through user evaluation.

Acknowledgement

This work was supported by NRF STEAM Project (RS-2024-00454458), IITP (2022-0-00566), and metaverse support program (IITP-2025-RS-2022-00156435), supported by MSIT, Korea.

REFERENCES

[1] 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 *2020 IEEE conference on virtual reality and 3D user interfaces (VR)*. IEEE, 2020, pp. 353–362.

[2] L. Yoon, D. Yang, J. Kim, C. Chung, and S.-H. Lee, "Placement retargeting of virtual avatars to dissimilar indoor environments," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 3, pp. 1619–1633, 2020.

[3] 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 Transactions on Visualization and Computer Graphics*, vol. 28, no. 11, pp. 3705–3714, 2022.

[4] D. Yang, J. Kang, T. Kim, and S.-H. Lee, "Visual guidance for user placement in avatar-mediated telepresence between dissimilar spaces," *IEEE Transactions on Visualization and Computer Graphics*, 2024.

[5] J. E. S. Grønbæk, K. Pfeuffer, E. Velloso, M. Astrup, M. I. S. Pedersen, M. Kjær, G. Leiva, and H. Gellersen, "Partially blended realities: Aligning dissimilar spaces for distributed mixed reality meetings," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023, pp. 1–16.

[6] D. I. Fink, J. Zagermann, H. Reiterer, and H.-C. Jetter, "Re-locations: Augmenting personal and shared workspaces to support remote collaboration in incongruent spaces," *Proceedings of the ACM on Human-Computer Interaction*, vol. 6, no. ISS, pp. 1–30, 2022.

[7] 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 *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2023, pp. 885–893.

[8] H.-S. Moon, Y.-C. Liao, C. Li, B. Lee, and A. Oulasvirta, "Real-time 3d target inference via biomechanical simulation," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–18.

[9] C. Chung and S.-H. Lee, "Continuous prediction of pointing targets with motion and eye-tracking in virtual reality," *IEEE Access*, 2024.

[10] Y. Wei, R. Shi, D. Yu, Y. Wang, Y. Li, L. Yu, and H.-N. Liang, "Predicting gaze-based target selection in augmented reality headsets based on eye and head endpoint distributions," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023, pp. 1–14.

[11] A. Maimone and H. Fuchs, "Encumbrance-free telepresence system with real-time 3d capture and display using commodity depth cameras," in *2011 10th IEEE International Symposium on Mixed and Augmented Reality*. IEEE, 2011, pp. 137–146.

[12] 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 *2013 IEEE Virtual Reality (VR)*. IEEE, 2013, pp. 23–26.

[13] S. Beck, A. Kunert, A. Kulik, and B. Froehlich, "Immersive group-to-group telepresence," *IEEE transactions on visualization and computer graphics*, vol. 19, no. 4, pp. 616–625, 2013.

[14] Y. Pan and A. Steed, "A gaze-preserving situated multiview telepresence system," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2014, pp. 2173–2176.

[15] T. Pejsa, J. Kantor, H. Benko, E. Ofek, and A. Wilson, "Room2room: Enabling life-size telepresence in a projected augmented reality environment," in *Proceedings of the 19th ACM conference on computer-supported cooperative work & social computing*, 2016, pp. 1716–1725.

[16] S. Orts-Escolano, C. Rhemann, S. Fanello, W. Chang, A. Kowdle, Y. Degtyarev, D. Kim, P. L. Davidson, S. Khamis, M. Dou *et al.*, "Holoportation: Virtual 3d teleportation in real-time," in *Proceedings of the 29th annual symposium on user interface software and technology*, 2016, pp. 741–754.

[17] B. Thoravi Kumaravel, F. Anderson, G. Fitzmaurice, B. Hartmann, and T. Grossman, "Loki: Facilitating remote instruction of physical tasks using bi-directional mixed-reality telepresence," in *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, 2019, pp. 161–174.

[18] N. H. Lehment, D. Merget, and G. Rigoll, "Creating automatically aligned consensus realities for ar videoconferencing," in *2014 IEEE international symposium on mixed and augmented reality (ISMAR)*. IEEE, 2014, pp. 201–206.

[19] D. Kim and W. Woo, "Edge-centric space rescaling with redirected walking for dissimilar physical-virtual space registration," in *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2023, pp. 829–838.

[20] D. Jo, K.-H. Kim, and G. J. Kim, "Spacetime: adaptive control of the teleported avatar for improved ar tele-conference experience," *Computer Animation and Virtual Worlds*, vol. 26, no. 3-4, pp. 259–269, 2015.

[21] 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, no. 2. Wiley Online Library, 2023, pp. 13–24.

[22] M. Gleicher, "Retargetting motion to new characters," in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, 1998, pp. 33–42.

[23] J. Lee and S. Y. Shin, "A hierarchical approach to interactive motion editing for human-like figures," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 1999, pp. 39–48.

[24] S. Tak and H.-S. Ko, "A physically-based motion retargeting filter," *ACM Transactions on Graphics (ToG)*, vol. 24, no. 1, pp. 98–117, 2005.

[25] R. Villegas, J. Yang, D. Ceylan, and H. Lee, "Neural kinematic networks for unsupervised motion retargetting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8639–8648.

[26] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[27] K. Aberman, P. Li, D. Lischinski, O. Sorkine-Hornung, D. Cohen-Or, and B. Chen, "Skeleton-aware networks for deep motion retargeting," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 62–1, 2020.

[28] R. Villegas, D. Ceylan, A. Hertzmann, J. Yang, and J. Saito, "Contact-aware retargeting of skinned motion," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9720–9729.

[29] H. Zhang, Z. Chen, H. Xu, L. Hao, X. Wu, S. Xu, Z. Zhang, Y. Wang, and R. Xiong, "Semantics-aware motion retargeting with vision-language models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[30] E. S. Ho, T. Komura, and C.-L. Tai, "Spatial relationship preserving character motion adaptation," in *ACM SIGGRAPH 2010 papers*, 2010, pp. 1–8.

[31] R. A. Al-Asqhar, T. Komura, and M. G. Choi, "Relationship descriptors for interactive motion adaptation," in *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2013, pp. 45–53.

[32] E. S. Ho and H. P. Shum, "Motion adaptation for humanoid robots in constrained environments," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3813–3818.

[33] S. Tonneau, R. A. Al-Ashqar, J. Pettré, T. Komura, and N. Mansard, "Character contact re-positioning under large environment deformation," in *Computer Graphics Forum*, vol. 35, no. 2. Wiley Online Library, 2016, pp. 127–138.

[34] Y. Kim, H. Park, S. Bang, and S.-H. Lee, "Retargeting human-object interaction to virtual avatars," *IEEE transactions on visualization and computer graphics*, vol. 22, no. 11, pp. 2405–2412, 2016.

[35] T. Jin, M. Kim, and S.-H. Lee, "Aura mesh: Motion retargeting to preserve the spatial relationships between skinned characters," in *Computer Graphics Forum*, vol. 37, no. 2. Wiley Online Library, 2018, pp. 311–320.

[36] T. Jin and S.-H. Lee, "Interaction motion retargeting to highly dissimilar furniture environment," in *Proceedings of the 18th annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2019, pp. 1–2.

[37] A. Simeonov, Y. Du, A. Tagliasacchi, J. B. Tenenbaum, A. Rodriguez, P. Agrawal, and V. Sitzmann, "Neural descriptor fields: Se (3)-equivariant object representations for manipulation," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6394–6400.

[38] Z. Huang, J. Xu, S. Dai, K. Xu, H. Zhang, H. Huang, and R. Hu, "Nift: Neural interaction field and template for object manipulation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1875–1881.

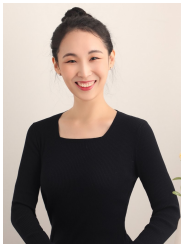
[39] Z. Huang, H. Xu, H. Huang, C. Ma, H. Huang, and R. Hu, "Spatial and surface correspondence field for interaction transfer," *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, vol. 43, no. 4, pp. 83:1–83:12, 2024.

[40] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, "Hand gesture recognition with 3d convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2015, pp. 1–7.

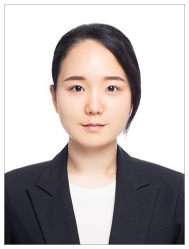
[41] P. Molchanov, S. Gupta, K. Kim, and K. Pulli, "Multi-sensor system for driver's hand-gesture recognition," in *2015 11th IEEE international*

conference and workshops on automatic face and gesture recognition (FG), vol. 1. IEEE, 2015, pp. 1–8.
[42] Q. Miao, Y. Li, W. Ouyang, Z. Ma, X. Xu, W. Shi, and X. Cao, "Multimodal gesture recognition based on the resc3d network," in *Proceedings of the IEEE international conference on computer vision workshops*, 2017, pp. 3047–3055.
[43] P. Narayana, R. Beveridge, and B. A. Draper, "Gesture recognition: Focus on the hands," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5235–5244.
[44] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, "Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4207–4215.
[45] V. Gupta, S. K. Dwivedi, R. Dabral, and A. Jain, "Progression modelling for online and early gesture detection," in *2019 International Conference on 3D Vision (3DV)*. IEEE, 2019, pp. 289–297.
[46] H. Long, S. Wang, H. Hu, Y. Wang, H. Xu, and H. Wang, "Mvmsfn: A multi-view and multi-scale fusion network for online detection of heterogeneous gestures," in *2024 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2024, pp. 1–7.
[47] A. Qureshi, C. Peters, and I. Apperly, "Interaction and engagement between an agent and participant in an on-line communication paradigm as mediated by gaze direction," in *Proceedings of the 2013 Inputs-Outputs Conference: An Interdisciplinary Conference on Engagement in HCI and Performance*, 2013, pp. 1–4.
[48] S. Nyatsanga, D. Roble, and M. Neff, "The impact of avatar retargeting on pointing and conversational communication," *IEEE Transactions on Visualization and Computer Graphics*, 2025.
[49] M. Neff, "Tunable tension for gesture animation," in *Proceedings of the 22nd acm international conference on intelligent virtual agents*, 2022, pp. 1–8.
[50] K. Plaumann, M. Weing, C. Winkler, M. Müller, and E. Rukzio, "Towards accurate cursorless pointing: the effects of ocular dominance and handedness," *Personal and Ubiquitous Computing*, vol. 22, pp. 633–646, 2018.
[51] F. N. Fritsch and J. Butland, "A method for constructing local monotone piecewise cubic interpolants," *SIAM journal on scientific and statistical computing*, vol. 5, no. 2, pp. 300–304, 1984.
[52] Root-Motion, "Final-ik," 2017.
[53] S. Mayer, K. Wolf, S. Schneegass, and N. Henze, "Modeling distant pointing for compensating systematic displacements," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015, pp. 4165–4168.
[54] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural computation*, vol. 3, no. 1, pp. 79–87, 1991.
[55] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2018.
[56] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5745–5753.
[57] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," *Advances in neural information processing systems*, vol. 28, 2015.
[58] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.
[59] S. Park, D.-K. Jang, and S.-H. Lee, "Diverse motion stylization for multiple style domains via spatial-temporal graph-based generative model," *Proceedings of the ACM on computer graphics and interactive techniques*, vol. 4, no. 3, pp. 1–17, 2021.
[60] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
[61] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.
[62] Y. Shi, J. Wang, X. Jiang, B. Lin, B. Dai, and X. B. Peng, "Interactive character control with auto-regressive motion diffusion models," *ACM Transactions on Graphics (TOG)*, vol. 43, no. 4, pp. 1–14, 2024.
[63] G. Tevet, S. Raab, B. Gordon, Y. Shafir, D. Cohen-Or, and A. H. Bermano, "Human motion diffusion model," in *International Conference on Learning Representations*, 2023.
[64] Y. Du, R. Kips, A. Pumarola, S. Starke, A. Thabet, and A. Sanakoyeu, "Avatars grow legs: Generating smooth human motion from sparse tracking inputs with diffusion model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 481–490.
[65] T. Van Wouwe, S. Lee, A. Falisse, S. Delp, and C. K. Liu, "Diffusionposer: Real-time human motion reconstruction from arbitrary sparse sensors using autoregressive diffusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.
[66] M. Lombard, T. B. Ditton, and L. Weinstein, "Measuring presence: the temple presence inventory," in *Proceedings of the 12th annual international workshop on presence*. International Society for Presence Research Los Angeles, CA, 2009, pp. 1–15.
[67] C. Harms and F. Biocca, "Internal consistency and reliability of the networked mindsmeasure of social presence," 2004. [Online]. Available: https://api.semanticscholar.org/CorpusID:142957981

Jiho Kang is a Ph.D. candidate with the Graduate School of Culture Technology at KAIST. He received an M.S. degree in Culture Technology from KAIST, Korea, in 2023 and a B.S. degree in Electrical Engineering from Handong Global University, Korea, in 2021. His research interests include human animation and MR telepresence.



Taehei Kim is a Ph.D. candidate with the Graduate School of Culture Technology in KAIST. She received an M.S. degree in Culture Technology from KAIST, Korea, in 2021 and a B.S. degree in Asian Studies and Computer Science from Yonsei University, Korea. Her research interest lies in adaptive MR telepresence systems.



Hyeshim Kim is a M.S. candidate with the Graduate School of Culture Technology in KAIST. She received a B.S. degree in Interaction Design from Korea National University of Arts, Korea, in 2020. Her research interests include MR telepresence.

Sung-Hee Lee is a Professor with the Graduate School of Culture Technology at KAIST. His research interests include autonomous human animation, avatar motion generation, and human modeling. He received a Ph.D. degree in Computer Science from the University of California, Los Angeles, USA, in 2008, and a B.S. and M.S. degree in Mechanical Engineering from Seoul National University, Korea, in 1996 and 2000, respectively.