Part-wise Translation of Upper-body Gestures via Real-time Target Estimation for Virtual Avatars in Dissimilar Environments

Jiho Kang, Jihun Shin, Kwang Bin Lee, Dong Hwan Shin, Yuna Choi, and Sung-Hee Lee

***Abstract*—In extended reality telepresence, virtual avatars must accurately convey users' intent across dissimilar environments. We propose a novel neural network framework for real-time, part-wise translation of upper-body gestures to virtual avatars in heterogeneous environments. Our approach advances beyond prior methods in three key aspects: (1) real-time estimation of users' interaction targets; (2) unsupervised learning of target-agnostic motion features that better preserve fine-grained motion details; and (3) part-wise translation in which the head and individual hands can independently reference different targets simultaneously. Together, these advances allow users to select arbitrary interaction targets from multiple candidates and reference different targets concurrently with different body parts, while enabling avatars to better preserve users' motion details. These capabilities are realized through two novel networks: a Target Estimation Network that interprets sparse user tracking signals to infer target positions for the head and hands, and a Body-Part Gesture Translation Network that autoregressively generates avatar poses conditioned on these targets. Despite being trained on data from a single individual, our framework translates natural upper-body gestures across diverse users and avatars. We validate the effectiveness of our approach through comparisons with state-of-the-art methods, qualitative animation results, and a user evaluation in telepresence scenarios.**

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

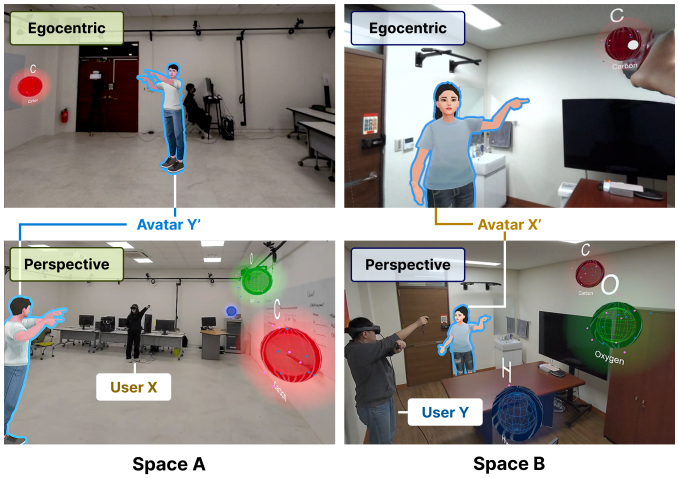


Fig. 1. User egocentric views (top) and room perspectives (bottom) of our XR telepresence system in two dissimilar spaces A (left) and B (right). Avatar X′ in space B represents user X from space A, and avatar Y′ in space A represents user Y from space B. Virtual avatars and objects are augmented at different locations and scales due to differences in room size and furniture layout. User X points at the green sphere with the left hand, looks at avatar Y′, and makes an explanatory gesture toward avatar Y′ with the right hand. User Y points at the red sphere with the right hand, points at the blue sphere with the left hand, and looks at avatar X′. Our system estimates each user's interaction targets in real-time and translates upper-body gestures in a part-wise manner, enabling seamless bi-directional interaction between remote users.

I. Introduction

Extended reality (XR) avatar-mediated telepresence is an immersive platform that augments users' avatars in each other's space, making users feel as if they are in the same space. A key consideration in implementing such systems is whether virtual avatars can accurately convey users' intentions despite remote users being in different environments. Researchers have developed techniques such as placing virtual avatars according to interaction context [1] and aligning spaces to create shared spaces [2]. In addition to these approaches, accurate gesture translation is critical for conveying user intentions across dissimilar environments. For example, simply replicating deictic gestures onto an avatar can fail to preserve communicative intent when gesture targets are positioned differently across environments.

Figure 1 illustrates that gesture translation in XR telepresence between dissimilar environments is a challenging problem. We define spatial dissimilarity as environments containing the same types of objects but with different positions, shapes, and sizes, where objects correspond only categorically rather than through any spatial alignment (classified as a *categorical transformation* by Wong et al. [3]). We assume one-to-one correspondence between user and avatar targets, so that estimating which target the user is gesturing directly identifies the corresponding avatar target. Target objects may include other users' avatars, displays, and virtual objects. These spatial dissimilarities arise naturally in XR, as each user arranges virtual and physical objects to fit their own room layout, which can change the relative ordering of objects across spaces and preclude any affine spatial correspondence.

Under this setting, gesture translation poses several key challenges. First, gesture translation is inherently a coupled problem requiring both target estimation and motion retargeting. Target estimation must identify what each body part is referencing based on the user's motion trajectory, while motion retargeting must generate avatar poses that accurately convey these references. Second, the system must handle diverse gesture types beyond deictic gestures, including explanatory and descriptive gestures, where each body part may indepen-

Jiho Kang, Jihun Shin, Kwang Bin Lee, Dong Hwan Shin, Yuna Choi, and Sung-Hee Lee (corresponding author) are with KAIST. E-mail: {jhkang0408, sunghee.lee}@kaist.ac.kr

dently perform different types of gestures toward different targets. Third, avatar motion must remain stable even when target estimation is uncertain or fluctuating, preventing jittery artifacts from propagating to the generated motion. Moreover, real-time application demands immediate, frame-level generation of avatar poses without access to future information, requiring a fast and lightweight framework that can maintain synchronization between user and avatar motions without temporal delay. Last, the system must generalize across diverse users with varying physical characteristics and across different skeletal avatars.

To address these challenges, we propose two separate deep neural networks that decompose the gesture translation problem into target estimation and motion generation: the Target Estimation Network (TENet) and the Body-Part Gesture Translation Network (BP-GTNet). Given that collecting motion data from diverse users is costly and often impractical for real-world deployment, we design both networks to be trained on motion sequences from only a few individuals (even a single person in this work), while ensuring generalization to diverse users and avatars.

TENet takes user-invariant behavioral features extracted from tracking signal history, along with candidate target positions, as input. Rather than directly regressing target positions, it computes posterior distributions over all candidate target combinations for the head, right hand, left hand using a conditional normalizing flow. This probabilistic formulation naturally represents uncertainty that is broad early in a gesture and sharpens as motion progresses, enabling robust inference even from partial or ambiguous movements.

BP-GTNet autoregressively generates avatar motions conditioned on the targets estimated by TENet. It learns target-agnostic (independent of target positions) motion features in an unsupervised manner, capturing detailed motion characteristics faithfully. To this end, it exploits a dual end-effector representation where the encoder and decoder operate on different representations, enabling target-agnostic encoding and target-conditioned decoding. The encoder extracts target-agnostic motion features from the user's tracking signals, while the decoder transforms these features into avatar-invariant outputs conditioned on the determined targets, including gaze direction, head and hand transformations, and fingertip positions. These outputs serve as end-effectors for inverse kinematics (IK), enabling application across different avatar structures. BP-GTNet further enables part-wise translation through a dedicated training scheme that ensures independent controllability of each body part, allowing simultaneous gesturing toward different targets while maintaining stable avatar motion even when target estimates are uncertain or fluctuating.

The prior data-driven method [4] maps user motions to manually-labeled sparse gesture categories and generates avatar poses from these intermediate representations. However, it has three key limitations: it assumes that ground-truth target positions are given, which prevents deployment in real systems where such information is unavailable; it compresses diverse motion patterns into sparse categories, discarding details such as gesture dynamics and curvature that are essential for realistic avatar animation; and it lacks part-wise controllability across body parts, failing to handle common scenarios where, for example, one hand points at an object while the other makes a descriptive gesture toward a conversational partner. Our two novel networks address all three limitations:

- TENet performs real-time target estimation from sparse tracking signals, eliminating the need for ground-truth targets.
- BP-GTNet learns target-agnostic motion features in an unsupervised manner, better preserving the richness of user motion.
- BP-GTNet's dedicated training scheme enables independent control of each body part, supporting simultaneous gestures toward different targets.

To the best of our knowledge, this is the first data-driven study that simultaneously addresses real-time target estimation and part-wise gesture translation for virtual avatars in dissimilar telepresence environments. We validated our method through quantitative and qualitative evaluations on 12 test subjects, and developed an XR telepresence prototype for real-time demonstration. Finally, we conducted a user evaluation in which participants assessed perceptual aspects such as motion fidelity, gesture clarity, and perceived social presence, to investigate how our framework facilitates social interaction and actual collaboration. Source code and dataset are available at https://github.com/jhkang0408/PTUGRTEVADE.

II. Related Work

A. XR Telepresence

Immersive telepresence requires a realistic representation of remote spaces, driving the development of diverse reconstruction approaches from depth-based systems with commodity cameras [5] and projection-based augmented reality (AR) [6], [7] to real-time 3D capture like Holoportation [8] and recent photorealistic methods using Gaussian Splatting [9]. However, even with high-fidelity reconstruction, spatial dissimilarity between the environments involved in telepresence poses another challenge. Researchers have explored shared area optimization [2], [10], adaptive spatial mapping [11]–[14], and avatar-based solutions including semantic positioning [1], visual guidance [15], locomotion control [16], [17], and reinforcement learning-based destination selection [18].

Beyond positioning and locomotion, users' references to objects must also be conveyed across dissimilar spaces. Remote collaboration systems [19], [20] often utilize explicit visual cues such as pointers, gaze cursors, and virtual hands, which resolve the referent directly but do not translate the user's body motion. Avatar-based approaches instead adjust the avatar's gestures themselves. Several works have employed IK to adapt avatar poses to dissimilar settings, such as scaling avatars for mixed-scale collaboration [21], retargeting gaze and deictic gestures based on heuristic interaction target detection [22], optimizing workspace integration for side-by-side remote collaboration [23], and preserving body pose during avatar adaptation [24].

Building on these foundations, recent data-driven methods [4], [25] have focused on improving the quality of translated avatar motion by learning from motion capture data. Kang et

al. [25] first introduced a deep learning-based deictic gesture retargeting framework that directly transfers user motions onto avatar motions using paired motion datasets, where user and avatar motions toward different target positions are synchronized in timing. While effective for deictic gestures with predictable endpoints, collecting paired data becomes impractical for diverse gestures with varying motion patterns. Kang et al. [4] addressed this by separating the mapping of angle relationships between targets and tracking signals into gesture categories from the avatar motion generation network, enabling unpaired learning that extends to diverse upper-body gestures. Their bi-directional user study confirmed that spatial adjustment and support for diverse gestures improve social presence and message understanding. However, their target-conditioned encoding assumes accurate target information is always available, leaving the system vulnerable to estimation errors in practice. Moreover, compressing diverse motions into sparse gesture categories can cause mean pose convergence and unnatural hand positioning, and independent body-part control is very limited to simple gaze shifts accompanying dominant-hand deictic gestures. We address these limitations by introducing real-time target estimation, unsupervised target-agnostic motion feature learning, and part-wise gesture translation.

B. Motion Retargeting

Our gesture translation is closely connected to the motion retargeting, which adapts an existing motion so that it fits a different character or environment. Early methods formulated this problem as spacetime optimization [26]–[29]. With the advent of deep learning, neural network-based approaches emerged to handle retargeting between characters with different skeletal structures by learning motion embeddings in an unsupervised manner [30]–[32]. Another line of research focuses on preserving spatial relationships during retargeting, introducing geometric descriptors to maintain the spatial context for motions involving close character-environment interactions [33]–[39]. More recently, neural descriptor fields have been proposed to encode object-agent interactions, enabling robust interaction transfer across objects with varying geometries [40]–[42]. In this paper, we explore the motion retargeting problem in different environments. While prior research has primarily focused on generating contact-based interactions at the sequence level, we address real-time frame-level translation of upper-body gestures.

C. Target Estimation

Target estimation in human-computer interaction originally focused on predicting targets on 2D displays, including desktops [43], [44] and touchscreens [45]. Recently, research has shifted toward identifying which 3D objects users focus on in XR environments. Lu et al. [46] adapted the bubble cursor concept [43] for ray-casting in virtual reality (VR), while Dalsgaard et al. [47] investigated which body motion features best characterize natural pointing. In AR, Wei et al. [48] modeled eye and head endpoint distributions for gaze-based selection. More recent studies have employed recurrent neural networks with multimodal inputs, including motion and eye-tracking data [49], [50].

Our approach is inspired by the neural density estimation framework [51]. They trained a real-time target inference model in which the probability distribution over targets becomes clearer as the user's movement progresses, allowing the system to dynamically refine its understanding of user intent and enabling flexible prediction under high uncertainty. While their work addresses single-hand pointing behavior using synthetic data from biomechanical simulation, we focus on a different problem: estimating separate targets for the head, right hand, and left hand, enabling user-invariant prediction for general upper-body gestures using only limited data from a single user.

III. Dataset

We first detail the dataset used to train and evaluate our framework. We extend the dataset of [4], which includes static deictic, beat, and descriptive gestures. In that dataset, however, deictic gestures were limited to the static end poses of pointing or touching, and independent body-part control was restricted to simple gaze shifts that accompanied dominant-hand deictic gestures. We address these limitations as follows. First, to enrich gesture diversity, we additionally capture dynamic deictic gestures (e.g., pointing with a circling motion or repeatedly touching a target) and referential gestures, in which the hand is loosely oriented toward a target to reference it implicitly without taking the explicit form of a deictic gesture (e.g., gesturing toward a target at a lowered hand height). We further capture their combinations and transitions with beat gestures (rhythmic movements driven by prosody) and descriptive gestures (mimetic, iconic, and metaphoric). Second, to enable independent body-part control, we record these gestures in multi-target scenarios in which the head and both hands independently reference different targets.

Using an HTC Vive Pro Eye headset and Noitom Hi5 VR gloves, we record the transformations of the head, both hands, and both index fingertips, along with the corresponding target positions, and downsample all motion data to 30 fps. After calibrating eye, finger, and joint tracking, each subject remained stationary and performed gestures toward targets displayed as simple spheres to remain visually neutral, in a virtual environment built with the Unity3D game engine. Subjects were given simple scenarios (e.g., office meetings, purchasing items, etc.) and instructed to treat the targets as people or objects while acting accordingly.

The newly captured data amounts to 152 minutes for training and 41 minutes for testing. Following [4], we arranged targets in a grid layout (azimuth × height × distance) spanning a 180° frontal range. Detailed configurations are provided in the supplementary material. Combined with [4], the training data comprises approximately 800 minutes (13.3 hours), while the test data amounts to 99 minutes (1.7 hours). Training data was collected from a single male subject (height: 170cm), while test data was collected from 12 subjects (5 from [4] and 7 newly captured; 5 females and 7 males; height range: 158–187cm, mean: 172.1cm, std: 8.4cm), including two left-

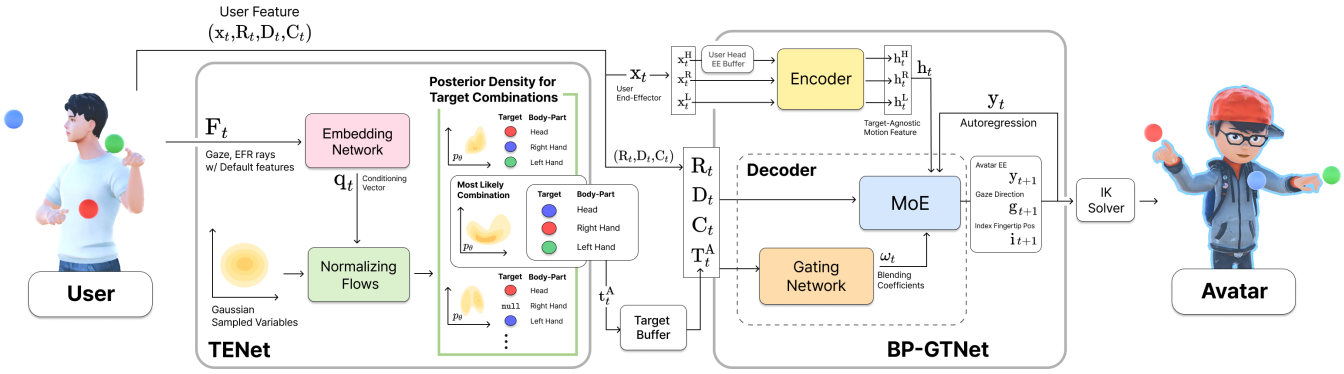


Fig. 2. Overview of the proposed framework. The system consists of two networks operating in sequence. First, TENet estimates per-body-part target positions by computing posterior densities over all candidate target combinations from user behavioral features. Next, BP-GTNet generates avatar motion through a dual end-effector representation: the encoder extracts target-agnostic motion features $\mathbf{h}$ from user end-effectors $\mathbf{x}$, while the decoder transforms $\mathbf{h}$ into avatar end-effectors $\mathbf{y}$ conditioned on the estimated targets. The generated $\mathbf{y}$ drives an IK solver to produce the final avatar pose, with autoregressive feedback ensuring temporal coherence.

handed and three left-eye-dominant individuals, with varying motion speeds and styles.

A single human annotator labels the current target of the head and each hand at every frame (we also provide an inter-rater reliability check with two additional annotators on a data subset, Fleiss' $\kappa = 0.95$, in the supplementary material) according to the following criteria. For the head, a target is labeled when the gaze moves toward or fixates on a specific target; when the gaze rests on an ambiguous point that does not correspond to any target, or the head remains in a forward-facing or idle posture, the head is labeled as null. For the hands, a hand kept lowered at rest is labeled as null, whereas a raised hand is labeled either with the head's target or with a different target, depending on the gesture type. Because the intended target is often unclear at the moment a hand begins to rise, the annotator determines the target retrospectively from the gesture's subsequent motion and applies that label from the onset of the rising motion. Specifically, for deictic and referential gestures, the hand is labeled with the target it is directed toward. For beat gestures, the hand is always labeled to align with the head's target. For descriptive gestures, a hand depicting something in a specific direction is labeled with the target in that direction, whereas a description without a specific direction is aligned with the head's target.

IV. Method

Figure 2 provides an overview of a framework that consists of two distinct models: TENet and BP-GTNet. Given the tracking signal history of the user's head and both hands, behavioral features are first extracted. TENet then takes these features as observations and computes the posterior distribution over all possible target combinations for each body part. The combination with the highest probability is selected as the current target. Conditioned on this target along with the user's tracking signals, BP-GTNet autoregressively regresses the transformations of the head and both hand end-effectors. Finally, an IK solver [52] computes the upper-body pose of the avatar for the current frame.

During training, TENet and BP-GTNet are each trained independently on the same captured motion sequences tailored

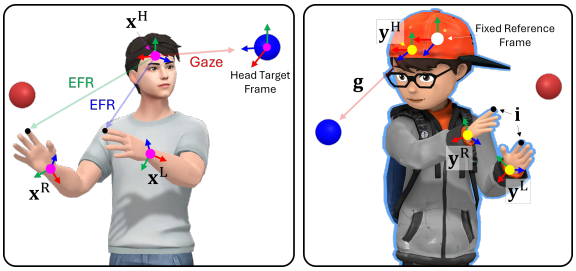


Fig. 3. Visualization of the input and output representations for TENet and BP-GTNet. (Left: User) The gaze vector and eye-finger rays (EFRs) for both hands, used as directional inputs to TENet, are shown as arrows. The encoder of BP-GTNet takes user-side end-effectors as input: $\mathbf{x}^H$ relative to the head target frame, and $\mathbf{x}^R$, $\mathbf{x}^L$ relative to head-relative frame. (Right: Avatar) The decoder of BP-GTNet computes the avatar-side end-effectors $\mathbf{y}^H$, $\mathbf{y}^R$, $\mathbf{y}^L$ in the fixed reference frame, along with the gaze direction $\mathbf{g}$ and index fingertip positions $\mathbf{i}$.

to their respective objectives, whereas at inference time, the two networks are linked together to operate as a single end-to-end pipeline.

Coordinate Frames. The method relies on three custom coordinate frames, defined as follows (Table I summarizes the features expressed in each frame and the networks that use them):

- Reference frame: its forward direction is the head's forward direction in the upright posture, projected onto the ground plane; its position follows the head.
- Fixed reference frame: its forward direction is the same as that of the reference frame; its position is fixed at the head's position in the upright posture.
- Head target frame: its position is at the head's target; its forward direction points toward the origin of the fixed reference frame.

In addition, we use standard joint-relative frames, each defined relative to the transformation of a specific joint (e.g., the head-relative and hand-relative frames).

A. Target Estimation Network (TENet)

Input and Output: TENet is designed to estimate the target positions of the head and both hands for a wide range of

TABLE I
SUMMARY OF NOTATION: EACH SYMBOL, THE COORDINATE FRAME IN WHICH EACH IS EXPRESSED, AND THE NETWORK THAT USES IT (T: TENET, B: BP-GTNET).

Symbol	Frame	Net.	Role
Gaze, EFR rays	Reference	T	Directional intent
$\mathbf{t}^{\mathrm{U}}$	Reference	T	User targets
$\mathbf{x}^{\mathrm{H}}$	Head target	B	Target-agnostic head input
$\mathbf{x}^{\mathrm{R}}, \mathbf{x}^{\mathrm{L}}$	Head-relative	B	Target-agnostic hand input
$\mathbf{t}^{\mathrm{A}}$	Fixed reference	B	Avatar targets
$\mathbf{y}$	Fixed reference	B	Target-conditioned output
$\mathbf{g}$	Head-relative	B	Generated gaze direction
$\mathbf{i}$	Hand-relative	B	Generated fingertip position
$\mathbf{d}$	—	T,B	Active vs. resting state
$\mathbf{r}$	—	B	Directed at a target
$\mathbf{c}$	—	B	Target-estimate consistency

users while relying on only a small amount of training motion data. This generalization is enabled by a direction-based and distance-based input representation that is invariant across users.

The input uses three direction vectors: the gaze vector and the eye-finger rays (EFRs) for both hands (Figure 3, left). An EFR is a ray from the midpoint between the eyes through the index fingertip, which has been shown to yield the highest accuracy for deictic gestures [53]. These vectors are expressed in the reference frame, which captures directional intent independent of individual physical characteristics. A default feature $\mathbf{d}$ is also defined for each body part to indicate whether it is in an active targeting state or a resting state (details in supplementary material).

TENet takes a sequence of three direction vectors ($\in \mathbb{R}^9$) and default features ($\mathbf{d} \in \mathbb{R}^3$) as input. The sequence spans 45 frames, including the current frame t and preceding frames (1.5 seconds at 30 fps), resulting in an input tensor $\mathbf{F}_t \in \mathbb{R}^{45\times 12}$. This window size was empirically determined to sufficiently capture the temporal dynamics of upper-body gestures.

TENet outputs an approximated posterior density given the input $\mathbf{F}$ and user target position $\mathbf{t}^{\mathrm{U}} = \{\mathbf{t}^{\mathrm{H}}, \mathbf{t}^{\mathrm{R}}, \mathbf{t}^{\mathrm{L}}\} \in \mathbb{R}^9$, which represents the target positions of the head, right hand, and left hand in spherical coordinates relative to the reference frame. Within TENet all targets are user-side, so we drop the superscript U and write $\mathbf{t}$ in the equations below. The distance is clamped to 1m if it exceeds this threshold. This design choice is motivated by the distance-invariant nature of pointing gestures [54], allowing the model to generalize across varying target distances.

Notably, TENet jointly predicts targets for all three body parts. This is because the system must handle not only deictic gestures but also diverse gesture types where independent per-part prediction is fundamentally insufficient for gesture types involving bimanual coordination or gaze-hand coupling. For example, when a user gazes at an object while describing its shape with both hands (a mimetic gesture), the hands are not directed at the object but performing descriptive motions. TENet nonetheless identifies all three body parts as referencing that object. We validate that joint prediction outperforms independent per-body-part prediction in Section V.

Network Architecture: Rather than using a regression model that directly predicts target positions, TENet employs a probabilistic approach to estimate the posterior distribution over targets. This design enables the system to represent and refine uncertainty as the user's movement progresses: early in a gesture, the distribution may be broad across multiple plausible targets, but it sharpens as more discriminative motion is observed. Furthermore, the probabilistic formulation enables direct comparison across candidate targets rather than relying on a single point estimate.

To realize this probabilistic framework, we adopt conditional normalizing flows following [51]. While alternative generative models such as variational autoencoders (VAEs) and generative adversarial networks (GANs) provide only approximate posteriors and diffusion models require iterative sampling steps, normalizing flows offer two clear advantages for target estimation: (1) through invertible mappings, they enable exact likelihood computation, and (2) they support efficient single-pass inference, making them suitable for real-time applications.

Given the behavioral feature sequence $\mathbf{F}$, the model learns an invertible mapping f_θ that transforms a simple latent distribution into the complex target distribution. Specifically, f_θ maps from a latent space $\mathbf{z} \in \mathbb{R}^9$ with a standard Gaussian prior $p(\mathbf{z})$ to the target space, while its inverse f_θ^{-1} maps targets back to the latent space. Thus, the posterior density over target $\mathbf{t}$ is computed via the change of variables formula:

$$p_\theta(\mathbf{t}|\mathbf{F}) = p(\mathbf{z} = f_\theta^{-1}(\mathbf{t};\mathbf{F})) \left| \det \left(\frac{\partial f_\theta^{-1}(\mathbf{t};\mathbf{F})}{\partial \mathbf{t}} \right) \right| \quad (1)$$

where the model is trained by maximizing the log-likelihood of ground-truth targets, $\mathbb{E}[\log p_\theta(\mathbf{t}|\mathbf{F})]$.

Our implementation employs a Transformer-based embedding network that extracts temporal features from the input sequence $\mathbf{F}$ while capturing inter-part dependencies through attention mechanisms. For the flow model, we adopt Glow [55]. Full architecture details are provided in the supplementary material.

At inference time, we leverage the joint formulation by enumerating all possible target combinations across the three body parts. Let $\{\mathbf{t}_i^{\mathrm{H}}\}$, $\{\mathbf{t}_j^{\mathrm{R}}\}$, $\{\mathbf{t}_k^{\mathrm{L}}\}$ denote the candidate user target sets for the head, right hand, and left hand, respectively, where i, j, k are indices ranging from 1 to $N+1$, covering the N targets and one null target representing the resting state. This yields $(N+1)^3$ possible combinations. We evaluate the posterior probability for each combination and select the most likely combination:

$$i_t^*, j_t^*, k_t^* = \arg\max_{i,j,k} p_\theta(\mathbf{t}_i^{\mathrm{H}}, \mathbf{t}_j^{\mathrm{R}}, \mathbf{t}_k^{\mathrm{L}}|\mathbf{F}_t). \quad (2)$$

B. Body-Part Gesture Translation Network (BP-GTNet)

Input and Output: The BP-GTNet takes the user's tracking signals as input and generates corresponding upper-body gestures for the avatar. To bridge diverse users and avatars, it operates on end-effector transformations rather than joint

angles. This choice reduces sensitivity to individual limb proportions on the user side, while providing an avatar-invariant output that drives an IK solver to produce poses for different skeletal configurations. The network is trained and evaluated using a commonly adopted 6D rotation representation [56].

Dual End-Effector Representation. BP-GTNet employs two distinct end-effector representations, $\mathbf{x}$ and $\mathbf{y}$, to enable target-agnostic motion encoding and target-conditioned decoding (Figure 3).

The first representation, $\mathbf{x} = \{\mathbf{x}^{\mathrm{H}}, \mathbf{x}^{\mathrm{R}}, \mathbf{x}^{\mathrm{L}}\} \in \mathbb{R}^{9\times 3}$, is defined as follows. $\mathbf{x}^{\mathrm{H}}$ represents the head transformation in the head target frame. The hand representations $\mathbf{x}^{\mathrm{R}}$ and $\mathbf{x}^{\mathrm{L}}$ are defined in the head-relative frame. The second representation, $\mathbf{y} = \{\mathbf{y}^{\mathrm{H}}, \mathbf{y}^{\mathrm{R}}, \mathbf{y}^{\mathrm{L}}\} \in \mathbb{R}^{9\times 3}$, expresses all three end-effectors (head, right hand, and left hand) directly in the fixed reference frame.

During training, both $\mathbf{x}$ and $\mathbf{y}$ are extracted from the same motion sequence. At inference time, however, $\mathbf{x}$ is obtained from the user's tracking signals, while $\mathbf{y}$ is generated for the avatar. Since constructing paired data between users and avatars is infeasible, our network learns target-agnostic motion features in an unsupervised manner: the encoder receives no explicit supervision on what constitutes a target-invariant representation, but must discover this property implicitly through the reconstruction objective on $\mathbf{y}$ alone. This dual representation design enables the transfer of motion from user to avatar across different target positions while preserving the original motion characteristics.

Additionally, a ray feature $\mathbf{r} \in \mathbb{R}^3$ indicates whether each body part is directed toward any candidate target, set to 1 when a candidate falls within 10° of the corresponding directional ray (gaze or EFR) and averaged over the past 15 frames. A target estimation consistency feature $\mathbf{c} \in \mathbb{R}^3$ measures how reliably the current target has been maintained: it is reset to 0 whenever the target changes, and otherwise updated by $\pm 1/15$ per frame within $[0,1]$ according to gaze fixation. This lets BP-GTNet distinguish confident targeting from transitional phases and produce smooth motion even when TENet's estimate momentarily flickers.

BP-GTNet takes as input 30 frames of the avatar target position sequence $\mathbf{T}_t^{\mathrm{A}} \in \mathbb{R}^{30\times 9}$, expressed in the fixed reference frame, along with ray feature $\mathbf{R}_t \in \mathbb{R}^{30\times 3}$, default feature $\mathbf{D}_t \in \mathbb{R}^{30\times 3}$, target estimation consistency feature $\mathbf{C}_t \in \mathbb{R}^{30\times 3}$, and the current frame's end-effector representations $\mathbf{x}_t$ and $\mathbf{y}_t$. We additionally provide a 4-frame history of $\mathbf{x}^{\mathrm{H}}$ to capture recent head dynamics. It then outputs the next frame's end-effector representation $\mathbf{y}_{t+1}$, as well as the gaze direction $\mathbf{g}_{t+1} \in \mathbb{R}^3$ and the index fingertip positions $\mathbf{i}_{t+1} \in \mathbb{R}^6$.

Network Architecture: BP-GTNet consists of a Transformer-based encoder and a Mixture-of-Experts (MoE) [57] decoder. The encoder processes $\mathbf{x}$ using separate Transformer encoders for the head and hands, producing compact motion features $\mathbf{h}^{\mathrm{H}}, \mathbf{h}^{\mathrm{R}}, \mathbf{h}^{\mathrm{L}}$ that capture target-agnostic motion characteristics. The decoder takes these features together with the avatar's target positions to generate $\mathbf{y}$.

The decoder follows two different pathways at training and inference time (Figure 4). During training, the motion features are discretized and fed to part-wise auxiliary decoders that supervise each body part independently. These two components, motion discretization and part-wise auxiliary decoders, are active only during training; we describe each below and explain how it shapes inference-time behavior. Because the decoder is autoregressive, we additionally employ scheduled sampling [58] during training to reduce the train–inference discrepancy.

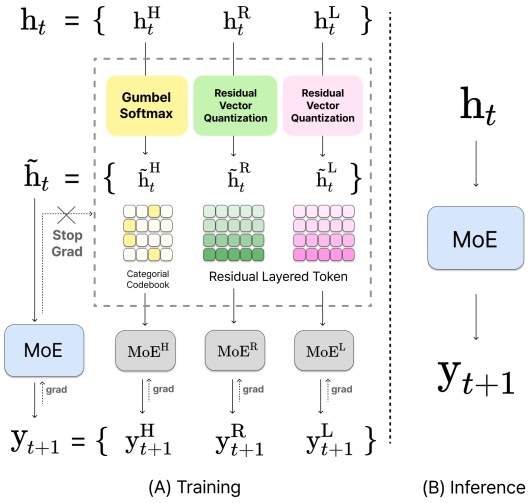


Fig. 4. Comparison of BP-GTNet's decoder pathway during training and inference. (A) During training, motion features are discretized via hybrid discretization and supervised by part-wise auxiliary decoders ($\mathrm{MoE}^{(p)}, p \in \{\mathrm{H,R,L}\}$). (B) At inference, the auxiliary decoders are removed and continuous features bypass discretization, yielding fine-grained motion generation.

Motion Discretization. During training, we apply discretization to the motion features:

$$\tilde{\mathbf{h}}_t^{\mathrm{H}} = \mathrm{GS}(\mathbf{h}_t^{\mathrm{H}}), \quad \tilde{\mathbf{h}}_t^{\mathrm{R}} = \mathrm{RVQ}(\mathbf{h}_t^{\mathrm{R}}), \quad \tilde{\mathbf{h}}_t^{\mathrm{L}} = \mathrm{RVQ}(\mathbf{h}_t^{\mathrm{L}}). \tag{3}$$

where $\mathrm{GS}(\cdot)$ denotes Gumbel-Softmax sampling [59] (6 channels of codebook size 6, temperature 5) and $\mathrm{RVQ}(\cdot)$ denotes Residual Vector Quantization [60] (codebook size 128, 5 residual layers). We adopt different discretization strategies for each body part based on its motion characteristics. For the head, $\mathbf{x}^{\mathrm{H}}$ exhibits discontinuities when targets change since it is defined relative to the target position. Gumbel-Softmax forces the network to encode head motion into discrete categories rather than relying on continuous interpolation across discontinuous boundaries. For the hands, $\mathbf{x}^{\mathrm{R}}$ and $\mathbf{x}^{\mathrm{L}}$ are continuous signals, and RVQ's coarse-to-fine strategy preserves fine-grained motion details through progressive refinement.

Discretization serves as a training-time information bottleneck. Restricting the decoder's input to a finite code set forces inputs near the same code to map to nearly identical outputs. At inference the decoder receives continuous features, but this learned insensitivity absorbs the minor fluctuations caused by unstable target estimation and thereby suppresses jitter, while variations large enough to cross code boundaries still pass through and preserve motion detail.

Part-wise Auxiliary Decoders. To ensure independent controllability, each auxiliary decoder reconstructs one body part from that part's features alone, so each representation must encode the part's motion without depending on the other parts. The main decoder receives these features with stop-gradient; otherwise its gradients would entangle cross-part information

TABLE II
INFERENCE TIME OF EACH COMPONENT BY NUMBER OF TARGETS (1–7).

Component	Inference Time [ms]						
	1	2	3	4	5	6	7
TENet	$6.2^{\pm 1.1}$	$7.1^{\pm 0.4}$	$9.5^{\pm 0.4}$	$13.9^{\pm 0.5}$	$19.9^{\pm 0.5}$	$28.8^{\pm 0.8}$	$41.3^{\pm 0.8}$
BP-GTNet	$4.6^{\pm 1.0}$	$4.3^{\pm 0.3}$	$4.3^{\pm 0.2}$	$4.2^{\pm 0.2}$	$4.3^{\pm 0.2}$	$4.3^{\pm 0.3}$	$4.4^{\pm 0.3}$
Total	$13.9^{\pm 2.5}$	$14.7^{\pm 1.4}$	$17.2^{\pm 1.3}$	$22.0^{\pm 1.5}$	$28.2^{\pm 1.6}$	$37.8^{\pm 1.8}$	$51.1^{\pm 1.8}$

into the representations. This separates the two roles: the per-part features carry disentangled motion, while the main decoder, taking all three features jointly, learns the inter-part relationships needed to combine them into coordinated motion. As a result, once the auxiliary decoders are removed at inference, each body part can be translated to a different target without affecting the others, yet the generated motion remains natural.

C. Inference

At inference, TENet first extracts the behavioral feature $\mathbf{F}$ from the user's tracking signal and estimates the corresponding target positions $\mathbf{t}^{\mathrm{U}}$. These user targets are then mapped to the avatar target positions $\mathbf{t}^{\mathrm{A}}$ through one-to-one correspondence. Subsequently, BP-GTNet generates the avatar motion by taking $(\mathbf{x}, \mathbf{R}, \mathbf{D}, \mathbf{C})$ derived from the user's tracking signal, along with $\mathbf{T}^{\mathrm{A}}$ and $\mathbf{y}$ obtained from the avatar's state.

Table II reports the inference time of each component (mean $\pm$ standard deviation over 45 s of live streaming), measured on an Intel Core Ultra 9 285K CPU with an NVIDIA RTX 5080 GPU. TENet and BP-GTNet have lightweight model sizes of 26.5 MB and 28.4 MB, respectively. The system runs on a single local desktop: Unity (6000.0.36f1) handles input feature extraction, IK, and post-processing, while both networks run in a Python module; the two communicate via ZeroMQ in a separate thread from the rendering loop. The inference time of TENet ($\approx 5.20 + 0.07\,(N+1)^3$ ms) scales cubically with the number of targets, ranging from 6.2 ms for 1 target to 41.3 ms for 7 targets, whereas BP-GTNet remains constant at about 4.3 ms. The total time, including the Unity-side overhead and communication, stays within the 33 ms frame budget of our 30 fps motion data for up to 5 targets (28.2 ms) and exceeds it from 6 targets onward, so the practical operating range of the current implementation is $N \leq 5$ on this hardware.

V. TECHNICAL EVALUATION

We evaluate our framework both quantitatively and qualitatively using our in-house dataset. The model is trained on motion sequences captured from a single person, while the test set includes 12 subjects with varying physical characteristics to assess generalization capability.

A. Evaluation on TENet

Since BP-GTNet's performance depends on the accuracy of target estimation, we first evaluate TENet. We report the average accuracy of target estimation across the head, right hand, and left hand, measured at every frame. To evaluate performance under varying numbers of candidate targets, we augment each motion sequence with dummy targets placed randomly within a 180° horizontal and 120° vertical field of view at distances of at least 0.4m. Since the original sequences contain between 1 and 3 targets, we add dummy targets to reach the target count for each condition (3, 5, or 7). As gesture targets in general XR scenarios follow no known spatial distribution within the user's field of view, we consider uniform random placement the fairest choice, as it avoids biasing the evaluation toward any particular layout. Due to the randomness in dummy target placement, we repeat each experiment 10 times and report the average accuracy. The standard deviation is below 0.1% across all experiments and thus omitted.

TABLE III
COMPARISON OF TENET WITH THE ALTERNATIVE METHODS ON TEST SUBJECTS WITH VARYING NUMBERS OF TARGETS (3,5,7).

Method	Accuracy↑ [%]		
	3	5	7
TENet (Ours)	**91.3**	**90.2**	**89.3**
TENet-Indep	71.8	70.7	69.7
Baseline	85.1	84.2	83.7
Diffusion	89.2	87.5	86.5

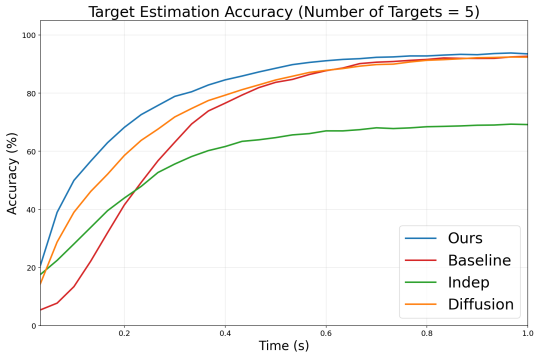


Fig. 5. Comparison of target estimation accuracy over time after target transition across network architectures (number of targets = 5). Accuracy is measured at each frame from the moment a ground-truth target changes (t=0) until 1 second later, averaged across all target transitions in the test set.

We compare TENet with the following baselines: Baseline [51], which uses Real NVP [61] with an MLP embedding network; Diffusion, a residual MLP-based diffusion regression model; and TENet-Indep, a variant of TENet that shares the same embedding network but uses three separate flow networks for each body part, unlike TENet which predicts all three simultaneously with a single flow network. For the Diffusion model, we map the regression output to the candidate target with the smallest Euclidean distance. All methods are trained with comparable model sizes, equal training time (5 hours), and identical training protocols (batch size, learning rate, etc.). Implementation details are provided in the supplementary material.

Table III shows that our method achieves the highest accuracy across all conditions. Compared to the Baseline, our Transformer-based encoding captures inter-part dependencies through attention mechanisms, providing more effective conditioning than an MLP. Additionally, Glow models more complex distributions than Real NVP. Diffusion outperforms Baseline by effectively modeling complex movements through

TABLE IV
COMPARISON OF BP-GTNET WITH THE ALTERNATIVE METHODS ON TEST SUBJECTS WITH VARYING NUMBERS OF TARGETS (3,5,7). ALL METRICS ARE COMPUTED ON THE RAW NETWORK OUTPUTS.

Method	DIP-PE↓ [°]			DIP-TE↓ [cm]			TMP-PE↓ [cm]			TMP-RE↓ [°]			FGD↓		
	3	5	7	3	5	7	3	5	7	3	5	7	3	5	7
UGNet (Baseline)	$3.6^{\pm0.3}$	$3.6^{\pm0.2}$	$3.6^{\pm0.2}$	$\mathbf{9.5}^{\pm0.7}$	$\mathbf{10.5}^{\pm0.8}$	$\mathbf{10.8}^{\pm1.0}$	$20.8^{\pm0.4}$	$21.0^{\pm0.3}$	$21.2^{\pm0.4}$	$41.6^{\pm0.5}$	$42.0^{\pm0.4}$	$42.1^{\pm0.5}$	$3.39^{\pm0.52}$	$3.21^{\pm0.31}$	$3.22^{\pm0.33}$
BP-GTNet (Ours)	$\underline{2.8}^{\pm0.2}$	$\underline{2.9}^{\pm0.2}$	$\underline{2.9}^{\pm0.1}$	$12.4^{\pm0.9}$	$13.6^{\pm0.7}$	$13.6^{\pm1.0}$	$\underline{18.5}^{\pm0.4}$	$\underline{18.8}^{\pm0.3}$	$\underline{19.1}^{\pm0.4}$	$\underline{24.8}^{\pm0.6}$	$\underline{25.5}^{\pm0.6}$	$\underline{25.9}^{\pm0.4}$	$\mathbf{1.01}^{\pm0.05}$	$\mathbf{0.98}^{\pm0.04}$	$\mathbf{0.97}^{\pm0.04}$
w/o MD	$\mathbf{2.7}^{\pm0.2}$	$\mathbf{2.8}^{\pm0.2}$	$\mathbf{2.8}^{\pm0.2}$	$13.7^{\pm0.9}$	$14.7^{\pm0.8}$	$14.8^{\pm0.8}$	$\mathbf{16.4}^{\pm0.4}$	$\mathbf{16.7}^{\pm0.4}$	$\mathbf{17.0}^{\pm0.4}$	$\mathbf{19.8}^{\pm0.5}$	$\mathbf{20.5}^{\pm0.5}$	$\mathbf{21.3}^{\pm0.5}$	$3.34^{\pm0.34}$	$3.29^{\pm0.18}$	$3.20^{\pm0.24}$
w/o PAD	$4.0^{\pm0.3}$	$4.0^{\pm0.3}$	$3.9^{\pm0.3}$	$\underline{12.3}^{\pm0.0}$	$\underline{13.4}^{\pm0.6}$	$\underline{13.4}^{\pm1.1}$	$19.3^{\pm0.0}$	$19.5^{\pm0.3}$	$19.8^{\pm0.4}$	$30.0^{\pm0.6}$	$30.5^{\pm0.6}$	$31.0^{\pm0.6}$	$\underline{1.15}^{\pm0.04}$	$\underline{1.13}^{\pm0.04}$	$\underline{1.13}^{\pm0.05}$
w/o EE-Dual	$14.5^{\pm0.7}$	$14.4^{\pm0.9}$	$13.6^{\pm1.4}$	$21.7^{\pm0.9}$	$22.9^{\pm1.0}$	$22.7^{\pm1.5}$	$19.4^{\pm0.3}$	$19.6^{\pm0.4}$	$19.9^{\pm0.4}$	$50.6^{\pm0.8}$	$51.1^{\pm0.5}$	$51.1^{\pm0.8}$	$1.37^{\pm0.06}$	$1.38^{\pm0.03}$	$1.40^{\pm0.04}$

iterative denoising, but it outputs a single regressed position that must be mapped to the nearest candidate target, limiting its ability to handle ambiguous situations where multiple targets are similarly plausible. Additionally, its iterative sampling process incurs computational overhead ($\leq$20 fps), which hinders real-time applications. TENet-Indep shows a significant performance drop from TENet because independent prediction ignores correlations among body parts; during explanatory gestures where the user gazes at a target while gesturing toward it, the head and hand targets are naturally aligned, and joint prediction enables the model to leverage this alignment to better distinguish between explanatory and deictic situations.

Figure 5 compares target estimation accuracy over time after a target transition. All methods start below 25% when no motion evidence is available. Our method rises most rapidly, reaching approximately 70% by 0.2s, while Baseline and Diffusion remain below 60% at this point. TENet-Indep saturates early and plateaus around 70%, while Ours, Baseline, and Diffusion continue to improve beyond 90%. This demonstrates that joint modeling is essential not only for achieving high final accuracy but also for enabling sustained improvement over time. Although Ours, Baseline, and Diffusion converge to similar accuracy ($\sim$93%) after 1 second, our early-phase advantage is critical for real-time telepresence, where immediate avatar response ensures natural communication.

B. Evaluation on BP-GTNet

We evaluate BP-GTNet using three metrics: deictic intention preservation (DIP), target-centric motion preservation (TMP), and Fréchet gesture distance (FGD). Detailed definitions are as follows:

- **DIP** measures how accurately the avatar conveys the user's deictic intentions for pointing and touching gestures. We evaluate sequences where the user starts from an upright posture and performs a single-target gesture with their dominant hand while the other hand remains stationary. For each sequence, we measure accuracy at the moment when the avatar's end-effector is closest to the target, capturing the gesture's apex where referencing intent is most clearly expressed. We report two sub-metrics: pointing error (DIP-PE [°]) measures the angular deviation between the avatar's EFR and the vector from the head to the target, providing a distance-invariant metric [54], and touching error (DIP-TE [cm]) measures the distance between the avatar's index fingertip and the target position.
- **TMP** measures how well the avatar preserves the user's motion characteristics when targeting different positions. For each body part, we define a target-centric coordinate frame located at the target position with its forward direction pointing toward the user's reference frame. We transform both the user's and avatar's end-effectors into their respective ground-truth target coordinate frames, then compute the difference between these transformed representations. Using ground-truth targets ensures fair evaluation: if TENet estimates an incorrect target, the method generates motion toward that wrong target, resulting in larger errors when measured against the ground-truth target frame. We report position error (TMP-PE [cm]) and rotation error (TMP-RE [°]) averaged over frames where the body part has a non-null target.
- **FGD** evaluates the naturalness of generated motions by comparing the distribution of real and generated gestures in a learned latent space; lower values indicate better alignment with real human gesture statistics. We train a two-layer convolutional autoencoder on the training set to reconstruct 30-frame sequences of head and both hands transformations, achieving reconstruction errors of 1.1 cm and 1.2° on the test set. Using the trained encoder, we extract latent vectors from both real user motions and generated avatar motions with a 30-frame window and 15-frame stride, then compute the Fréchet distance between the two distributions.

We compare BP-GTNet with UGNet (Upper-Body Gesture Network) [4] as our baseline and conduct ablations on motion discretization (MD), part-wise auxiliary decoder (PAD), and dual end-effector representation (EE-Dual). For the EE-Dual ablation, we replace $\mathbf{x}$ with $\mathbf{y}$ to use only a single end-effector representation. For UGNet, we train on our extended dataset using the same protocol as [4]. Since UGNet generates only a single dominant hand while translating the other hand relative to the head coordinate frame, we mirror the dataset to train separate models for right-handed and left-handed users. For the ablations, we use the same training protocol as BP-GTNet. Implementation details are provided in the supplementary material.

For evaluation, we use mocap sequences from test subjects as user input and estimate targets using TENet. Following the same protocol as the TENet evaluation, we add dummy targets to reach 3, 5, or 7 total targets. Avatar target positions are randomly placed to create dissimilar environments. We repeat each experiment 10 times and report the mean $\pm$ standard deviation. All methods are evaluated under identical conditions.

Table IV shows that BP-GTNet outperforms UGNet across all metrics except DIP-TE. The improvement is particularly notable in TMP and FGD metrics. This substantial improve-

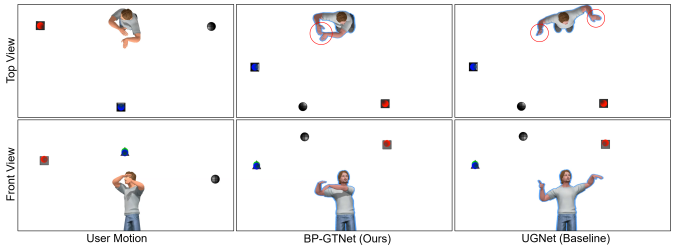


Fig. 6. Comparison between BP-GTNet (ours) and UGNet (baseline). Scenes contain 3 target objects (black sphere, black pyramid, black rectangle) positioned differently for the user and the avatar. Predicted targets for head (red sphere), right hand (green sphere), and left hand (blue sphere) are head→black rectangle, right hand→black pyramid, left hand→black pyramid. UGNet copies the non-dominant hand (left hand in this figure) from the user's head coordinates directly to the avatar's head coordinates, preventing part-wise translation and causing misalignment. Moreover, when the head and dominant hand (right hand) are directed at different targets, UGNet's dominant hand defaults to a deictic gesture, demonstrating its limited independent control.

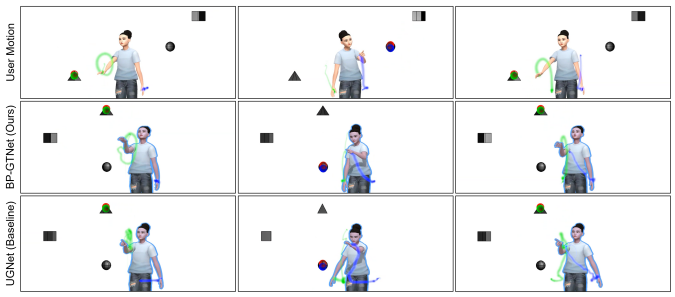


Fig. 7. Comparison between BP-GTNet (ours) and UGNet (baseline), shown in temporal order from left to right. Green and blue trajectories show right and left-hand paths, respectively. BP-GTNet generates more detailed and stable motion compared to UGNet.

ment can be attributed to three factors. First, UGNet generates only the dominant hand through the network while copying the non-dominant hand directly from the user's head coordinate frame, whereas our part-wise approach enables independent retargeting of both hands; moreover, UGNet provides only very limited independent control even for the dominant hand, as its motion remains tightly coupled to the head (Figure 6). Second, UGNet's target-conditioned encoding is designed under the assumption that ground-truth targets are always available, making it vulnerable to fluctuations and rapid changes in estimated targets at inference time, which leads to unstable motion. Third, UGNet compresses user motions into categorical gesture values, which loses fine-grained motion details, resulting in mean pose convergence and unnatural hand poses (Figure 7). In contrast, our unsupervised learning directly extracts motion features from the user's tracking signals, preserving individual movement characteristics. The significantly lower FGD further indicates that BP-GTNet generates more natural and stable motions.

The higher DIP-TE of BP-GTNet stems from the design choice that underlies these improvements: by learning target-agnostic motion features rather than fitting to the end poses of touching gestures, the network trades a residual positional offset for better motion preservation. In the full system, this offset is corrected by IK post-processing: when the user's index fingertip comes within 15 cm of the estimated target,

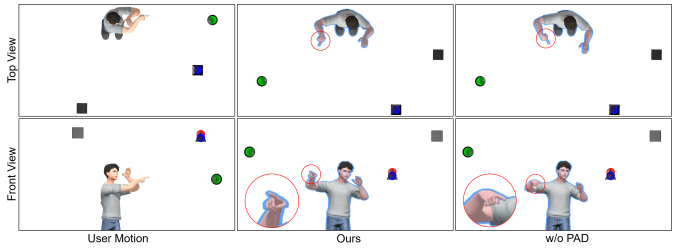


Fig. 8. Ablation result on part-wise auxiliary decoder (PAD). Predicted targets are head→black pyramid, right hand→black sphere, left hand→black pyramid. Without PAD, the model fails to learn part-wise controllability, resulting in the avatar's right hand not properly pointing toward the black sphere.

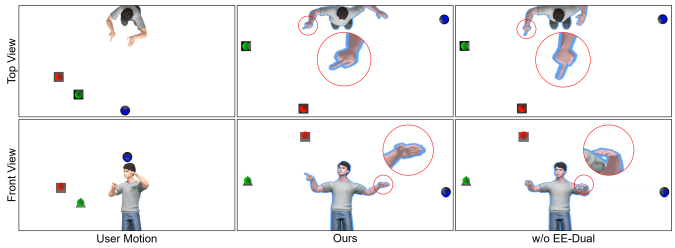


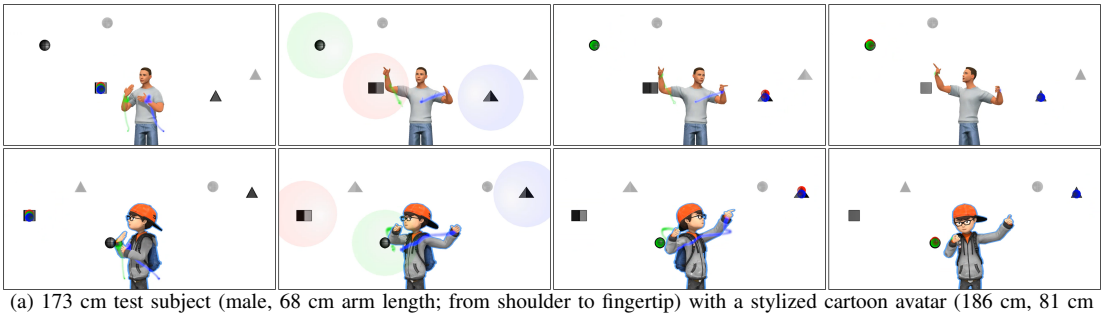
Fig. 9. Ablation result on dual end-effector representation (EE-Dual). Predicted targets are head→black rectangle, right hand→black pyramid, left hand→black sphere. Without EE-Dual, the model fails to learn target-agnostic motion features, resulting in both hands directing toward empty space.

the IK weight of the avatar's fingertip end-effector is smoothly increased to bring the fingertip into contact with the target. To quantify the effect of this correction, we apply the same IK post-processing to both methods and re-measure DIP-TE (Table V). After correction, the two methods achieve comparable error, and the remaining error mainly reflects target estimation errors of TENet. Qualitative examples are shown in Figure 10 (c) and (d). In contrast, UGNet's degradation in overall motion quality cannot be recovered by post-processing.

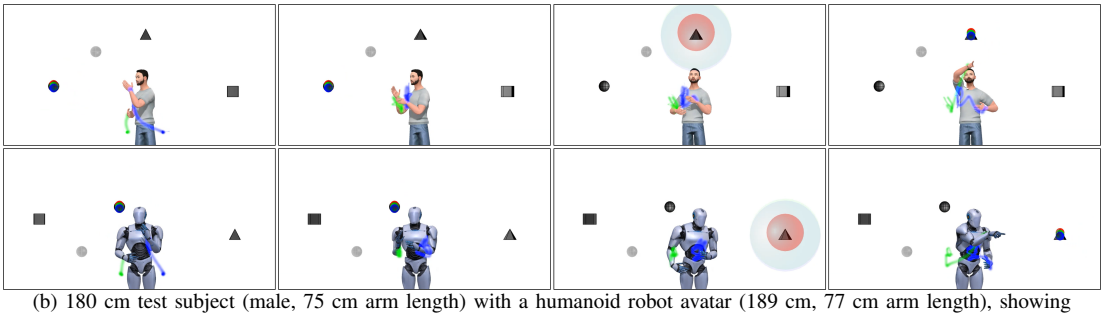
TABLE V
DIP-TE [CM] AFTER IK POST-PROCESSING

Method	3	5	7
UGNet (Baseline)	**2.1**$^{\pm 0.5}$	3.4$^{\pm 0.8}$	3.6$^{\pm 0.8}$
BP-GTNet (Ours)	2.2$^{\pm 0.5}$	**3.1**$^{\pm 0.6}$	**3.2**$^{\pm 0.5}$

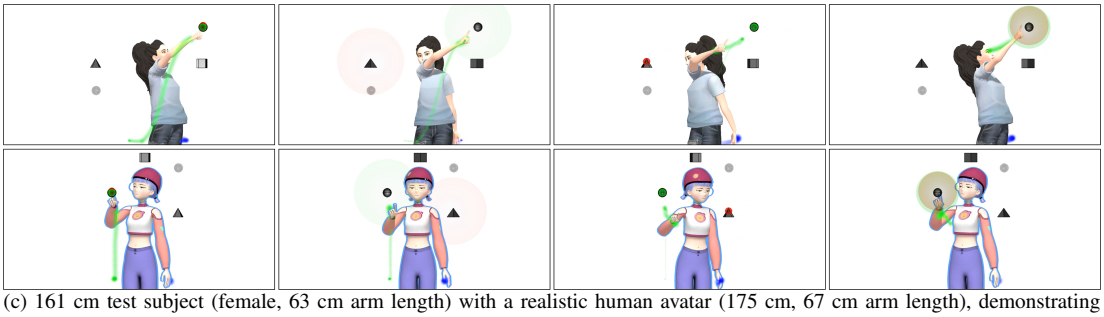
The ablation studies reveal the contribution of each component. Removing MD (w/o MD) yields the best TMP metrics but degrades FGD by more than 3× (1.01 → 3.34), the largest FGD gap among all ablations. This confirms that motion discretization acts as a training-time regularizer that suppresses jitter from unstable target estimation. Without it, target transitions push the generated motion distribution away from real human gesture statistics, as shown in the supplementary video. Removing PAD (w/o PAD) degrades DIP-PE and TMP, demonstrating that part-wise auxiliary decoders are essential for learning independent body-part controllability (Figure 8). The most significant degradation occurs when removing EE-Dual (w/o EE-Dual), where DIP and TMP metrics increase dramatically (Figure 9). This confirms that the dual end-effector representation is fundamental to our unsupervised



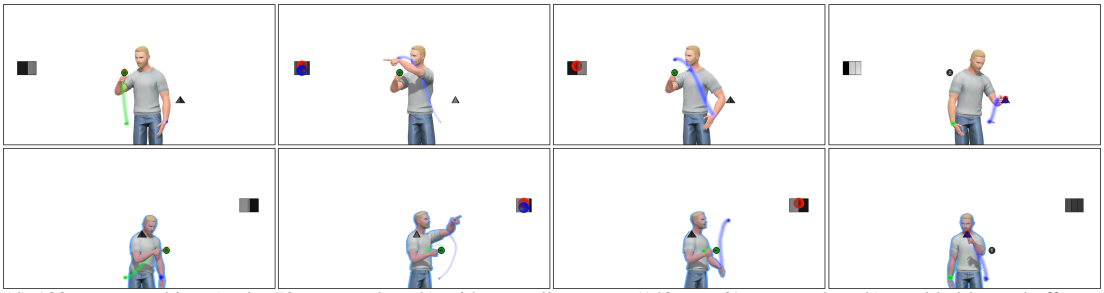
(a) 173 cm test subject (male, 68 cm arm length; from shoulder to fingertip) with a stylized cartoon avatar (186 cm, 81 cm arm length).



(b) 180 cm test subject (male, 75 cm arm length) with a humanoid robot avatar (189 cm, 77 cm arm length), showing potential applicability to teleoperated robots.



(c) 161 cm test subject (female, 63 cm arm length) with a realistic human avatar (175 cm, 67 cm arm length), demonstrating touching gestures via IK post-processing.



(d) 180 cm test subject (male, 72 cm arm length) with a smaller avatar (148 cm, 61 cm arm length), enabled by end-effector size adjustment.

Fig. 10. Real-time target estimation and part-wise motion translation results by our framework for various users (top) and avatars (bottom), shown in temporal order from left to right. Scenes contain 3–5 target objects (e.g., black sphere, black pyramid, black rectangle, gray sphere, gray pyramid) positioned differently for users and avatars. Predicted targets for the head (red), right hand (green), and left hand (blue) are shown. Smaller sphere sizes indicate higher target estimation consistency, whereas larger sizes reflect increased uncertainty during target transitions.

learning framework, enabling effective motion transfer across different target positions without paired training data.

For BP-GTNet, as the number of targets increases from 3 to 7, DIP and TMP metrics show slight increases while FGD remains stable or even slightly improves. This demonstrates that our framework maintains motion naturalness regardless of the number of candidate targets in the environment.

Figure 10 shows that our BP-GTNet successfully operates across diverse users with varying physical characteristics and different avatar configurations. We invite readers to view the supplementary video for real-time animation results, which also includes comparisons with the baseline and ablation studies.

VI. User Evaluation

We conducted a user evaluation to investigate how our framework supports live communication between two distant users in dissimilar XR telepresence environments.

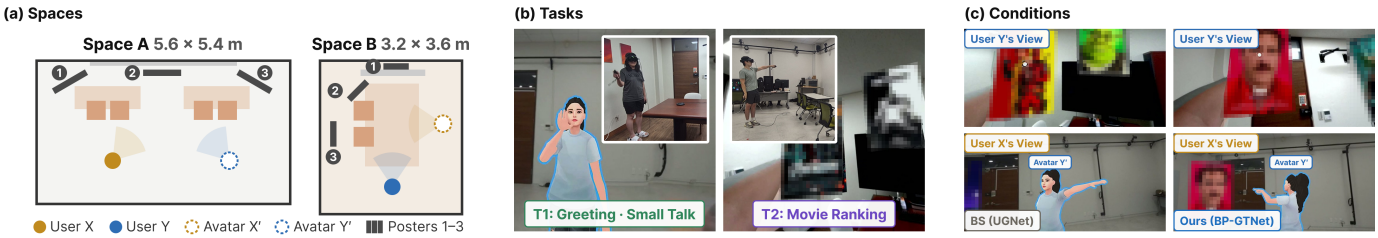


Fig. 11. Overview of the user study. (a) Top-down layouts of the two physical spaces with their different poster arrangements. (b) Task 1 (greeting and small talk) and Task 2 (movie preference ranking), shown from the counterpart's egocentric view with the actual participant in the inset. (c) The two conditions (left: BS, right: Ours). Within each column, the top row shows the egocentric view of the user pointing at a poster, and the bottom row shows the counterpart's egocentric view of that user's avatar at the same moment.

A. Methods

We employed a within-subjects design with two conditions differing only in the gesture translation network: the baseline UGNet [4] (**BS**) and BP-GTNet (**Ours**). Both conditions used TENet for target estimation, as UGNet assumes ground-truth targets unavailable in a live system. UGNet requires separate models for right-handed and left-handed users, so the left-handed participant was assigned the one trained on the mirrored dataset, whereas BP-GTNet uses a single model regardless of handedness.

B. Measures

Following prior avatar-mediated telepresence studies [1], [4], [18], we measured two aspects: the Temple Presence Inventory (TPI) [62], assessing perceived social presence between participants, and Message Understanding (MU) [63], assessing users' comprehension of their counterpart's messages. Seven additional custom items were included, each rated on a 7-point Likert scale (1 = strongly disagree, 7 = strongly agree): five asking participants to rate the counterpart avatar's upper-body gestures (Q1–Q5), and two assessing their own experience as a speaker (Q6–Q7):

- Q1: The avatar's upper-body gestures moved smoothly.
- Q2: The avatar's upper-body gestures looked natural and human-like.
- Q3: The avatar's upper-body gestures supported the flow of conversation.
- Q4: The avatar's upper-body gestures faithfully reflected the counterpart's communicative intent.
- Q5: The avatar's upper-body gestures clearly indicated which objects in my space they were referring to.
- Q6: I felt that my communicative intent was correctly conveyed to my counterpart.
- Q7: I did not need extra effort to make my counterpart understand which objects I was referring to.

C. Task

Each pair of participants performed two communication tasks (Figure 11 (b)) in both conditions. We designed the tasks to elicit both conversational and deictic upper-body gestures in natural interaction.

Task 1 (greetings and small talk). Participants greeted each other by waving at the counterpart's avatar, and then freely conversed on a small-talk topic provided by the experimenter.

Task 2 (movie preference ranking and discussion). Three virtual movie posters were placed in each participant's physical space. Participants took turns explaining their preference rankings with deictic gestures, and the experimenter encouraged them to alternate hands when pointing.

D. Procedure

Prior to the experiment, each participant completed a pre-survey in which they were given a list of well-known movies and asked to mark those they had watched. From the overlapping selections within each pair, we chose six movies and assigned three to each condition, ensuring that no movie poster was repeated across conditions.

Upon arrival at their respective spaces, participants signed consent forms and provided demographic information. After receiving instructions about the tasks, they put on the head-mounted display (HMD) and controllers and completed a calibration process.

Because the two rooms differed in dimensions, the avatars and virtual movie posters were arranged differently in each space to accommodate the physical constraints, creating dissimilar telepresence environments (Figure 11 (a)). In each condition, participants performed Task 1 and Task 2 in sequence, and then filled out questionnaires evaluating their experience. The condition order was randomized at the dyad level (i.e., both members of a dyad followed the same order).

After completing both conditions, the experimenter in each space conducted a semi-structured interview with the participant in that space, focusing on (1) the differences participants noticed between the two conditions and (2) how these differences affected their communication. The interviews were audio-recorded and transcribed. The first author, who did not conduct the interviews, read all transcripts, extracted statements relevant to these two questions, and inductively grouped them into the topics reported in Section VI-H. The total experiment duration was approximately 60 minutes. For additional details, please refer to the supplementary video.

E. Implementation

We implemented this multi-user prototype using the Unity 3D engine (6000.0.36f1) with the Meta SDK (83.0.1) and Depth API (77.0.0). Participants wore Meta Quest Pro HMDs, with user motion tracked via the HMD and its controllers. Each HMD was tethered via a link cable to a desktop in its

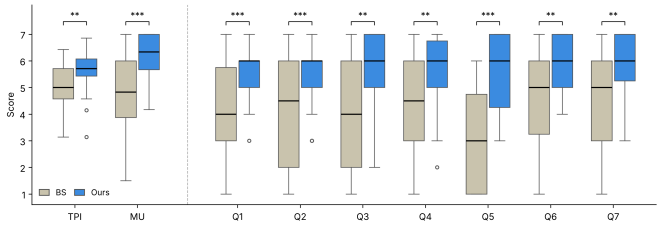


Fig. 12. Distributions of questionnaire responses in the two conditions (BS vs. Ours; N=26 per condition). Asterisks mark the condition effect (** $p_{\text{Holm}} < 0.01$, *** $p_{\text{Holm}} < 0.001$).

TABLE VI
QUESTIONNAIRE RESULTS (BS VS. OURS; N=26 PER CONDITION, MEAN$^{\pm\text{STD}}$): CONDITION EFFECT b WITH 95% CI, p_{HOLM}, AND EFFECT SIZE d.

Measure	BS	Ours	b [95% CI]	p_{Holm}	d
TPI	$5.049^{\pm 0.867}$	$5.577^{\pm 0.772}$	0.527 [0.233, 0.822]	0.003	0.641
MU	$4.769^{\pm 1.585}$	$6.212^{\pm 0.829}$	1.442 [0.878, 2.007]	<0.001	1.137
Q1	$4.154^{\pm 1.713}$	$5.654^{\pm 1.129}$	1.500 [0.861, 2.139]	<0.001	1.033
Q2	$4.115^{\pm 1.946}$	$5.538^{\pm 1.272}$	1.423 [0.765, 2.081]	<0.001	0.866
Q3	$4.115^{\pm 2.026}$	$5.577^{\pm 1.554}$	1.462 [0.636, 2.287]	0.003	0.804
Q4	$4.269^{\pm 1.801}$	$5.500^{\pm 1.449}$	1.231 [0.440, 2.021]	0.003	0.749
Q5	$2.885^{\pm 1.751}$	$5.654^{\pm 1.599}$	2.769 [1.959, 3.580]	<0.001	1.643
Q6	$4.731^{\pm 1.845}$	$6.038^{\pm 1.038}$	1.308 [0.634, 1.981]	0.002	0.870
Q7	$4.269^{\pm 2.089}$	$6.038^{\pm 1.076}$	1.769 [0.897, 2.642]	0.002	1.062

room with the same specification as in Section IV-C, which ran the Unity application and both networks locally with four candidate targets (three posters and the counterpart avatar). All avatars were created using Ready Player Me (7.4.0), with lip-sync animations driven by the Meta SDK's audio-based lip-sync. Network synchronization was implemented using Photon Fusion (2.0.9) and Photon Voice 2 (2.62). The two physical spaces measured 5.6 m × 5.4 m × 3.2 m and 3.2 m × 3.6 m × 2.5 m (width × depth × height), respectively.

F. Participants

We recruited 26 participants (13 dyads, 17 female, 9 male). Their ages ranged from 22 to 34 years (M=27.7, SD=3.8). We collected height (155 to 187 cm), arm length (63 to 79 cm), handedness (25 right, 1 left), and dominant eye (17 right, 9 left). Regarding prior experience, 25 participants had used VR before, while one had no experience; 22 participants had used mixed reality before, while 4 had no experience. Of the 13 dyads, 8 experienced BS first (BS→Ours) and 5 experienced Ours first (Ours→BS).

G. Results

Statistical analysis. Since the two members of a dyad are not independent, we analyzed each measure with a linear mixed-effects model with condition (BS vs. Ours) as a fixed effect and random intercepts for dyad and for participant nested within dyad, fitted by restricted maximum likelihood using lme4 [64] and lmerTest [65] with Satterthwaite's approximation for the degrees of freedom. A model that additionally included a dyad-level random slope of condition was singular for all measures except Q5 and yielded identical estimates of the condition effect, so we report the random-intercept models.

For each measure, we report the condition effect b (Ours−BS, in scale points) with its 95% CI based on the t-distribution with Satterthwaite degrees of freedom, and a standardized effect size d, defined as b divided by the square root of the sum of all variance components (dyad, participant, and residual) [66]. To control for multiple comparisons, the p-values of all nine measures (TPI, MU, and Q1–Q7) were corrected as a single family with the Holm–Bonferroni procedure; corrected p-values (p_{Holm}) are reported throughout.

To verify that the condition effect did not depend on condition order, we added order (BS→Ours vs. Ours→BS) and its interaction with condition to each model (Type III F-tests). The condition × order interaction was not significant for any measure (all $p \geq 0.22$), and the point estimate of the condition effect was positive in both order groups for every measure; we therefore report the models without order. A main effect of order was found for Q3–Q5, reflecting lower ratings in the Ours→BS dyads under both conditions; for Q5, the condition effect itself was nearly identical in the two order groups (Ours−BS: 2.75 vs. 2.80). Per-measure values are given in the supplementary material.

Internal consistency. Cronbach's α, computed separately for each condition, was acceptable for the TPI (BS: 0.75, Ours: 0.72) and high for the MU (0.96, 0.95). The custom items were analyzed individually rather than as a composite; for reference, α was 0.93 (BS) and 0.88 (Ours) for the five avatar-gesture items Q1–Q5, and the two speaker-experience items Q6 and Q7 were correlated at r=0.57 (BS) and r=0.86 (Ours).

Results. Figure 12 and Table VI summarize the questionnaire responses. Ours scored higher than BS on every measure, and all differences remained significant after Holm–Bonferroni correction (all $p_{\text{Holm}} < 0.01$), with effect sizes ranging from d=0.641 (TPI) to d=1.643 (Q5).

H. Discussion

Our results demonstrate that the choice of gesture translation network substantially affects communication quality in dissimilar telepresence environments. Both conditions shared the same system pipeline, including TENet-based target estimation, avatar appearance, voice, and lip-sync animation, isolating gesture translation as the source of the observed differences. As the two networks were compared as a whole, the user study establishes the overall advantage of BP-GTNet over UGNet; the item-wise interpretations below are therefore based on participants' comments.

First, Ours maintained referential clarity where BS did not (Figure 11 (c)). The largest effect was observed in Q5 (d=1.643), the only case in which the mean rating fell below the scale midpoint (BS: M=2.885), indicating that participants frequently failed to identify which object their counterpart was referring to. This failure was most pronounced when the task required independent control of body parts, such as alternating hands when pointing. Participants' feedback reflected this: "pointing with the left arm looked strange" (P1), and "the gaze and the gesture were misaligned" (P2). This referential

breakdown was accompanied by lower ratings on both sides of the interaction: listeners' comprehension degraded (MU, d=1.137), and speakers reported additional verbal effort (Q6–Q7), such as explaining verbally on top of gesturing (P2) or "repeatedly confirming which poster they were pointing at" (P4). In contrast, BP-GTNet retargets the head and both hands independently toward their respective estimated targets, and participants' comments reflected this clarity: "the target being pointed at was clearly conveyed, which made communication much easier" (P12).

Second, participants rated the translated motion of Ours as smoother, more natural, and more supportive of the conversation (Q1–Q3). The relatively large standard deviations in the BS condition suggest that UGNet's motion quality strongly depends on gesture type: when interactions mainly involved dominant-hand deictic gestures or remained relatively static, UGNet performed acceptably and even received high ratings from some participants. In more dynamic interactions, however, participants reported unstable motions: "the arm suddenly snapped into unnatural angles" (P18) and "the arm movement felt stiff" (P22). In contrast, BP-GTNet received consistently high ratings across gesture types (Q1, Q2).

Finally, while Ours achieved a higher TPI score, the effect was smaller than for MU (d=0.641 vs. 1.137), and BS also maintained a relatively high score (M=5.049). We attribute this to two factors. First, the shared audiovisual channels (voice communication and lip-sync animation) sustained a baseline sense of being together in both conditions. Second, gaze retargeting remained largely effective under both networks; one participant noted, "I could feel my partner's eyes looking at me, and this was similar in both conditions" (P17), although some perceived more accurate eye contact in Ours (P8). As P16 remarked, "feeling that we were in the same space and whether the interaction went smoothly were separate matters." Nevertheless, severe referential misalignment could break social presence itself: "my partner pointed at completely empty space, so it felt like we were in different places" (P13). This suggests that audiovisual channels and gaze establish social presence, but correct spatial translation of gestures is required to preserve it in avatar-mediated telepresence. Consistent with this, 25 out of 26 participants preferred Ours over BS, with the remaining participant reporting no noticeable difference.

Design alternatives. Instead of retargeting the user's motion itself, a referent could be conveyed in simpler ways. Explicit cues [19], [20] resolve the referent directly but leave the avatar's body uninformative; as P13 noted, an avatar pointing at empty space undermined co-location. Alternatively, the user could trigger a pre-defined pointing motion of the avatar toward the target once it is detected; this adds delay, discards the user's motion style, and cannot reproduce the continuous, body-part-wise gesture combinations observed in our study, such as alternating hands or gazing at one object while pointing at another. Empirically comparing these alternatives, including hybrids with explicit cues, remains future work.

VII. Application

We envision our gesture translation framework being utilized in XR avatar-mediated telepresence. To demonstrate its effectiveness, we present two scenarios: collaboration and teleoperation, both illustrated in the supplementary video.

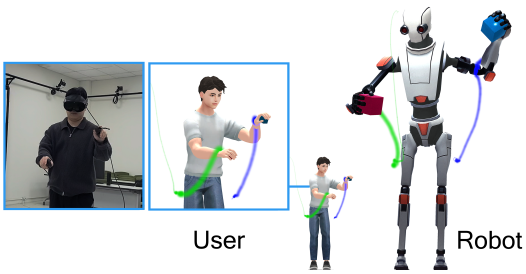


Fig. 13. Real-time teleoperation scenario demonstrating gesture translation across large scale differences. Both of the user's hand targets are estimated in real-time, and corresponding bimanual pick-and-place motions are generated for the robot despite differences in body size and object placement.

Collaboration. Figure 1 shows that two distant users conduct a chemistry lesson, where three virtual atoms are augmented at different positions and scales in each environment. Our framework enables natural teaching interactions as users explain molecular structures while gesturing toward different atoms simultaneously.

Teleoperation. Figure 13 demonstrates that a user remotely controls a humanoid robot avatar to perform bimanual pick-and-place tasks in an environment with different object positions and scales. Our part-wise translation enables precise coordination of both hands toward distinct targets despite the spatial dissimilarity.

VIII. Limitation

In this section, we discuss several limitations of our approach and outline future directions.

Automatic and dynamic target registration. Our system requires target candidates to be registered in advance at fixed positions, whereas in practice objects may be moved, added, or removed during the interaction. Since re-detecting all objects from the HMD vision stream in every frame is prohibitively expensive, a practical alternative is event-driven registration: a lightweight language model monitors the conversation and, when it indicates that relevant objects may have changed, triggers a vision model to verify the scene and update the candidate set.

One-to-one target correspondence. Our framework assumes that a correct one-to-one mapping between user and avatar targets is manually defined. An incorrect correspondence produces a natural gesture toward the wrong target, miscommunicating the user's intent, while a missing correspondence feeds a null target to BP-GTNet, yielding hand motion that mimics the user's pattern in an incoherent direction. A complete solution would establish and revise this correspondence continuously from spatial, semantic, and conversational cues as both environments evolve. Beyond such real-time matching, the reliability of each match should be an explicit input to both networks so that uncertain correspondences attenuate the gesture. Motion generation must also be extended to handle one-to-many, many-to-one, or missing correspondences, e.g., by arbitrating among plausible referents or falling back to target-agnostic motion.

Scalability of target estimation. TENet's joint estimation over all target combinations scales poorly as the number of candidates grows. Several pruning strategies could reduce this cost while preserving the joint formulation. Since not every target is a plausible referent for every body part, candidates can be pruned per body part; for example, a hand resting at the user's side is clearly not referencing any target, so all candidates for that hand can be excluded from the joint estimation. When targets are spatially clustered, a coarse-to-fine scheme can estimate the posterior at the cluster level and resolve individual targets only within the most probable cluster. Interaction context can prune candidates further, as discussed above for dynamic registration.

Dataset constraints. Our single-person dataset cannot cover the full spectrum of gestures arising in naturalistic dyadic interaction, and gestures deviating substantially from the training distribution may be translated with degraded quality; augmenting our training data with large-scale dyadic conversation datasets [67] could broaden gesture coverage. In addition, a generic IK solver does not reproduce subject-specific postural styles, such as habitual elbow elevation or torso involvement; sampling from a personalized neural upper-body pose prior could address this. Finally, the dataset covers only stationary upper-body motion; extending the framework to locomotion is a promising direction toward richer full-body telepresence.

Likewise, our FGD autoencoder is trained only on our data and has not been validated against external motion embeddings, so FGD should be read as a relative measure across methods; training it on such large-scale datasets could yield an embedding that better captures gesture naturalness.

Target distribution with larger differences. Our training data covers a 180° frontal range, and performance degrades outside it: target estimation worsens as the user's target is placed rearward, and the translated gestures deteriorate as the avatar's target moves toward its rear, even when the estimated target itself is accurate (see supplementary video).

A large discrepancy between the two target layouts poses a further challenge: when two successively referenced targets are close in the user's space but far apart in the avatar's space, the avatar's end-effector must traverse a much longer trajectory, forcing a trade-off between preserving the gesture's duration through unnaturally accelerated motion and preserving natural speed at the cost of a delay. Predicting the next target earlier, e.g., by extending TENet to estimate the target from the initial portion of the user's motion, would give the avatar a head start and thus alleviate this trade-off to some extent. However, no matter how early the prediction is made, a severe distance mismatch between the two layouts cannot be fully compensated for, and such cases remain a limitation of our approach.

Whether a given pair of layouts falls into such irrecoverable cases can be assessed before deployment. For a transition from target i to target j, assuming the avatar preserves the duration of the user's gesture, the required speed ratio equals the ratio of the corresponding inter-target distances, $r_{ij} = \|\mathbf{t}_i^{\mathrm{A}} - \mathbf{t}_j^{\mathrm{A}}\| / \|\mathbf{t}_i^{\mathrm{U}} - \mathbf{t}_j^{\mathrm{U}}\|$. The maximum and the spread of these ratios characterize the mismatch in complementary ways: $\max_{i,j} r_{ij}$ determines whether the end-effector can cover the required distances at a plausible speed, while the spread indicates how far the layouts deviate from a similarity transform, under which the ratios would be constant. Because which targets will be referenced in succession is unknown a priori, we evaluate the ratios over all pairs. When the maximum is moderate and the spread is small, a single global time-scaling of the translated motion suffices to handle every transition at a plausible speed; otherwise, some degree of acceleration or delay is unavoidable. Studying the perceptually tolerable range of these ratios, together with the resulting motion speed-up or delay, is thus important future work, as it would define the target layouts for which our approach remains applicable.

IX. Conclusion

We presented a deep neural network-based framework that performs part-wise translation of upper-body gestures via real-time target estimation for virtual avatars in dissimilar environments. Compared to the prior method that assumes predefined target positions and compresses motions into categorical values, our framework achieves real-time target estimation through TENet's normalizing flow-based posterior inference and preserves fine-grained motion details through BP-GTNet's unsupervised learning with dual end-effector representation. Furthermore, our part-wise approach enables individual controllability for each body part, resulting in more lifelike avatar movements. Our framework can be utilized in XR telepresence applications, and its effectiveness has been validated through a user evaluation.

Acknowledgement

This work was supported by NRF (RS-2025-25441313) and IITP (RS-2025-23524842), South Korea.

References

[1] 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.

[2] 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.

[3] E. Wong, A. Genay, J. E. S. Grønbæk, and E. Velloso, "Spatial heterogeneity in distributed mixed reality collaboration," in *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, 2025, pp. 1–19.

[4] J. Kang, T. Kim, H. Kim, and S.-H. Lee, "Real-time translation of upper-body gestures to virtual avatars in dissimilar telepresence environments," *IEEE Transactions on Visualization and Computer Graphics*, 2025.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] X. Huang, D. Frehlich, Z. Xia, P. Gholami, and R. Xiao, "Gaussian-nexus: Room-scale real-time ar/vr telepresence with gaussian splatting," in *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*, 2025, pp. 1–18.

[10] 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.

[11] 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.

[12] D. Kim, S. Kim, S. Choi, and W. Woo, "Spatial affordance-aware interactable subspace allocation for mixed reality telepresence," in *2024 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2024, pp. 1256–1265.

[13] 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.

[14] T. Kim, J. Shin, H. Kim, H. Jang, J. Kang, and S.-H. Lee, "Voronoi rooms: Dynamic visibility modulation of overlapping spaces for telepresence," *ACM Transactions on Graphics*, 2025.

[15] 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*, vol. 30, no. 12, pp. 7558–7570, 2024.

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

[17] 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.

[18] Y.-J. Li, H.-Z. Yang, W.-T. Shu, and M. Wang, "Semantics-aware avatar locomotion adaption for indoor cross-scene ar telepresence," *IEEE Transactions on Visualization and Computer Graphics*, 2025.

[19] S. Kim, G. Lee, W. Huang, H. Kim, W. Woo, and M. Billinghurst, "Evaluating the combination of visual communication cues for hmd-based mixed reality remote collaboration," in *Proceedings of the 2019 CHI conference on human factors in computing systems*, 2019, pp. 1–13.

[20] H. Bai, P. Sasikumar, J. Yang, and M. Billinghurst, "A user study on mixed reality remote collaboration with eye gaze and hand gesture sharing," in *Proceedings of the 2020 CHI conference on human factors in computing systems*, 2020, pp. 1–13.

[21] T. Piumsomboon, G. A. Lee, J. D. Hart, B. Ens, R. W. Lindeman, B. H. Thomas, and M. Billinghurst, "Mini-me: An adaptive avatar for mixed reality remote collaboration," in *Proceedings of the 2018 CHI conference on human factors in computing systems*, 2018, pp. 1–13.

[22] L. Yoon, D. Yang, C. Chung, and S.-H. Lee, "A mixed reality telepresence system for dissimilar spaces using full-body avatar," in *SIGGRAPH Asia 2020 XR*, 2020, pp. 1–2.

[23] L. Sidenmark, T. Zhang, L. Al Lababidi, J. Li, and T. Grossman, "Desk2desk: Optimization-based mixed reality workspace integration for remote side-by-side collaboration," in *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, 2024, pp. 1–15.

[24] A. Watkins, A. Ullal, and N. Sarkar, "Every "body" gets a say: An augmented optimization metric to preserve body pose during avatar adaptation in mixed/augmented reality," *IEEE transactions on visualization and computer graphics*, 2024.

[25] 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.

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

[27] 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.

[28] K.-J. Choi and H.-S. Ko, "Online motion retargetting," *The Journal of Visualization and Computer Animation*, vol. 11, no. 5, pp. 223–235, 2000.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

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

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] T. Jin, Y. Lee, and S.-H. Lee, "Interfacerays: Interaction-oriented furniture surface representation for human pose retargeting," in *Computer Graphics Forum*. Wiley Online Library, 2025, p. e70094.

[40] 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.

[41] 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.

[42] 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.

[43] T. Grossman and R. Balakrishnan, "The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2005, pp. 281–290.

[44] B. Ziebart, A. Dey, and J. A. Bagnell, "Probabilistic pointing target prediction via inverse optimal control," in *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*, 2012, pp. 1–10.

[45] X. Bi, Y. Li, and S. Zhai, "Ffitts law: modeling finger touch with fitts' law," in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2013, pp. 1363–1372.

[46] Y. Lu, C. Yu, and Y. Shi, "Investigating bubble mechanism for ray-casting to improve 3d target acquisition in virtual reality," in *2020 IEEE Conference on virtual reality and 3D user interfaces (VR)*. IEEE, 2020, pp. 35–43.

[47] T.-S. Dalsgaard, J. Knibbe, and J. Bergström, "Modeling pointing for 3d target selection in vr," in *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology*, 2021, pp. 1–10.

[48] 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.

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

[50] W. Xu, Y. Wei, X. Hu, W. Stuerzlinger, Y. Wang, and H.-N. Liang, "Predicting ray pointer landing poses in vr using multimodal lstm-based neural networks," in *2025 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2025, pp. 93–103.

[51] 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.
[52] Root-Motion, "Final-ik," 2017.
[53] S. Mayer, J. Reinhardt, R. Schweigert, B. Jelke, V. Schwind, K. Wolf, and N. Henze, "Improving humans' ability to interpret deictic gestures in virtual reality," in *Proceedings of the 2020 CHI conference on human factors in computing systems*, 2020, pp. 1–14.
[54] 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.
[55] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," *Advances in neural information processing systems*, vol. 31, 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] H. Zhang, S. Starke, T. Komura, and J. Saito, "Mode-adaptive neural networks for quadruped motion control," *ACM Transactions on Graphics (ToG)*, vol. 37, no. 4, pp. 1–11, 2018.
[58] 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.
[59] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in *International Conference on Learning Representations*, 2017.
[60] D. Lee, C. Kim, S. Kim, M. Cho, and W.-S. Han, "Autoregressive image generation using residual quantization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 523–11 532.
[61] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," in *International Conference on Learning Representations*, 2017.
[62] 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.
[63] 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
[64] D. Bates, M. Mächler, B. Bolker, and S. Walker, "Fitting linear mixed-effects models using lme4," *Journal of statistical software*, vol. 67, pp. 1–48, 2015.
[65] A. Kuznetsova, P. B. Brockhoff, and R. H. Christensen, "lmertest package: tests in linear mixed effects models," *Journal of statistical software*, vol. 82, pp. 1–26, 2017.
[66] C. M. Judd, J. Westfall, and D. A. Kenny, "Experiments with more than one random factor: Designs, analytic models, and statistical power," *Annual review of psychology*, vol. 68, pp. 601–625, 2017.
[67] V. Agrawal, A. Akinyemi, K. Alvero, M. Behrooz, J. Buffalini, F. M. Carlucci, J. Chen, J. Chen, Z. Chen, S. Cheng *et al.*, "Seamless interaction: Dyadic audiovisual motion modeling and large-scale dataset," *arXiv preprint arXiv:2506.22554*, 2025.



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 XR telepresence.



Jihun Shin 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 2025 and a B.S. degree in Oriental Painting from Hongik University, Korea, in 2023. His research interests include computer graphics and XR telepresence.



Kwang Bin Lee is a M.S. candidate with the Graduate School of Culture Technology at KAIST. He received a B.S. degree in Computer Science from Johns Hopkins University, USA, in 2017. His research includes embodied AI and scene-aware task planning.



Donghwan Shin is a M.S. candidate with the Graduate School of Culture Technology at KAIST. He received a B.S. degree in Computer Science and Electrical Engineering from KAIST, Korea, in 2025. His research interests include embodied AI and Agentic AI.



Yuna Choi is a M.S. candidate with the Graduate School of Culture Technology at KAIST. She received a B.S. degree in Computer Education from Sungkyunkwan University, Korea, in 2025. Her research interests include human animation and XR 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.

Part-wise Translation of Upper-body Gestures via Real-time Target Estimation for Virtual Avatars in Dissimilar Environments

Supplementary Material

I. Dataset Detail

Target configuration. For pointing scenarios, targets were arranged in a grid with azimuth spanning 180 (from -90° to 90° at 30° intervals), heights of -0.5m, 0m, and 0.5m relative to eye level, and a distance of 1.5m. Two targets were uniformly sampled from this grid, and one additional target was randomly sampled within the 180° frontal range at a height between -0.5m and 0.5m at the same distance. For touching scenarios, the grid shared the same azimuth configuration, with heights of -0.6m, -0.3m, and 0m relative to eye level and a distance of 0.45m. Two targets were sampled from this grid, and one additional target was randomly sampled within the 180° frontal range at a height between -0.3m and 0.3m and a distance of 1.5m, to represent a remote referent such as a conversational partner while touching a nearby object.

Inter-rater reliability of target annotations. All target labels used for training and evaluation were produced by a single (original) annotator following the criteria in Section III of the main manuscript. To assess their reliability, two additional annotators (A and B) independently labeled a subset of the newly captured portion of our dataset (Section III of the main manuscript), given only the written labeling protocol. The subset consists of 12 sequences containing various gesture types, including deictic gestures (both pointing and touching): 6 training sequences (165.3 s) and 6 test sequences (148.3 s), each test sequence from a different subject. In total, the subset comprises 28,224 frames (313.6 s) at the original 90 Hz capture rate, in which each annotator labeled the target (or null) of the head, right hand, and left hand at every frame.

Table I summarizes the agreement. Fleiss' κ across the three annotators was 0.953 (95% confidence interval, CI [0.926, 0.970]) and was consistent across body parts (head 0.957, right hand 0.939, left hand 0.958) and splits (training 0.951, test 0.955). Pairwise Cohen's κ ranged from 0.942 to 0.969. The original labels agreed with each new annotator (0.969 and 0.947) at least as well as the two new annotators agreed with each other (0.942), indicating no systematic deviation of the original labels. The original labels also matched the three-annotator majority vote on 99.2% of labels (A: 98.8%, B: 97.2%).

Of the remaining disagreements, 53–61% (depending on the annotator pair) occurred within ± 10 frames (± 111 ms) of a label boundary, i.e., differences in transition timing rather than in target identity. For hand onsets, annotator A placed onsets a median of 3–5 frames (33–56 ms) earlier and annotator B 4–7 frames (44–78 ms) later than the original annotator. Head onsets agreed within ± 2 frames (± 22 ms) across all annotators.

TABLE I
Inter-rater reliability of target labels. CIs were obtained by clip-level bootstrap resampling (2,000 resamples).

Metric	Value	95% CI
Fleiss' κ (All)	0.953	[0.926, 0.970]
Fleiss' κ (Head / Right Hand / Left Hand)	0.957 / 0.939 / 0.958	–
Cohen's κ (Original–A)	0.969	[0.949, 0.981]
Cohen's κ (Original–B)	0.947	[0.912, 0.971]
Cohen's κ (A–B)	0.942	[0.910, 0.963]
Majority Agreement (Original / A / B)	99.2 / 98.8 / 97.2 %	–

II. Input Feature Definition

We define three auxiliary input features used in TENet and BP-GTNet. These features are summarized in the main manuscript; here we provide their complete definitions:

- **Default feature $\mathbf{d} \in \mathbb{R}^3$** indicates whether each body part is in an active targeting state or a resting state. For the head, the feature is 0 if the current head direction deviates more than 10° from the forward direction at upright posture, and 1 otherwise. For each hand, the feature is 0 if the hand is raised more than 0.10m above its height at upright posture, and 1 otherwise. To ensure temporal stability, we average these binary values over the past 15 frames.
- **Ray feature $\mathbf{r} \in \mathbb{R}^3$** indicates whether each body part is directed toward any candidate target. For each candidate target position, we compute the angle between the head-to-target vector and each of the three directional rays (gaze and both EFRs). If any candidate target falls within 10° of a given ray, the corresponding feature is set to 1; otherwise, it is set to 0. As with the default feature, we average these values over the past 15 frames.
- **Target estimation consistency feature $\mathbf{c} \in \mathbb{R}^3$** measures how reliably the current estimated target has been maintained for each body part. The consistency value is reset to 0 when the target changes. Otherwise, it is updated by $\pm 1/15$ per frame (clamped to $[0, 1]$) based on gaze fixation, increasing when fixation is confirmed and decreasing otherwise. Gaze fixation is determined by whether the angle between the gaze ray and the head-to-target vector is within 10°. For hands, the value increases only when both the hand and head have non-null targets and gaze fixation is confirmed. This feature enables BP-GTNet to distinguish between confident targeting states and transitional phases, producing smooth motion even when TENet's estimation momentarily flickers.

III. Network Architecture Detail

A. TENet

TENet consists of an embedding network and a flow network. The embedding network transforms the input feature sequence $\mathbf{F}_t$ into a conditioning vector $\mathbf{q}_t \in \mathbb{R}^{256}$. First, the input is split by body part: each part receives a 4-dimensional feature per frame consisting of its direction vector and default feature. These are projected to 256 dimensions via a shared linear layer, then combined with learnable part embeddings and temporal positional embeddings through element-wise addition. The three sequences, each of length 45, are concatenated along the temporal axis to form a sequence of length $45 \times 3 = 135$, which is processed by a 6-layer Transformer encoder with 8 attention heads. We then extract the final temporal token from each body part's subsequence, concatenate them, and apply a linear projection to produce the conditioning vector $\mathbf{q}$.

The flow network consists of 5 stacked Glow blocks based on [1], where each block contains activation normalization, an invertible 1×1 convolution, and a conditional affine coupling layer. The coupling layer uses a 3-layer multilayer perceptron (MLP) with 128 hidden units to predict scale and translation parameters from $\mathbf{q}$.

To handle situations where a body part is not actively targeting any object (i.e., in a resting state), we introduce learnable null tokens for each body part. During training, when a body part has no assigned target, its target representation is replaced with the corresponding null token. This allows the model to explicitly distinguish between targeting a specific location and not targeting anything, preventing confusion between intentional rest poses and ambiguous gestures.

B. BP-GTNet

Encoder. The encoder uses learnable tokens for head ($\mathbf{e}^{\mathrm{H}} \in \mathbb{R}^{36}$), right hand ($\mathbf{e}^{\mathrm{R}} \in \mathbb{R}^{12}$), and left hand ($\mathbf{e}^{\mathrm{L}} \in \mathbb{R}^{12}$) respectively. These tokens serve as queries in the Transformer encoder, aggregating essential information from the input sequence to produce a compact representation of motion for each body part.

For the head, $\mathbf{x}^{\mathrm{H}}$ is projected to 36 dimensions via a linear embedding layer. Since $\mathbf{x}^{\mathrm{H}}$ is defined relative to the head target position, it exhibits discontinuities when the target changes. To handle this, we introduce learnable transition and default tokens ($\mathbf{e}^{\mathrm{trans}}, \mathbf{e}^{\mathrm{def}} \in \mathbb{R}^{36}$) that explicitly signal these discrete events to the network. For each frame t, the input sequence element is determined as:

$$\mathbf{s}_t^{\mathrm{H}} = \begin{cases} \mathbf{e}^{\mathrm{trans}} & \text{if } \mathbf{t}_t^{\mathrm{H}} \neq \mathbf{t}_{t-1}^{\mathrm{H}} \\ \mathbf{e}^{\mathrm{def}} & \text{if } \mathbf{t}_t^{\mathrm{H}} = \text{null} \\ \mathrm{Emb}(\mathbf{x}_t^{\mathrm{H}}) & \text{otherwise.} \end{cases} \tag{1}$$

The head motion feature is then computed as:

$$\mathbf{h}_t^{\mathrm{H}} = \mathrm{Transformer}([\mathbf{e}^{\mathrm{H}}; \mathbf{s}_{t-4}^{\mathrm{H}}, \ldots, \mathbf{s}_t^{\mathrm{H}}])_0 \tag{2}$$

where the subscript 0 denotes extracting the output corresponding to the first token. The Transformer encoder consists of 4 layers with single-head attention.

For the hands, $\mathbf{x}_t^{\mathrm{R}}$ and $\mathbf{x}_t^{\mathrm{L}}$ are continuous signals without target-induced discontinuities. Each hand representation is projected to 12 dimensions via separate linear embedding layers. Both hands share a single Transformer encoder to capture the bimanual coordination. The hand motion features are computed as:

$$[\mathbf{h}_t^{\mathrm{R}}; \mathbf{h}_t^{\mathrm{L}}] = \mathrm{Transformer}([\mathbf{e}^{\mathrm{R}}; \mathbf{e}^{\mathrm{L}}; \mathrm{Emb}(\mathbf{x}_t^{\mathrm{R}}); \mathrm{Emb}(\mathbf{x}_t^{\mathrm{L}})])_{0:2} \tag{3}$$

where the subscript 0:2 indicates extracting the first two output tokens. This Transformer encoder also consists of 4 layers with single-head attention.

Motion Discretization. For motion discretization, Gumbel-Softmax sampling uses a temperature of 5, reshaping the 36-dimensional head feature into 6 channels of 6-dimensional vectors before sampling. Residual Vector Quantization utilizes a codebook size of 128 and 5 residual layers, each trained separately on one hand.

Decoder. The MoE decoder's gating network is a 2-layer Transformer encoder with a 64-dimensional hidden size. It computes blending coefficients for 24 experts, each implemented as a 3-layer MLP. A learnable query token is prepended to the input sequence, and the Transformer output at this token position is projected to produce the blending coefficients ω_t:

$$\omega_t = \mathrm{GatingNet}(\tilde{\mathbf{T}}_t, \mathbf{R}_t, \mathbf{D}_t, \mathbf{C}_t) \tag{4}$$

$$\hat{\mathbf{y}}_{t+1}, \hat{\mathbf{g}}_{t+1}, \hat{\mathbf{i}}_{t+1} = \mathrm{MoE}(\mathbf{y}_t, \mathrm{sg}[\tilde{\mathbf{h}}_t], \tilde{\mathbf{T}}_t, \mathbf{R}_t, \mathbf{D}_t, \mathbf{C}_t; \omega_t) \tag{5}$$

where $\tilde{\mathbf{T}}_t$ denotes the target sequence in which each body part's target is replaced with a learnable null token when that part has a null target. $\mathrm{sg}[\cdot]$ denotes the stop-gradient operator.

Each auxiliary decoder follows the same architecture as the main decoder, but with a smaller capacity: 32-dimensional hidden size and 15 experts. It operates on single-part features:

$$\omega_t^{(p)} = \mathrm{GatingNet}^{(p)}(\tilde{\mathbf{T}}_t^{(p)}, \mathbf{R}_t^{(p)}, \mathbf{D}_t^{(p)}) \tag{6}$$

$$\bar{\mathbf{y}}_{t+1}^{(p)} = \mathrm{MoE}^{(p)}(\mathbf{y}_t^{(p)}, \tilde{\mathbf{h}}_t^{(p)}, \tilde{\mathbf{T}}_t^{(p)}, \mathbf{R}_t^{(p)}, \mathbf{D}_t^{(p)}; \omega_t^{(p)}) \tag{7}$$

where $p \in \{\mathrm{H}, \mathrm{R}, \mathrm{L}\}$ indexes the head, right hand, and left hand, respectively. These auxiliary outputs $\bar{\mathbf{y}}_{t+1}^{(p)}$ are used only during training and discarded at inference time.

a) Loss.: The total loss function is defined as:

$$\begin{aligned} L = {} & \|\mathbf{y}_{t+1} - \hat{\mathbf{y}}_{t+1}\|_1 + \|\mathbf{g}_{t+1} - \hat{\mathbf{g}}_{t+1}\|_1 + \|\mathbf{i}_{t+1} - \hat{\mathbf{i}}_{t+1}\|_1 \\ & + \sum_{p \in \{\mathrm{H},\mathrm{R},\mathrm{L}\}} \|\mathbf{y}_{t+1}^{(p)} - \bar{\mathbf{y}}_{t+1}^{(p)}\|_1 + \sum_{p \in \{\mathrm{R},\mathrm{L}\}} L_{\mathrm{VQ}}^{(p)} \end{aligned} \tag{8}$$

where the first three terms supervise the main decoder outputs, the fourth term provides auxiliary supervision for each body part, and $L_{\mathrm{VQ}}^{(p)}$ denotes the commitment loss for the RVQ of each hand.

IV. Training Detail

All experiments are conducted on an NVIDIA RTX 4090 GPU using the PyTorch framework.

A. TENet

We use the Adam optimizer with a learning rate of 10^{-4}, a batch size of 16, and an exponential learning rate decay schedule with a factor of 0.9 per epoch. Training takes approximately 5 hours.

For baseline comparisons, all methods are trained with the same training protocol:

- **Baseline** (28.2 MB) follows Moon et al. [2], using Real NVP [3] with an MLP encoder. The encoder consists of a 4-layer MLP with BatchNorm and ReLU activation, outputting a 360-dimensional conditioning vector. The flow network comprises 5 stacked affine coupling blocks, each containing a permutation layer and a conditional affine transformation. The conditioning network uses a 3-layer head network and a 3-layer main network with 128 hidden units and Tanh activation.
- **Diffusion** (26.8 MB) employs a residual MLP-based diffusion regression model. The decoder consists of 12 layers with a 512-dimensional hidden size, where each layer combines a main path and a residual path through a triple residual connection, using LayerNorm, Linear, and SiLU activation. Time embedding uses 64-dimensional sinusoidal positional embedding processed by a 2-layer MLP. Inference uses 20-step DDPM sampling with a cosine noise schedule.
- **TENet-Indep** (27.6 MB) shares the same Transformer encoder as TENet but uses three independent 3-dimensional Glow networks [1] for head, right hand, and left hand, respectively. Each flow network consists of 5 stacked Glow blocks, where each block contains activation normalization, invertible 1×1 convolution with LU decomposition, and a conditional affine coupling layer with 32 hidden units.

B. BP-GTNet

We train for 90 epochs with scheduled sampling [4] to bridge the gap between training and inference. The model is trained with ground-truth inputs for the first 30 epochs, then the ground-truth ratio is progressively reduced until epoch 60, after which the model relies entirely on its own predictions. Motion clips of 45 frames are extracted from captured sequences with 7-frame overlap. We shift by one frame at each training step to update the network weights. We use the AdamW optimizer with a learning rate of 10^{-4}, a batch size of 32, and an exponential learning rate decay schedule with a factor of 0.99 per epoch. Training takes approximately 50 hours.

V. Additional Statistics for the User Evaluation

Order effects. Table II reports the per-measure results of the order analysis summarized in Section VI-G of the main manuscript. For each measure, order (BS→Ours vs. Ours→BS) and its interaction with condition were added to the linear mixed-effects model (random intercepts for dyad and for participant nested within dyad, fitted by REML). The p-values are from Type III F-tests with Satterthwaite's approximation and are uncorrected. The condition effect within each order group is estimated as a simple-effect contrast of the same model, with a t-based 95% CI (Satterthwaite degrees of freedom).

TABLE II
Order analysis for each measure (8 BS→Ours and 5 Ours→BS dyads). p_{ORDER}, $p_{\text{COND}\times\text{ORDER}}$: main effect of order and condition × order interaction. b [95% CI]: condition effect (Ours−BS, in scale points) within each order group.

			b [95% CI] (Ours−BS)	
Measure	p_{order}	$p_{\text{cond}\times\text{order}}$	BS→Ours	Ours→BS
TPI	0.325	0.357	0.63 [0.26, 1.01]	0.36 [−0.12, 0.83]
MU	0.162	0.228	1.18 [0.46, 1.89]	1.87 [0.96, 2.77]
Q1	0.152	0.620	1.63 [0.80, 2.45]	1.30 [0.25, 2.35]
Q2	0.104	0.768	1.50 [0.64, 2.36]	1.30 [0.22, 2.38]
Q3	0.013	0.378	1.75 [0.69, 2.81]	1.00 [−0.34, 2.34]
Q4	0.023	0.510	1.44 [0.42, 2.45]	0.90 [−0.39, 2.19]
Q5	0.008	0.952	2.75 [1.69, 3.81]	2.80 [1.46, 4.14]
Q6	0.234	0.986	1.31 [0.43, 2.19]	1.30 [0.19, 2.41]
Q7	0.078	0.548	1.56 [0.43, 2.69]	2.10 [0.67, 3.53]

References

[1] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," *Advances in neural information processing systems*, vol. 31, 2018.

[2] 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.

[3] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," in *International Conference on Learning Representations*, 2017.

[4] 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.