HOIMotion: Forecasting Human Motion During Human-Object Interactions Using Egocentric 3D Object Bounding Boxes

Zhiming Hu, Zheming Yin, Daniel Haeufle, Syn Schmitt, Andreas Bulling

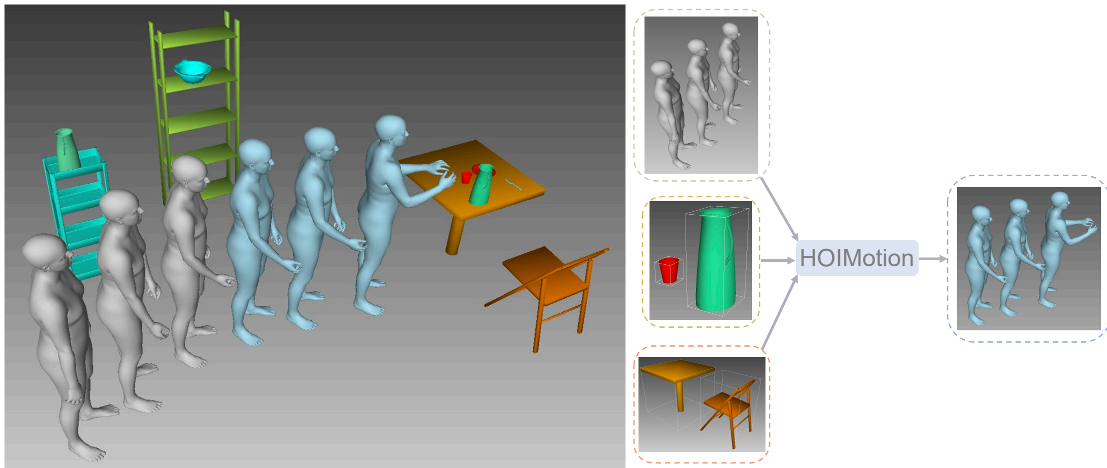

Fig. 1: HOIMotion is a novel method for forecasting human motion during daily human-object interaction activities. The left figure shows an example of daily pick activity. HOIMotion uses body poses in the past and the 3D bounding boxes of scene objects in the egocentric view to forecast future human motion and can achieve superior performances over prior methods that only use historical body poses.

Abstract—We present *HOIMotion* – a novel approach for human motion forecasting during human-object interactions that integrates information about past body poses and egocentric 3D object bounding boxes. Human motion forecasting is important in many augmented reality applications but most existing methods have only used past body poses to predict future motion. HOIMotion first uses an encoder-residual graph convolutional network (GCN) and multi-layer perceptrons to extract features from body poses and egocentric 3D object bounding boxes, respectively. Our method then fuses pose and object features into a novel pose-object graph and uses a residual-decoder GCN to forecast future body motion. We extensively evaluate our method on the Aria digital twin (ADT) and MoGaze datasets and show that HOIMotion consistently outperforms state-of-the-art methods by a large margin of up to 8.7% on ADT and 7.2% on MoGaze in terms of mean per joint position error. Complementing these evaluations, we report a human study (N=20) that shows that the improvements achieved by our method result in forecasted poses being perceived as both more precise and more realistic than those of existing methods. Taken together, these results reveal the significant information content available in egocentric 3D object bounding boxes for human motion forecasting and the effectiveness of our method in exploiting this information.

Index Terms—Human motion forecasting, human-object interaction, graph convolutional network, augmented reality

1 Introduction

Understanding and analysing human behaviour is a long-standing research challenge in virtual (VR) and augmented reality (AR) and is considered a crucial component for future human-aware intelligent VR/AR systems [13, 18]. Human motion forecasting in particular has significant relevance for a number of VR/AR applications including 1) redirected walking [5] that can redirect a user's walking path based on predicted future trajectories to create the illusion of an unlimited virtual interaction space; 2) collision avoidance [46] that can avoid potential collision between two users or between a user and the physical world by predicting future human motion and sending out a warning beforehand if a collision is likely to happen; 3) low-latency interaction [8] that can prepare the virtual content in advance based on predicted future human motion to provide users with a low-latency experience; as well as 4) assistive devices [4] that first predict users' desired future movements and then help them to accomplish them.

Human motion forecasting is typically formulated as a pose-focused sequence-to-sequence task, i.e. the task of using a sequence of past body poses to predict future poses. This approach works reasonably well given that human behaviour, particularly during procedural or repetitive tasks, has rich internal structure and consists of characteristic, and thus predictable, sequences of actions. Human motion, however, is also closely linked to the environment, especially during daily human-object interaction (HOI) activities. For example, as illustrated in Figure 1, if a user wants to pick up an object on the table, their body motion trajectories are strongly influenced by the location of the table while their arm movements are highly correlated with the specific position of the target object. Inspired by the close link between human motion and scene objects, in this work we explore the potential of using information on scene objects to improve motion forecasting. To ensure the

- *Zhiming Hu, Zheming Yin, Syn Schmitt, and Andreas Bulling are with the University of Stuttgart, Germany. E-mail: {zhiming.hu@vis.uni-stuttgart.de, st178328@stud.uni-stuttgart.de, schmitt@simtech.uni-stuttgart.de, andreas.bulling@vis.uni-stuttgart.de}.*
- *Daniel Haeufle is with Heidelberg University, Germany. E-mail: daniel.haeufle@ziti.uni-heidelberg.de.*
- *Daniel Haeufle, Syn Schmitt, and Andreas Bulling are with the Center for Bionic Intelligence Tuebingen Stuttgart (BITS), Germany.*
- *Zhiming Hu is the corresponding author.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

generalisability in various VR/AR scenarios, we propose to only use egocentric 3D object bounding boxes since such information is readily available in VR/AR systems [6, 18, 39].

Information on human body pose and egocentric 3D object bounding box are so different that we cannot directly integrate them into existing methods, whose architectures are designed to only model human motion [12, 30, 34, 35].To solve this problem, we present *HOIMotion* – a novel graph convolutional network-based (GCN-based) encoder-residual-decoder architecture that can efficiently combine historical body poses and egocentric 3D object bounding boxes to forecast human motion in the future. Our method first uses an encoder-residual GCN and multi-layer perceptrons (MLPs) to extract pose and object features respectively. These features are fused into a novel pose-object graph and a residual-decoder GCN is applied to forecast future body motion from the pose-object graph. We extensively evaluate our method at different future time horizons of up to 1 second (future 30 frames) on the Aria digital twin (ADT) dataset [38] for AR setting as well as on the MoGaze dataset [26] for real-world setting. Experimental results demonstrate that our method consistently outperforms several state-of-the-art methods that only use past body poses by a large margin, achieving an average improvement of 8.7% on ADT and 7.2% on MoGaze in terms of mean per joint position error (MPJPE). Using only past body poses as input, our method can achieve an average improvement of 5.3% on ADT and 3.7% on MoGaze, validating the effectiveness of our architecture. To qualitatively evaluate our method, we further conduct a user study and the responses from 20 users validate that our predictions are perceived as both more precise and more realistic than predictions of prior methods[1].

The specific contributions of our work are three-fold:

- We demonstrate the effectiveness of egocentric 3D object bounding boxes for motion forecasting, providing a new perspective for this challenging task.
- We propose a novel GCN-based encoder-residual-decoder architecture that uses an encoder-residual GCN and MLPs to extract pose and object features respectively, fuses these features into a pose-object graph, and applies a residual-decoder GCN to forecast future motion.
- We report extensive experiments on two public datasets for forecasting human motion at different future time horizons and demonstrate significant performance improvements over state of the art and report a user study that shows our method achieves superior performances over prior methods in both precision and realism.

2 Related Work

2.1 Human Behaviour Modelling

Understanding and modelling human behaviours is a long-standing research challenge in the areas of virtual and augmented reality [1, 3, 21, 23, 40, 44] and is considered a significant component for future human-aware intelligent VR/AR systems [13, 18]. Many researchers focused on visual attention modelling in virtual reality. Specifically, Sitzmann et al. modelled human visual saliency on 360-degree VR images and proposed to combine human head orientation to predict saliency map in VR [41]. Hu et al. focused on human gaze behaviours in VR and proposed several eye-head coordination models to predict eye gaze positions in free-viewing [20, 22] and task-oriented virtual environments [18]. Some researchers concentrated on modelling human cognitive load in virtual environments [9, 43]. For example, Tremmel et al. proposed to use electroencephalogram features to estimate cognitive load in an interactive virtual environment [43] while Dell'Agnola et al. extracted features from different physiological signals to detect the levels of cognitive load [9]. Hu et al. analysed different human activities in VR and proposed to use human eye and head movements to recognise user activities [19]. Kim et al. developed an electroencephalography-driven model to predict the degree of cybersickness in virtual environments [24]. In this work, we focused on human motion modelling, specifically predicting human full-body motion in the future.

2.2 Human Motion Forecasting

Human motion forecasting is a significant research topic in the area of human-centred computing and has great relevance for many VR/AR applications. Early works usually employed traditional machine learning methods to model human motion. Specifically, Wang et al. employed Gaussian process models to learn an effective representation of human motion data [45], Taylor et al. used a restricted Boltzmann machine to model the probability distribution of human body poses [42], while Lehrmann et al. proposed to model human motion through hidden Markov models [28]. While these early methods are useful for simple motions, they are less effective for forecasting more complex and long-term motion sequences [11]. Recently, with the rapid development of deep learning technology, many deep learning-based methods have been proposed to model human body motion. Considering the sequential structure of human motion, many researchers have explored to forecast human future motion using recurrent neural networks (RNNs) and have achieved superior performances over traditional methods [11, 27, 35]. In addition to RNNs, Transformers have also been applied to this task and have achieved good results [2, 34]. More recently, some researchers explored to forecast human motion using graph convolutional networks in light of the fact that human body pose can be viewed as a graph by treating each body joint as a graph node [7, 30]. To reduce the network complexity, multi-layer perceptrons have been proposed as a light-weight motion forecasting solution [12]. Existing motion forecasting methods typically only focused on human motion itself, forecasting future body poses using only historical poses. Recent work on offline human motion synthesis used the features from a global 3D scene point cloud to synthesise human motion [31, 46]. However, such features are difficult to acquire in many real application scenarios, especially in augmented or mixed reality settings, thus limiting the usefulness of these methods in real-day life outside clearly defined environments. In contrast with previous work, in this work we focus on real-time motion forecasting and combine body poses in the past and information on egocentric 3D object bounding boxes, which is readily available in VR/AR systems, to predict human motion in the future.

2.3 Human-object Interaction

Human-object interaction is a crucial interaction paradigm in virtual and augmented reality [15–17, 37]. Recent research has revealed the strong correlation between human behaviours and the scene objects during daily human-object interaction activities. Specifically, Hu et al. studied the visual search setting where users were required to search for a specific target object among many distractors and found that both the target and distractors have a strong influence on human gaze behaviours [18]. Li et al. revealed that users' spatial memory of the scene content influences their visual search strategies in large-scale immersive virtual environments [29]. David-John et al. found that during an item-selection activity in virtual reality human eye gaze is closely linked with the items that users intend to select [8]. Koulieris et al. revealed that during the process of game play, player actions are highly correlated with the present states of the game-related objects [25]. Emery et al. investigated open-ended VR games that covered various HOI activities such as shooting and object manipulation and revealed that human eye, head, and hand movements are strongly linked with the scene objects [10]. Inspired by the close link between human behaviours and scene objects, in this work we introduce to use information on scene objects to forecast human motion during human-object interactions.

3 Method

3.1 Problem Definition

We define egocentric scene object-aware human motion forecasting as the task of predicting a sequence of future body poses jointly from historical body poses and information on scene objects in the egocentric view. We use the 3D positions of all human body joints to represent body pose $p \in R^{3\times n}$, where n is the number of joints. We

[1] Source code and trained models will be released upon acceptance.

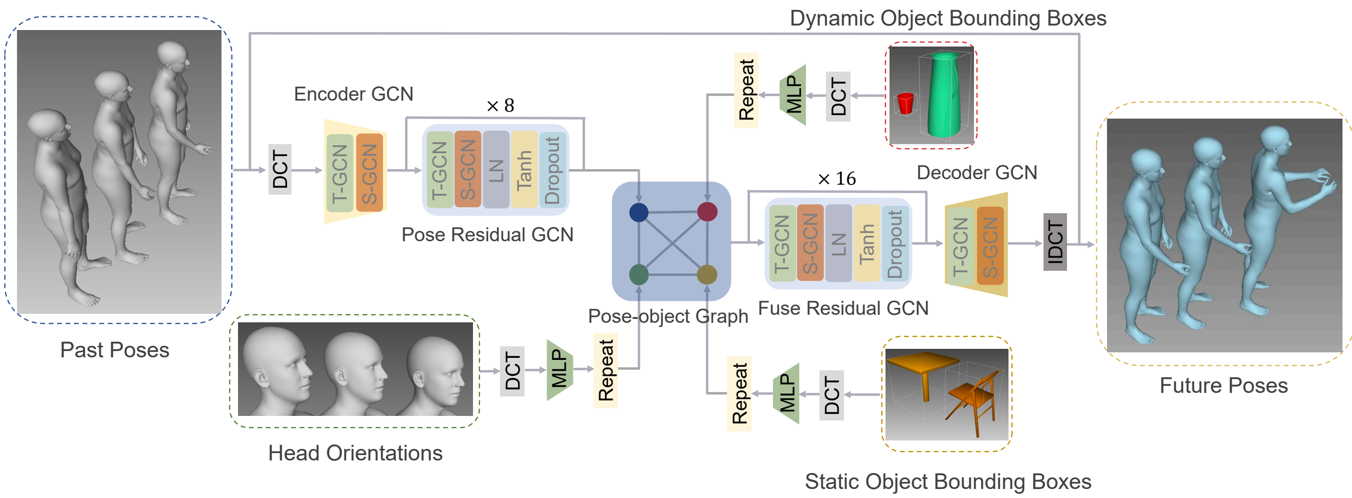

Fig. 2: Our method for egocentric scene object-aware human motion forecasting uses an encoder GCN and a pose residual GCN to extract features from historical body poses and employs three MLPs to respectively extract features from head orientations, and the bounding boxes of static and dynamic scene objects in the egocentric view. The pose, head, and object features are fused into a novel pose-object graph and a fuse residual GCN and a decoder GCN are applied to forecast future body motion from the pose-object graph.

represent scene objects using their bounding boxes given that the bounding box information can be easily and efficiently accessed in VR/AR systems and is highly relevant to human motion. Specifically, we use the 3D positions of the bounding box's eight vertices to represent scene objects $o \in R^{3\times 8\times m}$, where m is the number of objects. Considering that a user's egocentric viewport is determined by their head orientation in VR/AR systems, we also introduce to use human head orientation $h \in R^3$ in this task, where h is a unit vector indicating head forward direction. Given a sequence of historical body poses $P_{1:t} = \{p_1, p_2, ..., p_t\}$, head orientations $H_{1:t} = \{h_1, h_2, ..., h_t\}$, and scene objects $O_{1:t} = \{o_1, o_2, ..., o_t\}$, the task is to forecast body poses in the future $P_{t+1:T} = \{p_{t+1}, p_{t+2}, ..., p_T\}$. The core of our method is a GCN-based encoder-residual-decoder architecture. An encoder GCN and a pose residual GCN are used to extract features from historical body poses while MLPs are employed to extract features from head orientations and scene objects. The pose, head, and object features are fused into a novel pose-object graph, and a fuse residual GCN and a decoder GCN are applied to forecast future body motion from the pose-object graph (see Figure 2 for an overview of our method).

3.2 Pose Feature Extraction

To cover the future time horizon that we want to predict, we first padded the historical body poses $P_{1:t} \in R^{3\times n\times t}$ from a temporal length of t to T by repeating the last observed pose p_t for $T-t$ times following prior works [30, 33]. We applied discrete cosine transform (DCT) to encode the padded body poses $P \in R^{3\times n\times T}$ in the temporal domain in light of the good performance of DCT for encoding time series data [30, 32]:

$$P_{dct} = PM_{dct}, \tag{1}$$

where $M_{dct} \in R^{T\times T}$ is the DCT matrix and $P_{dct} \in R^{3\times n\times T}$ is the transformed body poses. We further proposed two GCN blocks, i.e. an encoder GCN block and a pose residual GCN block, to extract features from the transformed body pose data.

Encoder GCN The encoder GCN block aims at mapping the body pose data from its original space into a latent feature space. This block first employs a temporal GCN (T-GCN) to extract temporal features from the transformed pose data P_{dct}. The T-GCN views the body pose data as a fully-connected temporal graph that contains T nodes corresponding to T time steps of the pose data. The core of the T-GCN is a weighted adjacency matrix $A^T \in R^{T\times T}$ that learns the correlations between different temporal nodes and performs temporal convolution on the pose data:

$$f_{temp} = P_{dct}A^T. \tag{2}$$

$f_{temp} \in R^{3\times n\times T}$ was then permuted to $f_{temp} \in R^{T\times n\times 3}$ and a weight matrix $W^{start} \in R^{3\times 16}$ was used to map the original node features (3 dimensions) into latent space (16 dimensions):

$$f_{lat} = f_{temp}W^{start}, \tag{3}$$

where $f_{lat} \in R^{T\times n\times 16}$ represents the latent features. After the weight matrix, a spatial GCN (S-GCN) was employed to extract spatial features from the pose data. The S-GCN views the body pose data as a full-connected spatial graph that contains n nodes corresponding to n human body joints. The core of the S-GCN is a weighted adjacency matrix $A^S \in R^{n\times n}$ that learns the link between different spatial nodes and performs spatial convolution on the pose data:

$$f_{spa} = A^S f_{lat}. \tag{4}$$

$f_{spa} \in R^{T\times n\times 16}$ was further permuted to $f_{spa} \in R^{16\times n\times T}$ for further processing.

Pose Residual GCN The pose residual GCN block is designed to further enhance the body pose features. It first copies the body pose features along the temporal dimension ($R^{16\times n\times T} \rightarrow R^{16\times n\times 2T}$) to enhance the temporal features [30]. It then applies eight GCN components to further process the body pose data. Each GCN component contains a temporal GCN that learns the temporal features using the temporal adjacency matrix $A^T \in R^{2T\times 2T}$, a weight matrix $W^{res} \in R^{16\times 16}$ that learns latent features, a spatial GCN that extracts spatial features through the spatial adjacency matrix $A^S \in R^{n\times n}$, a layer normalisation (LN), a Tanh activation function, as well as a dropout layer with a dropout rate of 0.3 to prevent the GCN from overfitting. A residual connection was added for each GCN component to improve the network flow. The output of the pose residual GCN block was cut in half in the temporal dimension to obtain the body pose features $f_{pose} \in R^{16\times n\times T}$ that maintain the same temporal length as the original input to this block.

3.3 Head Feature Extraction

We first padded the historical head orientations $H_{1:t} \in R^{3\times t}$ to a temporal length of T by repeating the last observed head orientation h_t for $T-t$ times. A DCT is then applied to encode the head orientations in the temporal domain. We further used a multi-layer perceptron to process the head orientations given the effectiveness of MLP for encoding the sequences of human behaviour data [19, 20]. The MLP aims at mapping the head features (3 dimensions) at each time step into latent space (16 dimensions). Specifically, we used three linear layers with 128, 128, 16 neurons respectively to extract features from head orientations. The first two linear layers were followed by a layer normalisation, a Tanh activation function, as well as a dropout layer with a dropout rate of 0.5 to avoid overfitting while a layer normalisation and a Tanh activation function were employed after the third linear layer. Through the multi-layer perceptron, we obtained the processed head orientation features $f_{head} \in R^{16\times T}$.

3.4 Object Feature Extraction

Considering that different type of objects may have different influences on human motion during human-object interaction activities, we grouped the scene objects into two categories, i.e. *dynamic* and *static* objects, following prior work on object segmentation [38]. Dynamic objects refer to the objects that users can manipulate to change their positions during an HOI activity, e.g. the cup and jug in Figure 1, while static objects denote the objects that are stationary throughout the HOI activity, e.g. the table and chair in Figure 1. An environment may contain plenty of scene objects, many of which may have little influence on human body motion. To improve our method's efficiency, we only used the scene objects that were located in the central region of users' viewport since these central objects are more likely to influence human behaviours [20, 22, 41]. Specifically, at each time step we calculated the angular distance between the centre of each scene object and the centre of the viewport and ranked the dynamic and static objects respectively based on their angular distances from the viewport centre. We then selected the two dynamic objects $D_{1:t} = \{d_1, d_2, ..., d_t\} \in R^{3\times 8\times 2\times t}$ and two static objects $S_{1:t} = \{s_1, s_2, ..., s_t\} \in R^{3\times 8\times 2\times t}$ that are closest to the viewport centre and used their bounding box information for motion forecasting.

To cover the predicted future time horizon, we padded the object information to a temporal length of T by respectively repeating the last observed objects d_t and s_t for $T-t$ times. We then applied a DCT to encode the scene objects in the temporal domain and further used two MLPs to extract features from dynamic and static objects respectively. The two MLPs have the same structure and are designed to map the object features ($3\times 8\times 2$ dimensions) at each time step into latent space (16 dimensions). Each MLP contains three linear layers with 128, 128, 16 neurons respectively. A layer normalisation, a Tanh activation function, and a dropout layer with a dropout rate of 0.5 were applied after the first two linear layers while the third linear layer was followed by a layer normalisation and a Tanh activation function. Through the two MLPs, we obtained the processed features of both dynamic objects $f_{dynamic} \in R^{16\times T}$ and static objects $f_{static} \in R^{16\times T}$.

3.5 Pose-object Fusion

After the feature extraction process, we obtained the body pose features $f_{pose} \in R^{16\times n\times T}$, head orientation features $f_{head} \in R^{16\times T}$, dynamic object features $f_{dynamic} \in R^{16\times T}$, as well as static object features $f_{static} \in R^{16\times T}$. To enhance the head and object features, we respectively repeated the head and object features for four times along the spatial domain and obtained $f_{head} \in R^{16\times 5\times T}$, $f_{dynamic} \in R^{16\times 5\times T}$, and $f_{static} \in R^{16\times 5\times T}$. We then concatenated the pose, head, and object features along the spatial domain and obtained $f \in R^{16\times(n+15)\times T}$. To fuse different features for motion forecasting, we proposed a novel spatio-temporal pose-object graph: the temporal graph covers T nodes that correspond to the features at T time steps while the spatial graph contains $n+15$ nodes corresponding to the features of the body joints (n nodes), head orientations (5 nodes), dynamic objects (5 nodes), and static objects (5 nodes). Both the temporal and spatial graphs are fully-connected with their adjacency matrices measuring the weights between each pair of nodes.

3.6 Motion Forecasting

We further employed a fuse residual GCN block and an end GCN block to forecast future body poses from the pose-object graph.

Fuse Residual GCN The fuse residual GCN block aims at enhancing the fused pose-object features. It first copies the pose-object features along the temporal dimension ($R^{16\times(n+15)\times T} \rightarrow R^{16\times(n+15)\times 2T}$) to enhance the temporal features and then applies 16 GCN components to further process the pose-object data. Each GCN component consists of a temporal GCN that learns the temporal features using the temporal adjacency matrix $A^T \in R^{2T\times 2T}$, a weight matrix $W^{res} \in R^{16\times 16}$ that learns latent features, a spatial GCN that extracts spatial features through the spatial adjacency matrix $A^S \in R^{(n+15)\times(n+15)}$, a layer normalisation, a Tanh activation function, and a dropout layer with a dropout rate of 0.3 to avoid overfitting. A residual connection was added for each GCN component to improve the network flow. We cut the output of the fuse residual GCN block in half in the temporal dimension to maintain the same temporal length as the original input and obtained the spatio-temporal pose-object features $f \in R^{16\times(n+15)\times T}$.

Decoder GCN The decoder GCN block is employed to map the processed pose-object features from latent feature space to the original space. The decoder GCN consists of a temporal GCN that learns the temporal adjacency matrix, a weight matrix $W^{end} \in R^{16\times 3}$ that maps the latent features to three dimensions, and a spatial GCN that learns the spatial adjacency matrix. The output of the decoder GCN $Y_d \in R^{3\times(n+15)\times T}$ was converted back to the original representation space using an inverse discrete cosine transform (IDCT) matrix $M_{idct} \in R^{T\times T}$:

$$Y = Y_d M_{idct}. \tag{5}$$

We finally added a global residual connection between the pose input and the output of IDCT to obtain the predicted future poses $\hat{P}_{t+1:T} \in R^{3\times n\times T-t}$.

3.7 Loss Function

To ensure the precision and smoothness of our predictions, we employed a combination of motion loss ℓ_m and velocity loss ℓ_v as our loss function ℓ:

$$\ell = \ell_m + \ell_v. \tag{6}$$

ℓ_m is designed to measure our method's precision by calculating the mean per joint position error between the ground truth and the predicted future poses [30, 32]:

$$\ell_m = \frac{1}{n(T-t)} \sum_{i=t+1}^{T} \sum_{j=1}^{n} \|p_{i,j} - \hat{p}_{i,j}\|^2, \tag{7}$$

where $p_{i,j} \in R^3$ represents the ground truth 3D coordinates of the j^{th} joint at the future time of i while $\hat{p}_{i,j} \in R^3$ is the prediction of our method. ℓ_v aims at measuring the smoothness of our predictions by computing the mean per joint velocity error between the ground truth and the predicted future poses [12]:

$$\ell_v = \frac{1}{n(T-t-1)} \sum_{i=t+1}^{T-1} \sum_{j=1}^{n} \|v_{i,j} - \hat{v}_{i,j}\|^2, \tag{8}$$

where $v_{i,j} \in R^3$ is the ground truth pose velocity and $\hat{v}_{i,j} \in R^3$ represents the predicted pose velocity. The velocity is computed using the difference between two adjacent poses: $v_{i,j} = p_{i+1,j} - p_{i,j}$ and $\hat{v}_{i,j} = \hat{p}_{i+1,j} - \hat{p}_{i,j}$.

4 Experiments and Results

In this section, we conducted extensive experiments to evaluate our method's motion forecasting performance. Specifically, we first compared our method with the state-of-the-art methods that only use historical body poses on an AR dataset as well as on a real-world dataset. We further performed extensive ablation studies to validate the effectiveness of each component used in our method. We finally conducted a user study to qualitatively evaluate our method.

Table 1: Mean per joint position errors (unit: millimeters) of different methods for motion forecasting on the ADT and MoGaze datasets. Results are shown for different future time horizons of up to 1 second. Best results are in bold. Our method consistently outperforms prior methods in terms of average performance as well as performances at different time intervals. Even using only historical body poses as input, our method still achieves significantly better performances over prior methods.

Action	Method	100 ms	200 ms	300 ms	400 ms	500 ms	600 ms	700 ms	800 ms	900 ms	1000 ms	Average
ADT-work	*Res-RNN* [35]	28.4	38.7	47.8	58.1	67.3	76.7	85.9	95.1	104.2	113.1	69.1
	siMLPe [12]	26.0	28.4	33.2	39.8	47.9	55.1	62.5	71.4	79.4	89.4	51.2
	HisRep [34]	6.9	12.8	18.8	24.8	31.0	37.4	44.1	50.9	57.8	65.3	32.8
	PGBIG [30]	8.2	13.5	19.2	24.9	30.9	37.2	43.6	50.1	57.0	64.5	32.9
	Ours *pose only*	5.1	10.4	16.1	22.2	28.6	35.3	42.3	49.5	56.9	64.6	30.9
	Ours	**4.9**	**10.0**	**15.6**	**21.6**	**27.8**	**34.3**	**41.1**	**48.0**	**55.2**	**62.5**	**30.0**
ADT-decoration	*Res-RNN* [35]	22.5	33.4	46.2	59.7	73.5	87.4	101.5	115.6	129.8	144.2	77.4
	siMLPe [12]	27.2	33.6	44.6	56.7	70.6	85.4	101.2	118.5	136.9	156.6	78.7
	HisRep [34]	10.5	19.2	27.9	37.2	47.5	58.8	71.3	84.9	99.0	114.4	53.2
	PGBIG [30]	10.2	18.7	27.0	35.7	45.5	56.5	68.6	81.9	96.2	111.7	51.5
	Ours *pose only*	6.9	14.2	22.5	32.0	42.6	54.4	67.2	81.1	95.6	110.9	49.0
	Ours	**6.6**	**13.5**	**21.5**	**30.6**	**40.8**	**52.2**	**64.7**	**77.8**	**91.6**	**105.7**	**46.9**
ADT-meal	*Res-RNN* [35]	18.7	27.8	39.3	52.0	65.4	79.6	94.5	110.0	125.8	141.9	71.5
	siMLPe [12]	26.7	29.9	37.0	46.2	57.1	68.4	81.1	94.9	108.2	122.7	63.9
	HisRep [34]	8.0	15.0	22.4	30.4	39.1	48.6	58.6	69.2	80.0	91.3	43.2
	PGBIG [30]	8.5	15.2	22.2	29.7	38.0	47.1	56.8	67.0	77.9	89.7	42.3
	Ours *pose only*	5.6	11.6	18.6	26.6	35.4	44.9	55.2	66.1	77.4	89.0	40.0
	Ours	**5.3**	**11.2**	**18.0**	**25.8**	**34.4**	**43.8**	**53.8**	**64.2**	**75.1**	**86.2**	**38.9**
ADT-all	*Res-RNN* [35]	23.7	33.9	44.8	56.8	68.6	80.8	93.1	105.7	118.3	131.1	72.3
	siMLPe [12]	26.6	30.4	37.8	46.8	57.5	68.2	79.7	92.5	105.3	119.5	63.2
	HisRep [34]	8.3	15.4	22.6	30.2	38.4	47.2	56.6	66.6	76.8	87.8	42.0
	PGBIG [30]	8.9	15.5	22.4	29.6	37.4	46.0	55.0	64.7	75.0	86.2	41.3
	Ours *pose only*	5.8	11.9	18.8	26.4	34.8	43.9	53.6	63.9	74.7	85.8	39.1
	Ours	**5.5**	**11.4**	**18.1**	**25.6**	**33.7**	**42.5**	**52.0**	**61.8**	**72.0**	**82.5**	**37.7**
MoGaze-pick	*Res-RNN* [35]	36.1	50.1	67.9	88.1	110.6	135.1	161.3	189.1	218.1	248.0	123.6
	siMLPe [12]	27.4	37.7	51.5	67.4	84.7	103.8	125.1	147.4	171.2	196.1	95.4
	HisRep [34]	14.7	27.7	41.1	55.8	72.1	90.1	109.6	130.3	151.9	174.1	80.9
	PGBIG [30]	13.9	26.3	39.2	53.3	69.4	87.0	105.8	125.8	146.8	168.0	78.0
	Ours *pose only*	12.2	23.7	36.4	50.4	66.2	83.6	102.4	122.0	142.5	163.4	74.8
	Ours	**11.3**	**22.6**	**34.9**	**48.8**	**64.3**	**81.3**	**99.8**	**119.1**	**139.1**	**159.3**	**72.7**
MoGaze-place	*Res-RNN* [35]	42.9	58.6	77.1	97.0	118.1	140.1	162.4	184.5	206.4	227.6	125.6
	siMLPe [12]	31.3	45.9	62.8	80.3	98.3	118.0	139.6	162.1	185.5	210.0	107.1
	HisRep [34]	21.6	38.3	53.4	69.3	86.5	105.2	124.8	144.6	164.5	184.9	93.2
	PGBIG [30]	19.9	35.2	50.1	65.8	82.8	101.1	120.1	139.6	159.0	177.9	89.3
	Ours *pose only*	18.0	32.8	47.7	63.4	80.4	98.4	117.0	135.9	155.2	174.4	86.6
	Ours	**16.7**	**31.1**	**45.6**	**60.9**	**77.1**	**94.2**	**111.9**	**129.9**	**148.0**	**165.5**	**82.6**
MoGaze-all	*Res-RNN* [35]	38.5	53.1	71.1	91.3	113.2	136.8	161.7	187.5	214.0	240.8	124.3
	siMLPe [12]	28.8	40.6	55.5	72.0	89.4	108.8	130.2	152.6	176.3	201.0	99.5
	HisRep [34]	17.1	31.4	45.4	60.5	77.1	95.4	115.0	135.3	156.4	177.9	85.3
	PGBIG [30]	16.0	29.4	43.0	57.7	74.1	92.0	110.8	130.7	151.1	171.5	82.0
	Ours *pose only*	14.3	26.9	40.4	55.0	71.2	88.8	107.5	126.9	147.0	167.3	79.0
	Ours	**13.2**	**25.6**	**38.6**	**52.9**	**68.7**	**85.7**	**103.9**	**122.7**	**142.0**	**161.3**	**76.1**

4.1 Datasets

To test our method's generalisation capability for different settings, we evaluated our method on two public datasets including an AR dataset (Aria digital twin [38]) and a real-world dataset (MoGaze [26]).

ADT dataset [38] The Aria digital twin dataset is collected in two indoor environments (an apartment and an office environment) with simulated scene objects and contains human pose data performing various human-object interaction activities including *room decoration*, *meal preparation*, and *work*. Each human pose consists of the 3D coordinates of 21 human joints recorded at 30 Hz. The bounding box information and motion type (*dynamic* or *static*) of the scene objects are also provided. For experiments on ADT, we randomly selected 24 sequences for training and 10 sequences for testing.

MoGaze dataset [26] The MoGaze dataset is collected in a real-world indoor environment and contains human motion data recorded at 120 Hz from six people performing daily *pick* and *place* activities. The bounding box information and motion type (*dynamic* or *static*) of the scene objects are also recorded at 120 Hz. We down-sampled the human pose and object data to 30 Hz for simplicity [12, 30] and represented human poses using the 3D coordinates of 21 human joints. To evaluate motion forecasting on MoGaze, we used a leave-one-person-out cross-validation: We trained on five participants from scratch and tested on the remaining one, repeated the experiment six times with a different participant for testing, and calculated the average performance across all six iterations.

4.2 Evaluation Settings

Evaluation Metric As is common in human motion forecasting [12, 30, 46], we used the mean per joint position error (see Equation 7) in millimeters as our metric to evaluate different motion forecasting methods.

Baselines We compared our method with the following methods because they are not only prior state-of-the-art motion forecasting methods but also representatives of different network architectures, i.e. RNN, MLP, Transformer, and GCN:

- *Res-RNN* [35]: *Res-RNN* is a RNN-based method that applies a residual connection between the input pose and output pose to improve performance.
- *siMLPe* [12]: *siMLPe* is a light-weight MLP-based method that applies discrete cosine transform and residual connections to improve performance.
- *HisRep* [34]: *HisRep* is a Transformer-based method that extracts motion attention to capture the similarity between the current motion context and the historical motion sub-sequences.
- *PGBIG* [30]: *PGBIG* is a GCN-based method that employs a multi-stage framework to forecast human motions where each stage predicts

Table 2: Mean per joint position errors (unit: millimeters) of different ablated versions of our method for motion forecasting on the MoGaze dataset. Best results are in bold. Our method significantly outperforms the ablated versions, validating the effectiveness of each component used in our method.

Method	100 ms	200 ms	300 ms	400 ms	500 ms	600 ms	700 ms	800 ms	900 ms	1000 ms	Average
w/o *static*	13.8	26.3	39.7	54.3	70.2	87.2	105.3	124.1	143.4	162.6	77.3
w/o *dynamic*	13.8	26.2	39.6	54.1	69.9	86.9	105.0	123.9	143.2	162.4	77.1
w/o *static+dynamic*	13.9	26.6	40.0	54.5	70.5	87.8	106.0	124.9	144.3	163.9	77.8
w/o *head*	13.7	26.2	39.5	54.2	70.1	87.2	105.2	124.1	143.6	163.0	77.2
w/o *static+dynamic+head*	14.3	26.9	40.4	55.0	71.2	88.8	107.5	126.9	147.0	167.3	79.0
Ours	**13.2**	**25.6**	**38.6**	**52.9**	**68.7**	**85.7**	**103.9**	**122.7**	**142.0**	**161.3**	**76.1**

an initial guess for the next stage.

Time Horizons of Input and Output Sequences For experiments on the ADT and MoGaze datasets (30 Hz), we used 10 frames of data as input to forecast human poses in the future 30 frames (i.e. up to one second into the future), following the common evaluation settings for motion forecasting [30, 34].

Implementation Details We trained the baseline methods from scratch using their default parameters. For our motion forecasting network, we used the Adam optimiser with an initial learning rate of 0.01 and decayed the learning rate by 0.95 every epoch. A batch size of 32 was employed to train the motion forecasting network for 80 epochs. Our method was implemented using the PyTorch framework.

4.3 Motion Forecasting Results

Results on ADT Table 1 summarises the motion forecasting performances of different methods on individual actions (*ADT-work*, *ADT-decoration*, *ADT-meal*), and on all actions (*ADT-all*). The table shows the average MPJPE error (in millimeters) over all 30 frames as well as the prediction errors for different future time horizons: 100 ms, 200 ms, . . . , 1000 ms. As can be seen from the table, our method consistently outperforms the state-of-the-art methods on different actions (*work*, *decoration*, *meal*, or *all*). For *work*, *decoration*, and *meal* actions, our method achieves an average improvement of 8.5% (30.0 *vs.* 32.8), 8.9% (46.9 *vs.* 51.5), and 8.0% (38.9 *vs.* 42.3) respectively in terms of MPJPE error. For all actions, our method achieves an average improvement of 8.7% (37.7 *vs.* 41.3) over the state of the art. We also compared the prediction errors of different methods at future 100-1000 ms respectively and observed that our method consistently outperforms prior methods at all the future time intervals. We further performed a paired Wilcoxon signed-rank test to compare the performances of our method with the state of the art and the results validated that the differences between our method and the state of the art are statistically significant ($p < 0.01$). Figure 3 shows an example of the predicted body poses from different methods on a sample of the ADT dataset. On this sample, the user is going to squat down to touch an object on the ground. We can see that our method can accurately predict this future body motion while prior methods that only use historical body poses fail to predict this motion. See supplementary video for more prediction results.

Results on MoGaze The motion forecasting performances of different methods on individual actions (*MoGaze-pick*, *MoGaze-place*) and on all actions (*MoGaze-all*) are summarised in Table 1. The table shows the average MPJPE error (in millimeters) over all 30 frames as well as the prediction errors for different future time horizons: 100 ms, 200 ms, . . . , 1000 ms. We can see from the table that our method consistently outperforms the state-of-the-art methods on different actions (*pick*, *place*, and *all*). For *pick* and *place* actions, our method achieves an average improvement of 6.8% (72.7 *vs.* 78.0) and 7.5% (82.6 *vs.* 89.3) respectively in terms of MPJPE error. For all actions, our method achieves an average improvement of 7.2% (76.1 *vs.* 82.0) over the state of the art. We also compared our method with prior methods at different future time horizons and validated that our method can achieve superior performances at all the future time intervals. A paired Wilcoxon signed-rank test was further used to compare the performances of our method with the state of the art and the results validated that the differences between our method and the state of the art are statistically significant ($p < 0.01$).

Using Only Historical Body Poses Considering that prior methods only used historical body poses as input, for a more fair comparison, we retrained our method using only body poses and tested on the ADT and MoGaze datasets. Table 1 summarises the motion forecasting performances on individual actions as well as on all the actions. We can see that the ablated version of our method still outperforms prior methods on both the ADT and MoGaze datasets in terms of average performance as well as performances at different future time horizons. For *work*, *decoration*, *meal*, and *all* actions on the ADT dataset, our method using only body poses achieves an average improvement of 5.8% (30.9 *vs.* 32.8), 4.9% (49.0 *vs.* 51.5), 5.4% (40.0 *vs.* 42.3), and 5.3% (39.1 *vs.* 41.3). A paired Wilcoxon signed-rank test was further used to compare the performances of our method using only body poses with the state of the art and the results validated that the differences are statistically significant ($p < 0.01$). For *pick*, *place*, and *all* actions on the MoGaze dataset, our method using only body poses achieves an average improvement of 4.1% (74.8 *vs.* 78.0), 3.0% (86.6 *vs.* 89.3), and 3.7% (79.0 *vs.* 82.0). The differences between our method using only body poses and the state of the art are statistically significant (paired Wilcoxon signed-rank test, $p < 0.01$). The above results validate the overall superiority of our model architecture.

Time Costs and Model Size Table 4 shows the time costs and model size of different methods. We can see that our method is of medium size and is more efficient than prior methods in terms of test time per batch (10 ms), validating the usefulness of our method in real-time applications. The time costs were calculated on an NVIDIA Tesla V100 SXM2 32GB GPU with an Intel(R) Xeon(R) Platinum 8260 CPU @ 2.40GHz.

4.4 Ablation Study

Scene Objects and Head Orientation In addition to historical body poses, our method also used features from head orientation and egocentric scene objects. To evaluate the effectiveness of these inputs, we respectively removed *static objects*, *dynamic objects*, *static and dynamic*, *head orientation*, and *static, dynamic, and head*, and retrained the ablated methods. Table 2 shows the motion forecasting results of these ablated methods on the MoGaze dataset. We can see that our method consistently outperforms the ablated methods in terms of both average error and errors at different time horizons and the results are statistically significant (paired Wilcoxon signed-rank test, $p < 0.01$). The above results indicate that both scene objects and head orientation help improve our method's motion forecasting performance. Figure 4 shows an example of the predicted body poses from different ablated versions of our method on the MoGaze dataset. We can see that our method consistently outperforms the ablated versions at different future time horizons, validating the effectiveness of each component used in our method.

Scene Object Number In our method, we used the two dynamic objects and two static objects that are closest to the viewport centre as our input. We further tested different number of scene objects and indicated the motion forecasting performances on MoGaze in Table 3. The scene objects were selected according to their angular distances to the viewport centre (subsection 3.4). For simplicity, the number

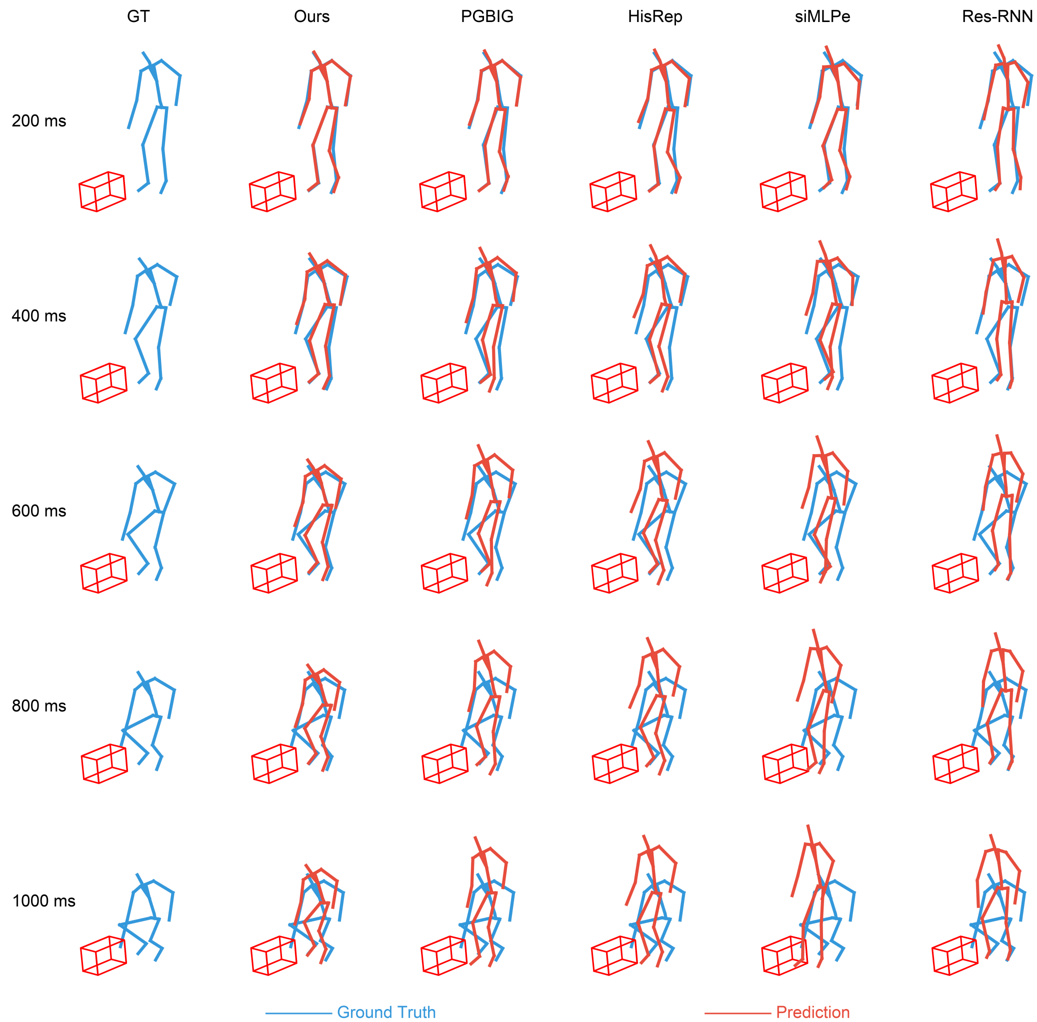

Fig. 3: Visualisation of the prediction results from different methods on a sample of the ADT dataset [38]. Our method can accurately predict the future body motion of *squat down to touch an object* while prior methods that only use historical body poses fail to predict this motion.

of static and dynamic objects were kept the same. We can see from Table 3 that using two dynamic and static objects achieves the best performances and the differences between different number of scene objects are statistically significant (paired Wilcoxon signed-rank test, $p < 0.01$). We also noticed that using more objects does not boost the motion forecasting performance, probably because users are more likely to interact with the objects that are closer to the viewport centre than the objects in the peripheral region [20, 22, 41] and thus adding information on peripheral objects cannot improve the performance.

Pose-object Graph In our pose-object graph, we respectively repeated the head and object features in the spatial domain to obtain five spatial nodes to enhance these features before fusing them with the pose features. To evaluate the effectiveness of this strategy, we further tested different number of spatial nodes and indicated the motion forecasting performances on MoGaze in Table 3. For simplicity, we used the same node number for head orientation, static objects, and dynamic objects. We can see from Table 3 that repeating the features to obtain five spatial nodes achieves the best performances and the differences between different number of spatial nodes are statistically significant (paired Wilcoxon signed-rank test, $p < 0.01$). We also noticed that repeating the features in the spatial domain (*spatial node* > 1 in Table 3) always performs better than no repeat (*spatial node* $= 1$), validating the effectiveness of the repeat strategy used in our pose-object fusion process.

Pose and Fuse Residual GCN In our method, we used pose residual GCNs to enhance the pose features and fuse residual GCNs to enhance the pose-object features. To evaluate the effectiveness of these two GCN blocks, we respectively removed these two blocks or changed the number of GCNs and indicated the motion forecasting performances on MoGaze in Table 3. We can see that using these two GCN blocks achieves significantly better performances than not using them (paired Wilcoxon signed-rank test, $p < 0.01$), validating the effectiveness of these two blocks. We also validated that using eight pose residual GCNs and 16 fuse residual GCNs achieves the best motion forecasting performance.

4.5 User Study

The results in Section 4.3 have quantitatively validated the effectiveness of our method. To further evaluate whether our method's improvements are significant in terms of human perception, we conducted a user study to compare our method with prior methods.

4.5.1 Stimuli

We randomly selected 20 motion forecasting samples from the ADT and MoGaze datasets (10 samples from each dataset) and used them as our stimuli. Each sample consisted of 30 frames of predictions (corresponding to future 1 second) and was visualised as a short video.

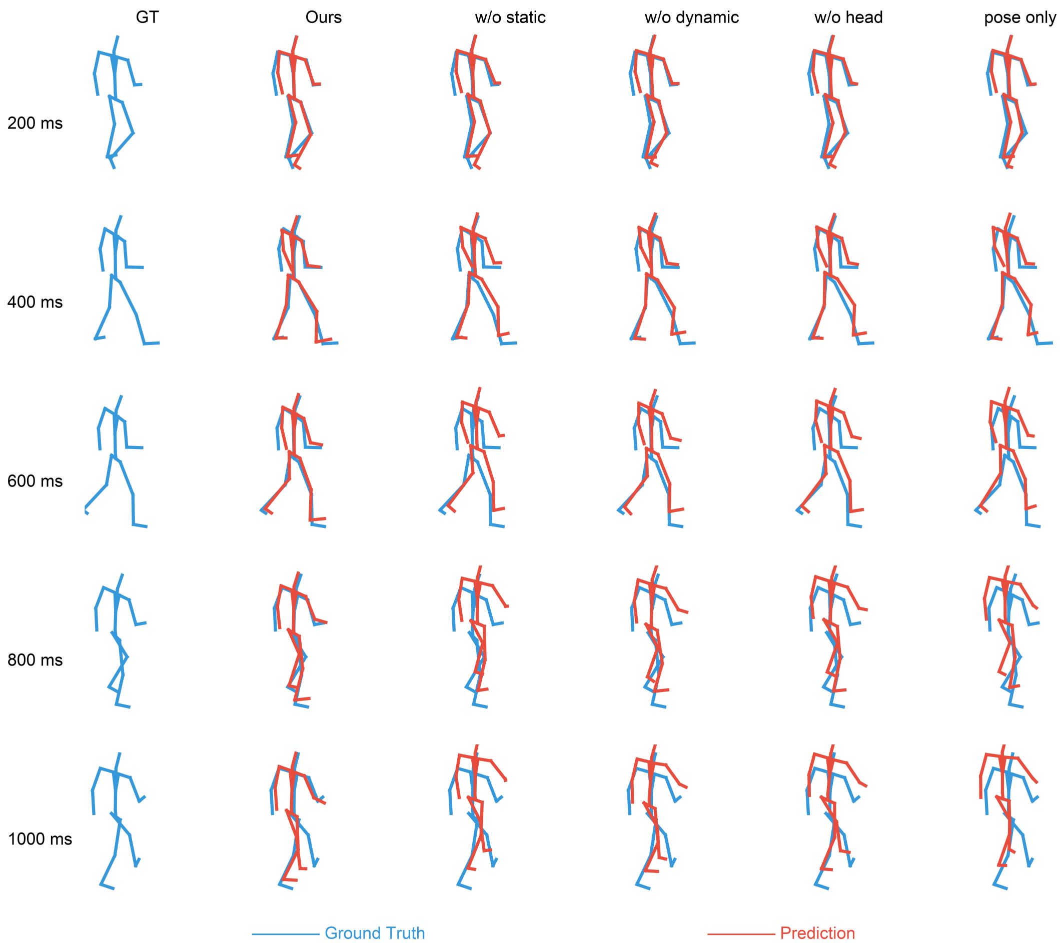

Fig. 4: Visualisation of different ablated versions of our method on a sample of the MoGaze dataset [26]. Our method consistently outperforms the ablated versions at different future time horizons.

4.5.2 Participants

We recruited 20 participants (10 males and 10 females, aged between 18 and 50 years, Mean=27.9, SD=6.8) to take part in our user study through university mailing lists and social networks. All of the participants reported normal or corrected-to-normal vision. The user study was approved by our university's ethical review board.

4.5.3 Procedure

We conducted our user study using a Google form. During the study, the ground truth future motions and the predictions of different methods were displayed to the participants in parallel using a layout that is similar to Figure 3. For simplicity, we only compared our method with PGBIG [30] and HisRep [34] since they are the strongest baselines from the results in Table 1. The names of different methods were hidden and the order of these methods were randomised. The visualisation videos of the ground truth and different methods were set to loop automatically, allowing participants to observe them with no time limit. Before the user study, participants were given detailed instructions (see the supplementary material) to get familiar with our experimental setting. During their observation, participants were required to rank different methods according to two criteria: *precision* and *realism*.

- *Precision*: check different methods to see whether they *align with the ground truth* and rank them based on your observation.
- *Realism*: check different methods to see whether they are *physically plausible* and rank them based on your observation.

We collected the responses from all the participants for further analysis.

4.5.4 Statistical Analysis

The medians, means and standard deviations (SDs) of different methods' rankings are shown in Table 5. We can see that our method outperforms the state of the art in terms of both precision (Median: 1.0 *vs.* 2.0, Mean: 1.2 *vs.* 2.3) and realism (Median: 1.0 *vs.* 2.0, Mean: 1.3 *vs.* 2.2). The results from a paired Wilcoxon signed-rank test validated that the differences between our method and the state of the art are statistically significant ($p < 0.01$). The above results demonstrate that our method achieves significantly better performances over prior methods in terms of human perception.

5 Discussion

Significance of Our Method Our method consistently outperforms prior methods in terms of average performance as well as performances at different time intervals (Table 1 and Figure 3), and the differences between our method and the state of the art are statistically significant (subsection 4.3, paired Wilcoxon signed-rank test, $p < 0.01$). The results from a user study further confirm that our improvements are significant in terms of human perception (subsection 4.5), implying that our method can be more effective in real applications.

Scene Objects for Motion Forecasting Our method combines past body poses with egocentric 3D object bounding boxes to forecast body motion in the future. Extensive experiments validate that the 3D bounding box information of scene objects can significantly improve the performances of motion forecasting (Table 2 and Figure 4). We also found that increasing the number of scene objects does not necessarily improve the motion forecasting performance (Table 3), revealing that users' body motions are mainly influenced by the objects that are

Table 3: Mean per joint position errors (unit: millimeters) of our method using different numbers of scene objects, spatial nodes, pose residual GCN, and fuse residual GCN for motion forecasting on the MoGaze dataset. Best results are in bold.

Method	100 ms	200 ms	300 ms	400 ms	500 ms	600 ms	700 ms	800 ms	900 ms	1000 ms	Average
object 0	13.9	26.6	40.0	54.5	70.5	87.8	106.0	124.9	144.3	163.9	77.8
object 1	13.5	26.0	39.3	54.0	69.9	87.1	105.1	124.1	143.5	162.8	77.1
object 2 (ours)	**13.2**	**25.6**	**38.6**	**52.9**	**68.7**	**85.7**	**103.9**	**122.7**	**142.0**	**161.3**	**76.1**
object 3	13.6	26.1	39.3	53.8	69.6	86.6	104.7	123.5	142.8	161.9	76.8
object 4	13.6	26.2	39.5	54.1	70.0	87.1	105.1	123.8	143.2	162.5	77.1
object 5	13.8	26.3	39.7	54.3	70.3	87.7	106.0	124.8	144.4	163.8	77.7
spatial node 1 (no repeat)	14.0	26.7	40.1	54.6	70.3	87.3	105.3	123.7	142.7	162.1	77.3
spatial node 5 (ours)	13.2	25.6	**38.6**	**52.9**	**68.7**	**85.7**	**103.9**	**122.7**	**142.0**	**161.3**	**76.1**
spatial node 10	13.3	25.9	39.2	53.8	69.8	87.0	105.2	124.1	143.5	162.9	77.0
spatial node 20	13.1	25.5	38.7	53.3	69.3	86.5	104.8	123.8	143.3	162.8	76.7
spatial node 30	**13.0**	**25.5**	38.8	53.4	69.3	86.8	105.1	124.2	143.6	162.8	76.8
pose residual GCN 0	13.4	25.8	39.1	53.6	69.4	86.5	104.6	123.3	142.7	161.9	76.6
pose residual GCN 8 (ours)	**13.2**	**25.6**	**38.6**	**52.9**	**68.7**	**85.7**	**103.9**	**122.7**	**142.0**	**161.3**	**76.1**
pose residual GCN 16	13.4	25.8	39.2	53.9	70.0	87.5	105.8	125.0	144.6	164.1	77.5
fuse residual GCN 0	16.1	31.0	46.0	61.5	78.0	95.7	114.7	134.5	155.3	176.6	85.1
fuse residual GCN 8	14.3	27.0	40.3	54.9	70.6	87.7	105.8	124.7	144.1	163.4	77.8
fuse residual GCN 16 (ours)	**13.2**	**25.6**	**38.6**	**52.9**	**68.7**	**85.7**	**103.9**	**122.7**	**142.0**	**161.3**	**76.1**
fuse residual GCN 32	13.3	25.8	39.2	53.8	69.8	87.1	105.6	124.8	144.7	164.4	77.4

Table 4: Time costs and model size of different methods.

Method	Training (Per Batch)	Test (Per Batch)	Model Size
Res-RNN [35]	37 ms	32 ms	3.41 M
siMLPe [12]	62 ms	36 ms	0.09 M
HisRep [34]	56 ms	37 ms	3.38 M
PGBIG [30]	97 ms	39 ms	1.93 M
Ours	59 ms	10 ms	2.23 M

Table 5: Statistical results of different methods' rankings in our user study. Best results are in bold. Rank 1 means best performance.

		Ours	*PGBIG* [30]	*HisRep* [34]
Precision	Median	**1.0**	2.0	3.0
	Mean	**1.2**	2.3	2.5
	SD	0.5	0.6	0.6
Realism	Median	**1.0**	2.0	2.0
	Mean	**1.3**	2.2	2.3
	SD	0.6	0.7	0.7

closer to the viewport centre. These results provide meaningful insights for developing future scene object-aware human motion forecasting methods.

Usability of Our Method Although our method requires additional information on the egocentric 3D object bounding boxes , such information is readily available in VR/AR systems [6, 14, 18, 39] and can also be easily accessed in real-world environments by applying existing 3D object bounding box estimation methods [36]. In addition, even using only historical body poses as input, our method still achieves significantly better performances over prior methods (Table 1), further validating the usability of our method in real applications.

Head Orientation vs. Eye Gaze We have tried to use eye gaze in our preliminary experiments and found that eye gaze performs worse than head orientation in terms of MPJPE error on ADT (38.6 vs. 37.7). This is probably because eye gaze is much more noisy than head orientation and thus degrades overall performance. Therefore, we opt to use head orientation in our architecture.

Limitations Despite all the advances that we have made, we identified several limitations that we plan to address in future work. First, to the best of our knowledge, the MoGaze and ADT datasets are the only public datasets that provide both full-body motion and information on 3D scene objects, thus unfortunately limiting the generalisability of our evaluation. In future work, we plan to assess our method for a broader range of activities and environments. In addition, our method is specifically designed for human-object interactions and may not work well for other situations such as human-human interactions. How to adapt our method to other situations remains to be explored. Finally, our method takes observed past body poses and scene objects as input while in real applications the input data may be incomplete due to tracking errors and may degrade the performance of our method. How to deal with incomplete observations is worthy of further study.

Future Work Besides overcoming the above limitations, many potential avenues of future work exist. First, it would be interesting to explore the effectiveness of other scene object-related information such as shape and colour for human motion forecasting. In addition, we are also looking forward to adding some physical constraints for the predicted human poses to make them more physically plausible. Furthermore, integrating our method into motion-related VR/AR applications is an interesting avenue of future work. Finally, adding prior knowledge on human intention during human-object interactions, e.g. the target object during a *pick* activity, to our pipeline may further boost the motion forecasting performance.

6 Conclusion

In this work we proposed a novel method for human motion forecasting during human-object interactions that first uses an encoder-residual GCN and multi-layer perceptrons to extract features from past body poses and egocentric 3D object bounding boxes respectively, fuses these features into a pose-object graph, and applies a residual-decoder GCN to forecast future motion. Through extensive experiments on two public datasets for motion forecasting at different time intervals we demonstrated that our method consistently outperformed several state-of-the-art methods by a large margin. We also validated that our predictions were more precise and more realistic than prior methods through a user study. As such, our work reveals the significant information content available in egocentric 3D object bounding boxes for human motion forecasting and informs future research on this promising research direction.

Acknowledgments

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2075 – 390740016. A. Bulling was funded by the European Research Council (ERC) under grant agreement 801708.

References

[1] S. Adebayo, S. McLoone, and J. C. Dessing. Hand-eye-object tracking for human intention inference. *IFAC-PapersOnLine*, 55(15):174–179, 2022. 2

[2] E. Aksan, M. Kaufmann, P. Cao, and O. Hilliges. A spatio-temporal transformer for 3d human motion prediction. In *Proceedings of the 2021 International Conference on 3D Vision*, pp. 565–574. IEEE, 2021. 2

[3] K. Bektaş, J. Strecker, S. Mayer, D. K. Garcia, J. Hermann, K. E. Jenss, Y. S. Antille, and M. Solèr. Gear: Gaze-enabled augmented reality for human activity recognition. In *Proceedings of the 2023 Symposium on Eye Tracking Research and Applications*, pp. 1–9, 2023. 2

[4] I. I. Butaslac, A. Luchetti, E. Parolin, Y. Fujimoto, M. Kanbara, M. De Cecco, and H. Kato. The feasibility of augmented reality as a support tool for motor rehabilitation. In *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*, pp. 165–173. Springer, 2020. 1

[5] Y.-H. Cho, D.-Y. Lee, and I.-K. Lee. Path prediction using lstm network for redirected walking. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 527–528. IEEE, 2018. 1

[6] A. Crivellaro, M. Rad, Y. Verdie, K. M. Yi, P. Fua, and V. Lepetit. Robust 3d object tracking from monocular images using stable parts. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1465–1479, 2017. 2, 9

[7] L. Dang, Y. Nie, C. Long, Q. Zhang, and G. Li. Msr-gcn: Multi-scale residual graph convolution networks for human motion prediction. In *Proceedings of the 2021 IEEE International Conference on Computer Vision*, pp. 11467–11476, 2021. 2

[8] B. David-John, C. E. Peacock, T. Zhang, T. S. Murdison, H. Benko, and T. R. Jonker. Towards gaze-based prediction of the intent to interact in virtual reality. In *Proceedings of the 2021 ACM Symposium on Eye Tracking Research and Applications*, pp. 1–7, 2021. 1, 2

[9] F. Dell'Agnola, N. Momeni, A. Arza, and D. Atienza. Cognitive workload monitoring in virtual reality based rescue missions with drones. In *Proceedings of the 2020 International Conference on Human-Computer Interaction*, pp. 397–409, 2020. 2

[10] K. J. Emery, M. Zannoli, J. Warren, L. Xiao, and S. S. Talathi. Openneeds: A dataset of gaze, head, hand, and scene signals during exploration in open-ended vr environments. In *Proceedings of the 2021 ACM Symposium on Eye Tracking Research and Applications*, pp. 1–7, 2021. 2

[11] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik. Recurrent network models for human dynamics. In *Proceedings of the 2015 IEEE International Conference on Computer Vision*, pp. 4346–4354, 2015. 2

[12] W. Guo, Y. Du, X. Shen, V. Lepetit, X. Alameda-Pineda, and F. Moreno-Noguer. Back to mlp: A simple baseline for human motion prediction. In *Proceedings of the 2023 IEEE Winter Conference on Applications of Computer Vision*, pp. 4809–4819, 2023. 2, 4, 5, 9

[13] J. Hadnett-Hunter, G. Nicolaou, E. O'Neill, and M. Proulx. The effect of task on visual attention in interactive virtual environments. *ACM Transactions on Applied Perception*, 16(3):1–17, 2019. 1, 2

[14] A. Hale and C. Leuze. Holoyolo: Understand the real world by running object detection on the hololens 2 and projecting the detection results in space. `https://devpost.com/software/holoyolo`. 9

[15] Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. J. Black, I. Laptev, and C. Schmid. Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11807–11816, 2019. 2

[16] Z. Hu. Gaze analysis and prediction in virtual reality. In *Proceedings of the 2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops*, pp. 543–544. IEEE, 2020. 2

[17] Z. Hu. [dc] eye fixation forecasting in task-oriented virtual reality. In *Proceedings of the 2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops*, pp. 707–708. IEEE, 2021. 2

[18] Z. Hu, A. Bulling, S. Li, and G. Wang. Fixationnet: forecasting eye fixations in task-oriented virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 27(5):2681–2690, 2021. 1, 2, 9

[19] Z. Hu, A. Bulling, S. Li, and G. Wang. Ehtask: Recognizing user tasks from eye and head movements in immersive virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, 2022. 2, 4

[20] Z. Hu, S. Li, C. Zhang, K. Yi, G. Wang, and D. Manocha. Dgaze: Cnn-based gaze prediction in dynamic scenes. *IEEE Transactions on Visualization and Computer Graphics*, 26(5):1902–1911, 2020. 2, 4, 7

[21] Z. Hu, J. Xu, S. Schmitt, and A. Bulling. Pose2gaze: Eye-body coordination during daily activities for gaze prediction from full-body poses. *IEEE Transactions on Visualization and Computer Graphics*, 2024. 2

[22] Z. Hu, C. Zhang, S. Li, G. Wang, and D. Manocha. Sgaze: a data-driven eye-head coordination model for realtime gaze prediction. *IEEE Transactions on Visualization and Computer Graphics*, 25(5):2002–2010, 2019. 2, 4, 7

[23] C. Jiao, Z. Hu, M. Bâce, and A. Bulling. Supreyes: Super resolution for eyes using implicit neural representation learning. In *Proceedings of the 2023 ACM Symposium on User Interface Software and Technology*, pp. 1–13, 2023. 2

[24] J. Kim, W. Kim, H. Oh, S. Lee, and S. Lee. A deep cybersickness predictor based on brain signal analysis for virtual reality contents. In *Proceedings of the 2019 IEEE International Conference on Computer Vision*, pp. 10580–10589, 2019. 2

[25] G. A. Koulieris, G. Drettakis, D. Cunningham, and K. Mania. Gaze prediction using machine learning for dynamic stereo manipulation in games. In *Proceedings of the 2016 IEEE Virtual Reality*, pp. 113–120. IEEE, 2016. 2

[26] P. Kratzer, S. Bihlmaier, N. B. Midlagajni, R. Prakash, M. Toussaint, and J. Mainprice. Mogaze: A dataset of full-body motions that includes workspace geometry and eye-gaze. *IEEE Robotics and Automation Letters*, 6(2):367–373, 2020. 2, 5, 8

[27] A. T. Le, P. Kratzer, S. Hagenmayer, M. Toussaint, and J. Mainprice. Hierarchical human-motion prediction and logic-geometric programming for minimal interference human-robot tasks. In *Proceedings of the 2021 IEEE International Conference on Robot and Human Interactive Communication*, pp. 7–14. IEEE, 2021. 2

[28] A. M. Lehrmann, P. V. Gehler, and S. Nowozin. Efficient nonlinear markov models for human motion. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1314–1321, 2014. 2

[29] C.-L. Li, M. P. Aivar, M. H. Tong, and M. M. Hayhoe. Memory shapes visual search strategies in large-scale environments. *Scientific Reports*, 8(1):1–11, 2018. 2

[30] T. Ma, Y. Nie, C. Long, Q. Zhang, and G. Li. Progressively generating better initial guesses towards next stages for high-quality human motion prediction. In *Proceedings of the 2022 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6437–6446, 2022. 2, 3, 4, 5, 6, 8, 9

[31] W. Mao, R. I. Hartley, M. Salzmann, et al. Contact-aware human motion forecasting. *Advances in Neural Information Processing Systems*, 35:7356–7367, 2022. 2

[32] W. Mao, M. Liu, and M. Salzmann. History repeats itself: Human motion prediction via motion attention. In *Proceedings of the 2020 European Conference on Computer Vision*, pp. 474–489. Springer, 2020. 3, 4

[33] W. Mao, M. Liu, M. Salzmann, and H. Li. Learning trajectory dependencies for human motion prediction. In *Proceedings of the 2019 IEEE International Conference on Computer Vision*, pp. 9489–9497, 2019. 3

[34] W. Mao, M. Liu, M. Salzmann, and H. Li. Multi-level motion attention for human motion prediction. *International Journal of Computer Vision*, 129(9):2513–2535, 2021. 2, 5, 6, 8, 9

[35] J. Martinez, M. J. Black, and J. Romero. On human motion prediction using recurrent neural networks. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2891–2900, 2017. 2, 5, 9

[36] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka. 3d bounding box estimation using deep learning and geometry. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7074–7082, 2017. 9

[37] T. Ohkawa, K. He, F. Sener, T. Hodan, L. Tran, and C. Keskin. Assemblyhands: Towards egocentric activity understanding via 3d hand pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12999–13008, 2023. 2

[38] X. Pan, N. Charron, Y. Yang, S. Peters, T. Whelan, C. Kong, O. Parkhi, R. Newcombe, and Y. C. Ren. Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In *Proceedings of the 2023 IEEE International Conference on Computer Vision*, pp. 20133–20143, 2023. 2, 4, 5, 7

[39] Y. Park, V. Lepetit, and W. Woo. Multiple 3d object tracking for augmented reality. In *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pp. 117–120. IEEE, 2008. 2, 9

[40] H. Razali and Y. Demiris. Using eye gaze to forecast human pose in everyday pick and place actions. In *Proceedings of the 2022 International Conference on Robotics and Automation*, pp. 8497–8503. IEEE, 2022. 2

[41] V. Sitzmann, A. Serrano, A. Pavel, M. Agrawala, D. Gutierrez, B. Masia, and G. Wetzstein. Saliency in vr: how do people explore virtual environments? *IEEE Transactions on Visualization and Computer Graphics*, 24(4):1633–1642, 2018. 2, 4, 7

[42] G. W. Taylor, G. E. Hinton, and S. Roweis. Modeling human motion using binary latent variables. *Advances in Neural Information Processing Systems*, 19, 2006. 2

[43] C. Tremmel, C. Herff, T. Sato, K. Rechowicz, Y. Yamani, and D. J. Krusienski. Estimating cognitive workload in an interactive virtual reality envi-

ronment using eeg. *Frontiers in Human Neuroscience*, 13:401, 2019. 2

[44] D. Valkov, P. Kockwelp, F. Daiber, and A. Krüger. Reach prediction using finger motion dynamics. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, pp. 1–8, 2023. 2

[45] J. Wang, A. Hertzmann, and D. J. Fleet. Gaussian process dynamical models. *Advances in Neural Information Processing Systems*, 18, 2005. 2

[46] Y. Zheng, Y. Yang, K. Mo, J. Li, T. Yu, Y. Liu, K. Liu, and L. J. Guibas. Gimo: Gaze-informed human motion prediction in context. In *Proceedings of the 2022 European Conference on Computer Vision*, 2022. 1, 2, 5