

# Remembering What Is Important: A Factorised Multi-Head Retrieval and Auxiliary Memory Stabilisation Scheme for Human Motion Prediction

Tharindu Fernando, *Member, IEEE*, Harshala Gammulle, *Member, IEEE*, Sridha Sridharan, *Life Senior Member, IEEE*, Simon Denman, *Member, IEEE*, and Clinton Fookes, *Senior Member, IEEE*.

**Abstract**—Human’s exhibit complex motions that vary depending on the task that they are performing, the interactions they engage in, as well as subject-specific preferences. Therefore, forecasting future poses based on the history of the previous motions is a challenging task. This paper presents an innovative auxiliary-memory-powered deep neural network framework for the improved modelling of historical knowledge. Specifically, we disentangle subject-specific, task-specific, and other auxiliary information from the observed pose sequences and utilise these factorised features to query the memory. A novel Multi-Head knowledge retrieval scheme leverages these factorised feature embeddings to perform multiple querying operations over the historical observations captured within the auxiliary memory. Moreover, our proposed dynamic masking strategy makes this feature disentanglement process dynamic. Two novel loss functions are introduced to encourage diversity within the auxiliary memory while ensuring the stability of the memory contents, such that it can locate and store salient information that can aid the long-term prediction of future motion, irrespective of data imbalances or the diversity of the input data distribution. With extensive experiments conducted on two public benchmarks, Human3.6M and CMU-Mocap, we demonstrate that these design choices collectively allow the proposed approach to outperform the current state-of-the-art methods by significant margins:  $> 17\%$  on the Human3.6M dataset and  $> 9\%$  on the CMU-Mocap dataset.

**Index Terms**—Auxiliary Memory, Feature Factorisation, Memory Stabilisation, Human Motion Prediction.

## I. INTRODUCTION

IN real-world day-to-day activities human exhibit complex and highly varied poses. For instance, considering the motion as a person walks, depending on objects that the person is carrying, and any interactions that take place, we may observe diverse variations between different human motion sequences [1]. Moreover, subject specific details such as limb lengths, and skeleton structure result in nuances which hamper attempts to predict future skeleton motion [1]. As such, there are global covariance factors across different scenes, camera setups, and tasks that the predictive algorithms should compensate for when predicting future motion. In addition, subject-specific local variations within the same task, scene, and camera setups should also be considered for accurate forecasting.

Existing state-of-the-art algorithms have adapted multi-scale joint pooling or multi-scale skeleton segmentation as a method

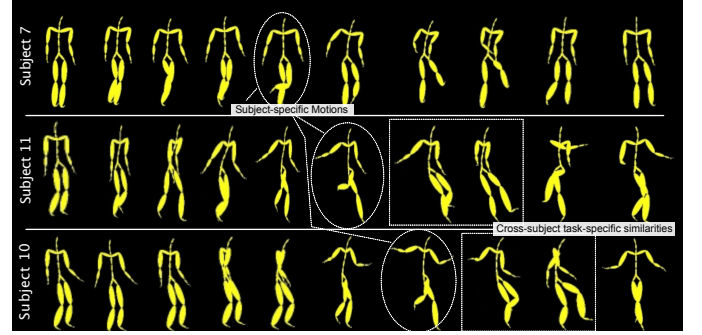


Fig. 1: Visual illustration of subject specificity and cross-subject task-specific similarities using samples for soccer action taken from the CMU-Mocap dataset. The subject specificity is illustrated with ellipses while the cross-subject task-specific similarities are shown with squares.

to compensate for global and local influencing factors [2], [3], [4]. We acknowledge the fact that such multi-scale modelling would help model distinct structural dependencies within human motion patterns. However, such a modelling approach fails to explicitly capture subject-specific, task-specific, and global factors that influence distinct motion patterns. Explicitly capturing these factors will help to capture semantically meaningful features, yielding better forecasting accuracy. Fig. 1 illustrates some examples from the CMU-Mocap dataset<sup>1</sup> that show subject specificity and cross-subject task-specific similarities that the learning framework can leverage; yet the factorisation of subject-specific, task-specific and other influential factors using the existing human motion modelling framework is an intricate task. In addition, the prevailing feed-forward deep learning pipeline that is leveraged within the state-of-the-art motion modelling literature does not allow for comparisons across samples that the model has previously seen during training, which is vital for the effective separation of salient task-specific and subject-specific attributes.

This paper proposes a novel deep neural network framework which effectively disentangles subject-specific, task-specific and other auxiliary features from human motion representations. In addition, we design an innovative auxiliary memory-powered multi-head retrieval scheme to efficiently incorporate these factorised details for knowledge retrieval. Specifically,

T. Fernando, H. Gammulle, S. Sridharan, S. Denman, and C. Fookes are with The Signal Processing, Artificial Intelligence and Vision Technologies (SAIVT), Queensland University of Technology, Australia.

<sup>1</sup><http://mocap.cs.cmu.edu/>

our auxiliary memory allows us to perform querying of the features from historically observed motion sequences in relation to different factorised features. For example, we can query the history in relation to the task that the subject of interest is performing, the observations that our model has seen which were performed by the same subject, or the observations related to the same background auxiliary features, and efficiently combine them for our predictive task at hand. Moreover, we make this feature disentanglement process dynamic where the feature masks that we used to factorise the features can dynamically change depending on the input, even after model training.

Several prior works have demonstrated the need for specialised losses to optimise the memory update procedure to work in situations with highly imbalanced data distributions. Relying on the auxiliary memory update procedure to be optimised from the classification or regression objective alone could lead the auxiliary memory to memorise only the information from the frequent classes. To alleviate this issue we propose a novel loss function which rewards diverse auxiliary memory content and penalises similar content. Moreover, smaller auxiliary memory sizes and/or high diversity in the input data distribution can lead to frequent updates and unstable auxiliary memory content. As such, we additionally propose an innovative loss function which penalises large memory updates that occur after the content of a particular auxiliary memory slot has been stabilised or consolidated, which we measure based on the time that content is initially written to the auxiliary memory and the frequency of the updates of that particular memory slot.

To the best of our knowledge, this is the first work to combine feature factorisation and auxiliary memory-powered knowledge retrieval. The main technical contributions of this paper, through which we introduce the proposed Factorised Multi-Head retrieval and Stabilisation based Auxiliary Memory (FMS-AM), can be summarised as follows.

- 1) We introduce a novel feature factorisation strategy to disentangle subject-specific, task-specific, and auxiliary attributes from human motion representations.
- 2) We present a new mask generation strategy that induces dynamicity in mask generation.
- 3) We propose an innovative multi-head access-based feature querying strategy to effectively retrieve knowledge embedded within an auxiliary memory.
- 4) Two novel loss functions are proposed to encourage diversity and stabilisation of the memory content while encouraging continual learning of important facts.

## II. RELATED WORK

In this section, we summarise the related works of this article which we categorise into works on Skeleton-based Human Motion Prediction (Sec. II-A), and literature on Auxiliary Memory Powered Neural Networks (Sec. II-B).

### A. Skeleton-based Human Motion Prediction

3D Skeleton-based human motion prediction is considered a fundamental research topic that benefits a myriad of application areas including intelligent surveillance, autonomous

driving, and human-robot interaction. A variety of human motion prediction methods have been proposed that range from traditional machine learning-based methods [5], [6], [7] to deep learning-based methods [8], [9], [10], [3], [11], [4]. Most recent success within this domain has been achieved using Graph Neural Networks (GNNs), with which researchers have exploited the relationships and constraints between different body components. Specifically, the Dynamic Multiscale Graph Neural Networks (DMGNN) [3] architecture models the human body in a multi-scale graph in which nodes are body components at various scales, and edges represent pairwise relations between those components. The multi-Scale Graph Computational Unit (MGCU) is the main processing component within the DMGNN architecture in which single-scale graphs extract features at their respective scales, and these extracted features are passed through different scale graphs to connect body components across two scales. Graph connections are initialized using predefined physical connections and are adaptively adjusted during model training. Then a graph-based GRU (G-GRU) is employed to predict future poses using the extracted representations. The MGCN [11] architecture follows a similar structure to DMGNN where the authors have utilised a Scale Interactional Module (SIM) to encode the human pose at multiple scales. Similarly, the MSR-GCN [4] architecture is motivated by the concept of coarse to fine-grained prediction generation. The descending GCN blocks are used to abstract the human pose at four levels of 12, 7, and 4 joints respectively, after which ascending GCN blocks reconstruct future poses at increasingly fine-grained scales. More recently, a multi-stage prediction framework named Spatial Dense Graph Convolutional Networks (S-DGCN) is proposed in [12]. In this progressively evolving architecture, the observed pose sequence and the initial guess of the future pose, which is predicted by the previous stage, are taken as the input. The authors show that this recursively updating approach leads to the progressive improvement of the initial guess predicted by the first stage.

In a different line of work, [13] proposed a Geometric Algebra-based Multi-view Interaction network (GA-MIN) leveraging Geometric Algebra tools to mitigate feature similarity issues when aggregating high-dimensional features using a deep and multi-stage GCN. Specifically, the authors use a Graph Spectrum Self-interaction module to discover the repeated motion within the input sequences, and use a Graph Spectrum Global-interaction module to extract informative motion representations within spectrum bands. The architecture of PK-GCN [14] is motivated by the observation that sequence interpolation is easier than extrapolation. As such two networks, the named InTerPolation learning Network (ITP-Network) and Final Prediction Network (FP-Network), are proposed. In contrast to an approach where we directly extrapolate the relationship between the observed sequence and the target, the ITP-Network learns to encode the input and a privileged sequence to interpolate the in-between frames in the predicted sequence. The FP-Network receives the encoded input but the privileged sequence is not visible to it. It uses a PK-Simulator that distills the privileged sequence based on the observed sequence. As such the FP-Network is able

to imitate the interpolation process. The Dynamic Pattern-based collaborative modeling network (DPnet) [15] considers preserving dynamic information of the joints. The authors show that global modeling of joint relationships could lead to the introduction of undesired trajectory constraints. To address this issue the authors propose a keyframe-enhanced module that augments the extracted temporal features by encoding the input into different-length sub-sequences. The dynamic patterns of different joints are discriminated using a dynamic pattern-guided feature extractor.

Considering the recent success of Transformer Networks in numerous sequence modeling tasks [16], [17], [18], they also have been used to model human motion sequences [19], [20]. The self-attention mechanism within the transformer has been adapted to compute the pair-wise joint relationships. However, the authors of [12] have shown that GCNs are more robust and efficient compared to transformers in modeling the pairwise relations of joints. As such, we adopt a GCN-based backbone to encode input motion sequences. However, in contrast to existing works in skeleton-based human motion prediction, we leverage a novel feature factorisation strategy to disentangle subject-specific, task-specific, and other auxiliary features from the encoded inputs. Furthermore, a novel auxiliary memory architecture is proposed to query subject-specific historical local patterns, as well as global task-specific patterns, that are embedded within the memory architecture.

### B. Auxiliary Memory Powered Neural Networks

Auxiliary Memory-Powered Neural Networks (AMNNs) [21], [22], [23], [24], [25], [26], [27], [28] are a recent and pivotal development within deep learning. They have shown tremendous success in automatically deriving long-term dependencies between input observations, and have been able to attain state-of-the-art results in various machine learning tasks, including, anomaly detection [23], [21], [24], interaction modelling [26], multimodal data fusion [22], [27], visual tracking [25], visual question answering [29], and human action recognition [30].

Specifically, an AMNN utilises explicit storage (memory) to store important facts and automatically retrieve relevant long-term dependencies when making a decision regarding a particular input, which is highly beneficial when extrapolating into the distant future. AMNNs fall under the category of stateful neural networks which maintain and temporarily evolve their states across the entire training/testing phase, in contrast to typical feed-forward and recurrent neural network architectures which only map the relationships within a particular input. This statefulness offers a greater utility for modeling relationships across different data elements in the dataset, enabling elevated levels of knowledge extraction.

There are three primary functions that facilitate this temporal evolution of the knowledge captured in the AMNN. A query function (also called an input controller), which is composed of trainable neural network layers, transforms the input embedding into a vector to query the memory. Using this query vector, the similarity between the content of each memory slot and the query is measured and relevant memory slots

for knowledge retrieval are identified. A composer function is used to transform the content retrieved from the identified memory slots into the memory output. The final task is to update the auxiliary memory content and propagate it into its next state. A memory update/write function receives the current memory output and it generates a vector to update the memory. Then the content of the slots which we leveraged in the memory read are updated using the generated memory update vector. It can be seen that the query function plays a pivotal role in identifying salient content in the memory which relates to the current input and can aid the task at hand. However, we observe that the current single-head access schemes used by such methods limit the identification of salient memory slots, as it focuses on the entire content of input and a particular memory slot. But in applications such as human motion synthesis, there are numerous task-specific and subject-specific factors that could be represented to various degrees within the embeddings. As such, a multi-head retrieval scheme is preferred in which varying levels of attention can be paid to these numerous influential factors. To the best of our knowledge, this is the first work to propose such a multi-head knowledge retrieval scheme in AMNNs. Furthermore, we propose two novel objective functions which encourage diversity in the memory content while also encouraging the memory content to be stable (discouraging frequent updates).

## III. METHODS

In this section we outline our proposed approach. We first introduce the encoder that we use to encode the input pose sequence (see Sec. III-A). Sec. III-B discuss the feature factorisation strategy that we implement to disentangle the encoded features. In Sec. III-C we present our pipeline for generating dynamic masks based on the encoded information in the input feature, and sections III-D and III-E present the proposed multi-head knowledge retrieval and auxiliary memory stabilisation procedures respectively, within the proposed auxiliary memory module. The pose sequence prediction process using our decoder is presented in Sec. III-F. Finally, the implementation details of the framework are discussed in Sec. III-G.

### A. Multi-scale GCN Encoder

Input to our encoder is a sequence of human poses, and our encoder transforms this to a deep representation by hierarchically encoding it at multiple scales. In this subsection, we explain this hierarchical encoding process.

Formally, let  $X_{1:T_{obs}} = [x_1, x_2, \dots, x_{T_{obs}}]$  consist of  $T_{obs}$  consecutive human pose observations, where  $x_i \in \mathbb{R}^K$  and  $K$  is the data dimension that we utilise to describe each pose observation.  $x_i \in \mathbb{R}^K \in \mathbb{R}^{J \times D}$  represents a single human pose which is composed of  $J$  joints, and each joint is represented in  $D$ -dimensional space. In the datasets used in this work pose is observed in 3 dimensions, thus  $D = 3$ . Our objective is to anticipate future poses for the duration  $T_{obs} + 1$  to  $T_{obs} + T$ . We denote the predicted pose sequence as  $\hat{Y}_{T_{obs}+1:T_{obs}+T} = [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_T]$ , and the ground truth observation as  $Y_{T_{obs}+1:T_{obs}+T} = [y_1, y_2, \dots, y_T]$ .

Inspired by the recent success of graph neural networks to model the spatial structure of the poses [4], [3] and the proven ability of the graph convolution operation to retrieve spatial and structural dependencies between human joints, we utilise a Graph Convolution Network (GCN) as our backbone to extract features from the input pose sequence. Following [4], [3] we first replicate the last pose observation,  $x_{T_{obs}}$ ,  $T$  times making the input sequence of length  $T_{obs} + T$ . Similar to [4], [3] we represent pose as a fully connected graph with  $K$  nodes and the adjacency matrix,  $A \in \mathbb{R}^{K \times K}$ , which is learned during the training process represents the strength of the dependencies between pairs of joints.

Our GCN-based pose encoder is composed of  $L$  graph stacked convolution layers. At each level,  $l \in [1, 2, \dots, L]$ , the output,  $H^{l+1}$ , can be defined as,

$$H^{l+1} = f_{GCN}(A^l, H^l, W^l), \quad (1)$$

where  $F_{GCN}$  is the activation function and  $W^l \in \mathbb{R}^{F^l \times F^{l+1}}$  are the trainable parameters of the graph convolution layer.  $H^{l+1} \in \mathbb{R}^{K \times F^{l+1}}$  is the output which is passed to the next graph convolution layer.  $F^l$  denotes the embedding dimension of the layer  $l$ .

Motivated by the success of [4], [3] in leveraging multi-scale GCNs for capturing the hierarchical spatial and structural relationships of human pose using multi-scale representations, we also employ a series of GCNs to abstract the human pose. Our descending GCN blocks downsample the resolution of the pose sequence by pooling adjacent joints. For instance, if the input pose is represented with 22 joints (i.e.  $J = 22$ ) in three dimensions (i.e.  $D = 3$ ), then the input feature space to our first downsampling block,  $DN_0$  is of shape  $K_0 \times F$  with  $K_0 = 22 \times 3 = 66$ . This input is down-sampled and the input to the second downsampling block,  $DN_1$ , is of shape  $K_1 = 12 \times 3 = 36$ . Similarly, the 3<sup>rd</sup> and 4<sup>th</sup> downsampling blocks,  $DN_2, DN_3$ , set  $K_2 = 7 \times 3 = 21$  and  $K_3 = 4 \times 3 = 12$ , respectively. Note that for all the downsampling blocks we use the same embedding dimension  $F$ . We note that this encoding process is identical to the encoder of [4].

### B. Factorised Embeddings

This subsection illustrates how we disentangle the embeddings generated by the encoder introduced in Sec. III-A. Specifically, we leverage masking operations to disentangle subject-specific, task-specific, and other auxiliary features from the output of our encoder. Formally, let  $z_t^{DN_3} \in \mathbb{R}^{K_3 \times F}$  denote the output of the final downsampling block,  $DN_3$ , for the observed pose input at time instance  $t$ . Then,

$$\begin{aligned} z_t^{sub} &= z_t^{DN_3} \otimes m_t^{sub}, \\ z_t^{task} &= z_t^{DN_3} \otimes m_t^{task}, \\ z_t^{aux} &= z_t^{DN_3} \otimes m_t^{aux}, \end{aligned} \quad (2)$$

can be used to split the respective features across subject, task, and auxiliary segments. Here  $\otimes$  denotes element-wise multiplication by each mask,  $m_i \in \mathbb{R}^{K_3 \times F}$ . It should be noted that the masking operation only occurs in the embedding dimension,  $F$ , and we retain all the elements in dimension  $K_3$ .

Details regarding the mask generation process are presented in Sec. III-C.

To ensure that the features are properly disentangled, and the segregated embeddings capture the intended attributes for the specified segments (i.e. subject, task, etc.) and only for that specified segment, we embed additional classification objectives alongside the primary future pose sequence prediction task.

Specifically, each of the supplementary classification heads first concatenates the relevant embeddings for the entire observed sequence such that,

$$\begin{aligned} Z^{sub} &= [z_1^{sub} \oplus z_2^{sub} \oplus \dots \oplus z_t^{sub}], \\ Z^{task} &= [z_1^{task} \oplus z_2^{task} \oplus \dots \oplus z_t^{task}], \end{aligned} \quad (3)$$

where  $\oplus$  denotes column-wise concatenation. The resultant feature vectors,  $Z^{sub}$  and  $Z^{task}$ , are passed through a global average pooling operation,  $f_{GAP}$ , which transforms the input feature vectors to  $\hat{z}^{sub}$  and  $\hat{z}^{task}$  respectively, where  $\hat{z}^{sub} \in \mathbb{R}^{K_3 \times F}$  and  $\hat{z}^{task} \in \mathbb{R}^{K_3 \times F}$ . Then we pass the resultant embeddings,  $\hat{z}^{sub}$  and  $\hat{z}^{task}$ , through the respective classifiers such that,

$$\begin{aligned} \hat{y}^{sub} &= f_{SUB}(\hat{z}^{sub}), \\ \hat{y}^{task} &= f_{TASK}(\hat{z}^{task}), \end{aligned} \quad (4)$$

where  $f_{SUB}$  and  $f_{TASK}$  denote subject and task classification sub-networks, respectively. Note that the evaluation protocol for skeleton-based motion synthesis in most of the popular benchmarks is leave-one-subject-out. Therefore, direct classification of the subject identity is not appropriate as the model is observing completely unseen subjects during the testing phase. To alleviate this issue we construct the subject classification as a contrastive learning task where two arbitrarily sampled pose sequences, which could be of the same subject or different subjects, are presented to the model and using the two  $\hat{z}^{sub}$  embeddings extracted for the two inputs the  $f_{SUB}$  network classifies whether they belong to the same subject or not. We use contrastive loss [31] to optimise this task. This contrastive learning helps the model to identify salient characteristics embedded within the input pose sequences that are subject-specific which will, in turn, help the motion prediction. Furthermore, we can use this same pipeline for both the training and testing phases. For task classification, noting that we have the same set of tasks in the training and testing sets, we use categorical cross-entropy loss.

### C. Dynamic Mask Generation

The next task is to generate the masks such that subject, task, and auxiliary features can be disentangled. A naive way to generate the masks is to use a fixed mask such that a certain fixed region (of length  $l_{sub}$ ) is completely dedicated to a specific subset of features, and the rest of the elements in  $z_t^{DN_3}$  are completely masked out. For instance, if we select the first  $l_{sub}$  elements for subject-specific features, the next  $l_{task}$  elements for task-specific features, and the rest of the elements to carry auxiliary information, then the 3 masks can be visualised as in Fig. 2.

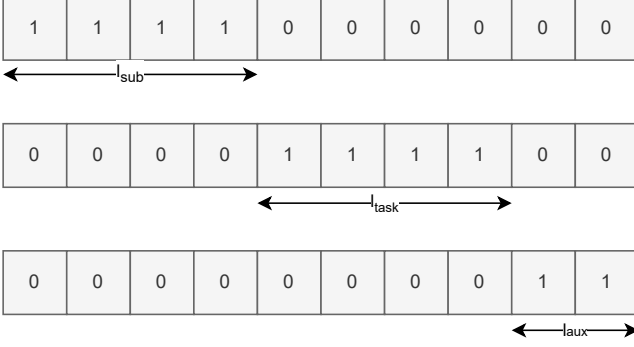


Fig. 2: Illustration of fixed masks where we pre-define the regions for subject-specific, task-specific, and auxiliary features.

We see several drawbacks to this approach. First, it assumes that there exist hard/rigid boundaries between subject, task, and auxiliary segments in the feature vector and those elements are either fully active or not; i.e. it does not allow partial activation of the mask. Second, it makes the feature disentanglement process static after the model training process, and doesn't allow the feature selection to dynamically change based on the embedded information. To resolve these drawbacks we propose to dynamically generate masks based on the encoded information in  $z_t^{DN_3}$ .

Specifically, we utilise a set of hard-coded masks (i.e.  $\hat{m}_t^{sub}$ ,  $\hat{m}_t^{task}$  and  $\hat{m}_t^{aux}$ ) based on the proportion in the vector  $z_t^{DN_3}$  that we desire the subject, task and auxiliary features to occupy. Then, using a neural network that is parameterised by the function  $f_{MASK}$  we generate residuals to augment the fixed masks,  $\hat{m}_t^{sub}$ ,  $\hat{m}_t^{task}$  and  $\hat{m}_t^{aux}$ . Fig. 3 visually illustrates this mask generation process.

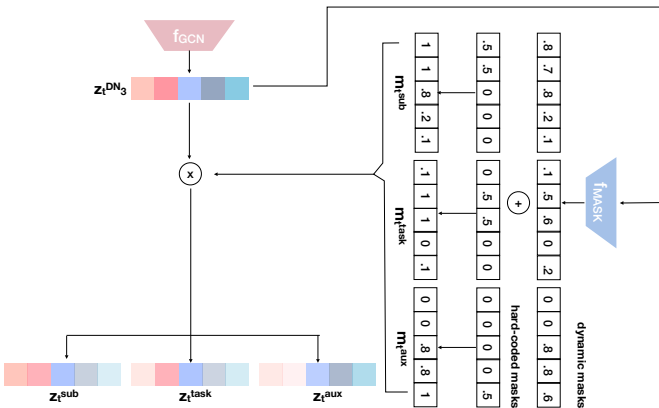


Fig. 3: Dynamic Mask Generation and Feature Factorisation: Using the feature vector,  $z_t^{DN_3}$ , a neural network generates residuals to augment the fixed mask. These masks are then utilised to factorize  $z_t^{DN_3}$ .

Formally,  $f_{MASK}$  outputs 3 mask vectors such that,

$$(\tilde{m}_t^{sub}, \tilde{m}_t^{task}, \tilde{m}_t^{aux}) = f_{MASK}(z_t^{DN_3}), \quad (5)$$

where  $\tilde{m}_t^{sub}$ ,  $\tilde{m}_t^{task}$  and  $\tilde{m}_t^{aux}$  each  $\in \mathbb{R}^{K_3 \times F}$  and represent the residual masks for the subject, task and auxiliary regions respectively. Then, the augmented mask can be generated using

$$\begin{aligned} m_t^{sub} &= \hat{m}_t^{sub} + \tilde{m}_t^{sub}, \\ m_t^{task} &= \hat{m}_t^{task} + \tilde{m}_t^{task}, \\ m_t^{aux} &= \hat{m}_t^{aux} + \tilde{m}_t^{aux}. \end{aligned} \quad (6)$$

This can be seen as injecting plasticity [24], [32] into a fixed set of hard-coded masks. Different initialisation functions can be used to initialise the fixed masks and for simplicity we used a uniform function such that all the elements within the particular segment (i.e.  $l_{sub}$ ,  $l_{task}$ ,  $l_{aux}$ ) are initialised to a value of 0.5 and  $l_{sub} = l_{task} = l_{aux} = F/3$ . Furthermore, the generated masks are normalised using the Gumbell softmax operation which is a differentiable relaxed one-hot vector-like operation. The temperature parameter,  $\tau$ , within the Gumbell softmax controls the sparsity of the resultant mask, with lower temperature values increasing output sparsity. We experimentally choose the value for  $\tau$  which allows us to balance the fixed and plastic portions of the generated masks.

We compare this mask generation process with the splitting network proposed in our prior work [33]. In [33] we generated the entire mask while in the proposed work we generate only the residuals for the fixed mask. We believe predicting the entire mask using Gumbell softmax operation is too restrictive as it can lead to most values in the predicted mask being zero. We experimentally compare the proposed architecture and the masking process of [33] where we demonstrate this drawback. We refer readers to the ablation evaluations in Sec. IV-D details. Furthermore, we note that to the best of our knowledge, this is the first work to embed plastic/dynamic mask generation methods within stateful, continual learning neural networks.

#### D. Auxiliary Memory and Multi-Head Retrieval

The main goal of our feature factorisation process in Sec. III-B is to effectively utilise this to augment the knowledge retrieval process in the auxiliary memory. As illustrated in Sec. II-B, a typical memory architecture is composed of a memory stack with  $s$  slots, and  $M_{\lambda-1}$  denotes the state of the memory at time  $\lambda - 1$ . Note that this time is measured with respect to the execution of the memory module (i.e. training/testing iteration) and not with respect to the time in input sequences (i.e.  $t$ ).

In a traditional memory retrieval operation we first pass the entire input to the memory,  $z_t^{DN_3}$ , through a query function,  $f_{QUERY}$ , to generate a query vector,  $q_\lambda^{DN_3}$ , to query the memory such that

$$q_\lambda^{DN_3} = f_{QUERY}(z_t^{DN_3}). \quad (7)$$

Then we retrieve memory slots that contain information related to our query using

$$\beta_\lambda^{DN_3} = \text{softmax}([q_\lambda^{DN_3}]^\top M_{\lambda-1}), \quad (8)$$

and the memory output,  $\mu_\lambda$ , at time instance  $\lambda$  is computed using,

$$\mu_\lambda = [\beta_\lambda^{DN_3}]^\top M_{\lambda-1}. \quad (9)$$

However, we observe several limitations of the direct application of a single-head access scheme for memory knowledge retrieval in motion synthesis. In particular, when there exist numerous task-specific and subject-specific variations embedded in the same feature vector this could negatively impact the softmax score-based identification of relevant memory slots. For instance, if the memory contains information related to the same subject in the query but the tasks are different, there is a possibility that such memory slots will not be identified as relevant to the current query due to the mismatch of the task-specific features. Similarly, information for similar tasks but with different subjects may be ignored as the query operation is considering the entire query vector. This issue can be overcome using the proposed feature factorisation strategy where we can generate multiple queries using the individual factorised attributes. Specifically, we use  $z_t^{sub}$  and  $z_t^{task}$  in addition to  $Z_t^{DN_3}$  as inputs to the memory module, and generate two additional query vectors such that,

$$\begin{aligned} q_\lambda^{sub} &= f_{QUERY,SUB}(z_t^{sub}), \\ q_\lambda^{task} &= f_{QUERY,TASK}(z_t^{task}), \end{aligned} \quad (10)$$

which are leveraged to retrieve memory slots using

$$\begin{aligned} \beta_\lambda^{sub} &= \text{softmax}([q_t^{sub}]^\top [M_{\lambda-1}][m_t^{sub} \odot e^s]), \\ \beta_\lambda^{task} &= \text{softmax}([q_t^{task}]^\top [M_{\lambda-1}][m_t^{task} \odot e^s]), \end{aligned} \quad (11)$$

where  $e^s$  is a matrix of ones, and  $\odot$  denotes the outer product which duplicates its left vector  $s$  times to form a matrix. Then the knowledge retrieved from memory can be defined by,

$$\begin{aligned} \mu_\lambda^{sub} &= [\beta_t^{sub}]^\top [M_{\lambda-1}][m_t^{sub} \odot e^s], \\ \mu_\lambda^{task} &= [\beta_t^{task}]^\top [M_{\lambda-1}][m_t^{task} \odot e^s]. \end{aligned} \quad (12)$$

Then, we aggregate the information retrieved using the overall, subject-specific, and task-specific queries and concatenate them across the final dimension,

$$\dot{\mu}_\lambda = [\mu_\lambda \oplus \mu_\lambda^{sub} \oplus \mu_\lambda^{task}], \quad (13)$$

where  $\dot{\mu}_\lambda \in \mathbb{R}^{K_3 \times 3F}$ . We also concatenate  $\dot{\mu}_\lambda$  output with  $z_t^{DN_3}$  such that  $\tilde{z}_t = [z_t^{DN_3} \oplus \dot{\mu}_\lambda]$  and  $\tilde{z}_t \in \mathbb{R}^{K_3 \times 4F}$ . We apply average pooling across the final dimension of  $\tilde{z}_t$  and the resultant feature vector,  $\tilde{z}_t \in \mathbb{R}^{K_3 \times F}$ . This augmented feature vector is then used to predict future motion. Fig. 4 visually illustrates the subject-specific feature retrieval process using the proposed multi-head retrieval scheme. Note that an identical process is leveraged to retrieve task-specific features from the auxiliary memory.

The architecture of our decoder module which predicts future motion is presented in Sec. III-F, and in Sec. III-E we outline the process of memory update which allows our model to perform continual learning.

#### E. Memory Update and Auxiliary Memory Stabilisation Losses

Once informative content is retrieved from the auxiliary memory the next task is to update the memory module to ensure the continual evolution of the knowledge embedded in our memory. For this task the typical process is to leverage the

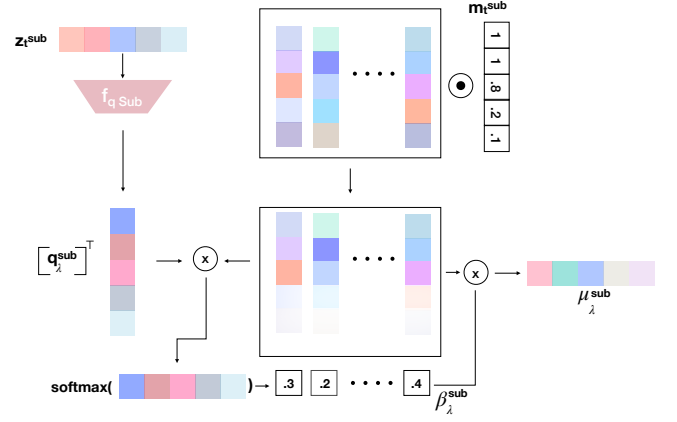


Fig. 4: Illustration of Subject-Specific Feature Retrieval Using the Proposed Multi-Head Retrieval Scheme): Using the augmented mask,  $m_t^{sub}$ , which we generate for disengagement of the subject-specific features we first segment the relevant subject-specific memory content within each of the memory slots. Then using the factorised subject-specific feature vector,  $z_t^{sub}$ , we generate a subject-specific query vector,  $q_\lambda^{sub}$ , and we attend to the content of each of the slots and quantify the similarity between the content and  $q_\lambda^{sub}$  which is captured by  $\beta_\lambda^{sub}$ . Finally, the subject-specific memory output,  $\mu_\lambda^{sub}$ , related to the subject-specific query vector is generated as per Eq. 12.

memory output,  $\dot{\mu}_\lambda$ , and pass it through a non-linear function,  $f_{WRITE}$  to generate a vector to update the memory such that,

$$\check{\mu}_\lambda = f_{WRITE}(\dot{\mu}_\lambda), \quad (14)$$

where  $\check{\mu}_\lambda \in \mathbb{R}^{K_3 \times F}$ . Then the subsequent state of the memory can be generated using,

$$M_\lambda = M_{\lambda-1}[I - \beta_\lambda \odot e^F]^\top + [\check{\mu}_\lambda \odot e^s][\beta_\lambda \odot e^F]^\top, \quad (15)$$

where  $I$  is a matrix of ones,  $e^s \in \mathbb{R}^s$ ,  $e^F \in \mathbb{R}^F$  are vectors of ones,  $\odot$  denotes the outer product which duplicates its left vector  $s$  or  $F$  times to form a matrix and,

$$\beta_\lambda = \beta_\lambda^{sub} + \beta_\lambda^{task} + \beta_\lambda^{DN_3}. \quad (16)$$

However, our prior investigations [34], [23], [24] revealed that reliance on the downstream (i.e. classification, regression) objective alone could lead to sub-optimal memory states. For instance, imbalanced or poorly curated datasets could lead the memory to memorize the most frequently observed types of data, ignoring the less frequent classes as these contribute less to the overall loss. Moreover, if the diversity of the dataset is high, or if the embedding size or the number of memory slots is small, this could lead to frequent memory updates making the knowledge stored in the memory highly volatile and unstable making the discovery of long-term dependencies infeasible. To this end, we propose two novel architectural innovations. First, we incorporate an additional neural network-based predictor parameterized by the function  $f_\beta$  which predicts the memory slots that need updating where,

$$\check{\beta}_\lambda = f_\beta(\check{m}_\lambda, \beta_\lambda). \quad (17)$$



Second, we propose two loss functions that directly operate on the memory content and encourage diversity and stability. Specifically, our diversity loss,  $L_{div}$ , will iteratively estimate the similarity between a slot's content and the rest of the content in memory. Formally, our diversity loss can be defined as,

$$L_{div} = \frac{1}{s(s-1)} \sum_{i=1}^s \sum_{\substack{j=1 \\ i < j}}^s f_{COS}(M_{\lambda-1}^i, M_{\lambda-1}^j), \quad (18)$$

where  $f_{COS}$  is the cosine similarity loss and  $M_{\lambda-1}^i$  denotes the  $i^{th}$  memory slot in the memory.

Next, we introduce  $L_{cons}$ , which helps consolidate memory content and penalises unstable memory updates. Formally, let  $\check{\beta}_{\lambda}^i$  denote the  $i^{th}$  element (i.e.  $i^{th}$  memory slot) in the output of Eq. 17 at time instance  $\lambda$ , let  $w$  define the window size, then we generate  $\frac{\lambda}{w}$  windows to inspect the change in memory updates for the period from time instance 1 to  $\lambda$ . The change in memory updates between two consecutive time intervals can be evaluated as,

$$\Delta_j^i = |\log \beta_j^i - \log \beta_{j-1}^i|, \quad (19)$$

where,

$$\Delta_c^i = \begin{cases} \frac{1}{|w|} \sum_{j=0}^w \Delta_j^i, & \text{if } \lambda - cw > j > \lambda - 2cw \\ 0, & \text{otherwise} \end{cases} \quad (20)$$

Now we define the memory consolidation loss,  $L_{cons}$ , as,

$$L_{cons} = \frac{1}{|s|} \sum_{i \in s} \sum_{c \in \frac{\lambda}{w}} \Delta_c^i 2^c. \quad (21)$$

This formulation of the loss penalises large memory updates that occur between large temporal intervals. Using the windowing operation in Eq. 19, we obtain the average amount of the change of the slot content within the window and using Eq. 21, and penalise large changes (due to the log scale) that occur once the content has been initially written (due to the exponential scale of  $c$ ). Using the window size we can control the period that we allow the slot content to stabilise and after which we exponentially penalise the content updates. As such, this loss does not negatively impact the continual learning ability of the memory.

We highlight the use of  $\check{\beta}_{\lambda}$  for the loss calculations instead of using  $M_{\lambda-1}$  directly. Direct use of the memory content would require the individual memory states at all time instances to be stored, leading to inefficiencies. However,  $\check{\beta}_{\lambda}$  provides a snapshot of how individual memory slots have been updated, and as such, we utilise this vector.

#### F. Multi-Scale GCN Decoder

Mirroring the downsampling blocks in our encoder, our decoder is equipped with upsampling GCN modules that gradually increase the resolution of the predicted pose representations. Specifically, the first upsampling block,  $AN_3$ , receives  $\check{z}_t \in \mathbb{R}^{K_3 \times F}$  and decodes it to a feature vector  $\check{z}_t^{AN_2} \in \mathbb{R}^{K_2 \times F}$ . Similar to our encoder, our decoder has four upsampling blocks,  $AN_3, AN_2, AN_1$ , and  $AN_0$ . As in

[4] four end-GCNs,  $E_0, E_1, E_2$ , and  $E_3$ , each with two graph convolution layers, are used to generate future pose sequences at four different scales. However, in contrast to [4] where respective downsampled (i.e.  $\check{z}_t^{DN_0}, \check{z}_t^{DN_1}, \check{z}_t^{DN_2}, \check{z}_t^{DN_3}$ ) and upsampled (i.e.  $\check{z}_t^{AN_0}, \check{z}_t^{AN_1}, \check{z}_t^{AN_2}, \check{z}_t^{AN_3}$ ) feature vectors in individual scales are concatenated and passed to the respective end GCNs, we only pass the upsampled features.

We use  $L_2$  loss between the ground truth and predicted pose sequences to train the pose prediction head of our framework. We denote this loss as  $L_{pose}$ .

The overall loss function of our framework then becomes

$$L^* = \theta_{pose} L_{pose} + \theta_{div} L_{div} + \theta_{cons} L_{cons} + \theta_{sub} L_{sub} + \theta_{task} L_{task}, \quad (22)$$

where  $L_{sub}$  is the contrastive loss for the subject identification task and  $L_{task}$  is the categorical cross-entropy loss for task identification.  $\theta_{pose}, \theta_{div}, \theta_{cons}, \theta_{sub}$  and  $\theta_{task}$  are loss weights that control the contributions from individual losses.

#### G. Implementation Details

Implementation of this framework is completed using PyTorch. The Adam [35] optimiser with an initial learning rate of  $2e^{-4}$  is used for optimisation. The learning rate is decreased by 0.98 every two epochs. The model is trained for 100 epochs on an NVIDIA A100 GPU. The embedding size,  $F$ , was experimentally chosen and was set to 300. Similarly, hyperparameters,  $w, \theta_{pose}, \theta_{div}, \theta_{cons}, \theta_{sub}, \theta_{task}$  were experimentally chosen and were set to 15, 0.4, 0.15, 0.15, 0.15, and 0.15 respectively.

### IV. EXPERIMENTS

In this section, we report the results of experiments that we conducted to evaluate and compare the efficiency of the proposed skeleton-based human motion prediction model, FMS-AM, with respect to existing state-of-the-art methods. We first introduce the details of the two datasets that we used for our evaluations (Sec. IV-A), then present the evaluation metrics that we use to measure the model performance (Sec. IV-B). The main experimental results where we compare our proposed method with existing state-of-the-art approaches are presented in Sec. IV-C. Ablation evaluations that were conducted to demonstrate the efficacy of the proposed feature disentanglement strategy, the multi-head retrieval strategy, the novel stabilisation losses, and the dynamic mask generation process are presented in Sec. IV-D. In Sec. IV-E we discuss the time complexity of our FMS-AM model.

#### A. Datasets

For our evaluations, following the state-of-the-art methods we use two popular motion capture benchmark datasets, namely, the Human3.6M (H3.6M) and CMU Motion Capture (CMU-Mocap) datasets. Details of these datasets are provided in the following subsections.

1) *H3.6M dataset* [1]: The H3.6M dataset consists of motions that performed by 11 professional actors, 5 female, and 6 male. This is a challenging dataset with 15 different action categories, including, Taking Photos, Waiting, Giving Directions, Walking Pair, Phone Talk, Sitting on the floor, Smoking, Sitting on a chair, etc. Similar to prior works [2], [3], [4] we use the data of seven subjects, S1, S5, S6, S7, S8, S9, and S11. Following [2], [3], [4] we use the data from S5 for testing, S11 for validation, and the rest of the subjects for model training. 22 body joints from the original 32 joints are chosen to represent the body pose and the data is mapped to a 3D joint coordinate space. We downsample all pose sequences by a factor of two along the temporal axis.

2) *CMU-Mocap dataset*: The CMU-Mocap dataset <sup>2</sup> has 5 abstract action classes including ‘human interaction’, ‘interaction with environment’, ‘locomotion’, ‘physical activities & sports’, and ‘situations & scenarios’. To maintain consistency with the H3.6M dataset we choose the following 8 detailed action categories from the dataset: basketball, basketball signal, directing traffic, jumping, running, soccer, walking, and washing a window. Similar to the H3.6M dataset 22 body joints from the original 38 joints were filtered and the data is mapped to a 3D joint coordinate space.

### B. Evaluation Protocol

Mean Per Joint Position Error (MPJPE) has been widely used as the evaluation metric in numerous recent works [4], [12], [15] due to its ability to directly compare different frameworks, its intuitive nature, and the ability to evaluate it directly on skeleton kinematics. Therefore, as our evaluation metric we report MPJPE in millimeters, calculated using

$$L_{\text{MPJPE}} = \frac{1}{J \times T} \sum_{t=1}^T \sum_{j=1}^J \|\hat{p}_{j,t} - p_{j,t}\|^2, \quad (23)$$

where  $\hat{p}_{j,t} \in \mathbb{R}^3$  is the predicted position of  $j^{\text{th}}$  joint in  $t^{\text{th}}$  frame while  $p_{j,t}$  is the corresponding ground truth. Lower error values indicate better agreement between the predictions and ground truths.

As in [3], [4] we generate predictions for different short-term and long-term prediction horizons. Specifically, pose sequence predictions of 80 ms, 160 ms, 320 ms, and 400 ms (i.e. 10 frame sequence) were generated as short-term predictions, while 560ms and 1000ms (i.e. 25 frames) length sequences were generated for long-term predictions.

### C. Comparisons with Existing State-of-the-art Methods

As baseline methods, we use state-of-the-art methods including Residual Sup [36], Traj-GCN [2], DMGNN [3], MSR-GCN [4], S-DGCN [12], PK-GCN [14], DANet [37], DPnet [15] and GA-MIN [13]. When choosing our baselines we ensured that a variety of different deep learning approaches, including recurrent neural networks, attention-based methods, graph neural networks, hybrid approaches, multi-scale and multi-stage GCN architecture, and geometric-inspired neural

network architectures, are compared, enabling a comprehensive comparison.

Quantitative comparisons for short-term and long-term prediction results for the H3.6M dataset are presented in Tabs. I and II, respectively. Following [14], [3], [11], [4] for long-term prediction we report average error metrics and evaluations only for five popular action categories for ease of presentation. GCN-based approaches such as MSR-GCN [4], S-DGCN [12], PK-GCN [14] have been able to achieve comparatively higher performance compared to the RNN-based Residual sup method. However, demonstrating the feature contamination issue when aggregating high-dimensional features using multi-stage graph convolution operation, they struggle to reach the robustness level of GA-MIN [13]. However, when comparing FMS-AM with GA-MIN, which is the previous state-of-the-art method we observe that our method has the lowest MPJPE on average for both short-term and long-term predictions. Specifically, we observe that the GA-MIN method struggles to generate accurate pose predictions over 400 ms on the H3.6M dataset. Large errors in the long-term prediction setting (i.e. Tab. II, 1000ms setting) for classes such as directions (i.e. MPJPE  $\sim 100$ ) and discussion (i.e. MPJPE  $> 106$ ) were observed where there is subject specificity compared to more generalised actions such as walking where we observe comparatively lower MPJPE (i.e. MPJPE  $\sim 43$ ). In contrast, our FMS-AM framework has been able to achieve consistent performance across all the action classes irrespective of subject-specific or class-specific motions. The average results of MPJPEs from Tabs I and II show that FMS-AM on average lowers the short-term prediction error (at 400ms) by 17.40% and the long-term prediction error by 17.54%, both significant margins. We believe the proposed feature factorisation strategy coupled with the innovative multi-head retrieval of the auxiliary memory has allowed superior learning capabilities where subject-specific, class-specific, and auxiliary information are better incorporated into the prediction of future motion sequences.

Similar, consistent performance of the FMS-AM framework is observed on the CMU-Mocap dataset for both short-term and long-term prediction tasks which are presented in Tabs. III and IV, respectively. Specifically, the previous state-of-the-art method, GA-MIN [13], struggles in short-term prediction of action classes such as jumping and soccer where we observe significant errors  $> 90\%$  and  $49\%$ , respectively at the 400 ms setting. Furthermore, for long-term predictions at the 1000 ms setting GA-MIN’s average error is  $\sim 85\%$ . In contrast, we observe a significant reduction in prediction error across both short-term and long-term prediction settings in our FMS-AM model where on average it lowers the short-term prediction error by 9.3% at the 320ms setting, and the long-term prediction at the 1000ms setting is reduced by 32 %. These results clearly indicate the utility of the proposed innovations.

In addition to quantitative comparisons with existing state-of-the-art methods in Fig. 5 we present a comparison between GA-MIN [13] and the proposed FMS-AM method for predicting long-term motion patterns in the CMU-Mocap dataset for the running and soccer action classes. When comparing

<sup>2</sup><http://mocap.cs.cmu.edu/>



TABLE I: Comparisons between the proposed method and state-of-the-art methods in terms of MPJPE for short-term prediction on the 15 action categories of H3.6M dataset. We also report the average across the action categories. The best results are highlighted in bold

Model	walking				eating				smoking				discussion			
	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Residual Sup [36]	29.36	50.82	76.03	81.51	16.84	30.60	56.92	68.65	22.96	42.64	70.14	82.68	32.94	61.18	90.92	96.19
DMGNN [3]	17.32	30.67	54.56	65.20	10.96	21.39	36.18	43.88	8.97	17.62	32.05	40.30	17.33	34.78	61.03	69.80
Traj-GCN [2]	12.29	23.03	39.77	46.12	8.36	16.90	33.19	40.70	7.94	16.24	31.90	38.90	12.50	27.40	58.51	71.68
MSR-GCN [4]	12.16	22.65	38.64	45.24	8.39	17.05	33.03	40.43	8.02	16.27	31.32	38.15	11.98	26.76	57.08	69.74
S-DGCN [12]	9.5	19.7	34.6	40.0	7.7	16.2	31.1	38.9	6.8	14.5	28.1	34.9	9.1	20.6	52.4	66.2
PK-GCN [14]	8.9	15.9	28.0	31.6	8.1	17.7	33.6	41.8	7.4	14.3	24.4	29.2	10.3	22.9	42.0	47.2
DANet [37]	9.7	19.0	33.5	39.4	6.1	13.6	27.6	34.7	6.4	13.1	25.6	31.9	8.8	19.1	39.6	50.1
DPnet [15]	7.3	15.2	30.1	32.6	8.6	18.3	36.4	43.5	6.9	13.5	24.3	28.7	8.2	20.1	38.2	43.0
GA-MIN [13]	7.5	13.5	28.2	30.6	5.8	12.5	25.3	33.8	6.2	12.1	24.2	24.2	8.2	18.6	30.5	46.3
FMS-AM	<b>5.1</b>	<b>10.3</b>	<b>18.8</b>	<b>20.4</b>	<b>4.2</b>	<b>8.7</b>	<b>16.5</b>	<b>21.2</b>	<b>4.6</b>	<b>9.3</b>	<b>17.1</b>	<b>12.7</b>	<b>5.5</b>	<b>12.1</b>	<b>19.6</b>	<b>26.8</b>

Model	directions				greeting				phoning				posing			
	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Residual Sup [36]	35.36	57.27	76.30	87.67	34.46	63.36	124.60	142.50	37.96	69.32	115.00	126.73	36.10	69.12	130.46	157.08
DMGNN [3]	13.14	24.62	64.68	81.86	23.30	50.32	107.30	132.10	12.47	25.77	48.08	58.29	15.27	29.27	71.54	96.65
Traj-GCN [2]	8.97	19.87	43.35	53.74	18.65	38.68	77.74	93.39	10.24	21.02	42.54	52.30	13.66	29.89	66.62	84.05
MSR-GCN [4]	8.61	19.65	43.28	53.82	16.48	36.95	77.32	93.38	10.10	20.74	41.51	51.26	12.79	29.38	66.95	85.01
S-DGCN [12]	8.3	18.8	42.5	51.9	13.7	30.4	68.6	85.3	8.1	18.0	37.6	47.9	8.8	22.9	58.3	73.8
PK-GCN [14]	8.6	23.7	46.5	56.2	13.3	27.2	67.3	83.1	11.4	20.2	37.7	43.2	9.1	23.6	65.8	81.2
DANet [37]	6.9	17.6	43.0	54.9	12.7	28.6	61.6	75.9	8.3	17.9	38.2	48.3	8.4	19.4	43.4	56.1
DPnet [15]	10.1	21.0	45.8	56.7	12.7	27.1	65.6	82.9	10.2	17.4	35.7	41.3	7.4	21.9	63.5	78.8
GA-MIN [13]	6.8	15.3	42.1	50.2	12.8	26.3	61.8	75.8	8.3	17.8	37.9	44.8	7.8	19.3	43.4	56.0
FMS-AM	<b>4.3</b>	<b>10.1</b>	<b>25.4</b>	<b>30.1</b>	<b>9.1</b>	<b>18.2</b>	<b>42.4</b>	<b>53.7</b>	<b>6.1</b>	<b>10.2</b>	<b>22.7</b>	<b>28.4</b>	<b>5.4</b>	<b>14.1</b>	<b>19.8</b>	<b>33.4</b>

Model	purchases				sitting				sittingdown				takingphoto			
	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Residual Sup [36]	36.33	60.30	86.53	95.92	42.55	81.40	134.70	151.78	47.28	85.95	145.75	168.86	26.10	47.61	81.40	94.73
DMGNN [3]	21.35	38.71	75.67	92.74	11.92	25.11	44.59	50.20	14.95	32.88	77.06	93.00	13.61	28.95	45.99	58.79
Traj-GCN [2]	15.60	32.78	65.72	79.25	10.62	21.90	46.33	57.91	16.14	31.12	61.47	75.46	9.88	20.89	44.95	56.58
MSR-GCN [4]	14.75	32.39	66.13	79.64	10.53	21.99	46.26	57.80	16.10	31.63	62.45	76.84	9.89	21.01	44.56	56.30
S-DGCN [12]	12.2	28.6	59.9	74.3	9.0	20.0	43.5	54.9	12.7	28.5	58.8	72.3	8.3	18.6	43.1	53.7
PK-GCN [14]	15.2	31.4	57.9	68.0	10.1	24.6	47.8	57.3	11.5	27.5	56.8	67.3	7.6	16.1	39.7	51.3
DANet [37]	12.4	28.5	60.1	73.6	8.9	19.3	42.6	53.9	14.6	30.7	59.9	73.0	8.0	17.7	39.9	51.0
DPnet [15]	17.8	37.0	62.1	65.6	9.1	23.0	48.1	62.8	9.7	24.2	49.7	62.0	5.7	14.4	35.6	47.9
GA-MIN [13]	12.4	28.5	60.0	72.9	8.1	18.5	41.9	53.2	14.5	25.5	56.3	70.3	8.3	16.6	38.2	49.0
FMS-AM	<b>10.2</b>	<b>20.8</b>	<b>37.3</b>	<b>51.5</b>	<b>5.9</b>	<b>12.6</b>	<b>20.8</b>	<b>32.1</b>	<b>11.2</b>	<b>17.1</b>	<b>34.6</b>	<b>46.4</b>	<b>6.1</b>	<b>10.5</b>	<b>27.0</b>	<b>36.2</b>

Model	waiting				walkingdog				walkingtogether				Average			
	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Residual Sup [36]	30.62	57.82	106.22	121.45	64.18	102.10	141.07	164.35	26.79	50.07	80.16	92.23	34.66	61.97	101.08	115.49
DMGNN [3]	12.20	24.17	59.62	77.54	47.09	93.33	160.13	171.20	14.34	26.67	50.08	63.22	16.95	33.62	65.90	79.65
Traj-GCN [2]	11.43	23.99	50.06	61.48	23.39	46.17	83.47	95.96	16.14	10.47	21.04	38.47	12.68	26.06	52.27	63.51
MSR-GCN [4]	10.68	23.06	48.25	59.23	20.65	42.88	80.35	93.31	10.56	20.92	37.40	43.85	12.11	25.56	51.64	62.93
S-DGCN [12]	8.7	19.4	43.7	54.8	18.5	38.8	71.8	85.3	8.5	18.3	35.2	41.9	10.0	22.2	47.3	58.4
PK-GCN [14]	9.5	23.0	55.9	63.6	21.3	42.4	83.7	95.1	9.4	19.3	36.3	44.8	10.8	23.3	48.2	57.4
DANet [37]	8.1	18.3	41.6	52.8	19.0	38.7	71.0	84.5	8.3	17.2	33.1	39.6	9.8	21.2	44.0	54.6
DPnet [15]	8.4	20.5	53.6	69.1	25.7	51.8	94.9	112.3	8.3	18.8	35.6	44.8	10.3	22.9	47.9	58.1
GA-MIN [13]	7.5	17.2	41.1	52.3	18.9	38.5	70.9	84.0	8.5	18.3	34.2	39.9	9.4	19.9	42.4	52.2
FMS-AM	<b>5.7</b>	<b>12.1</b>	<b>29.6</b>	<b>37.4</b>	<b>14.1</b>	<b>22.8</b>	<b>53.1</b>	<b>67.5</b>	<b>6.2</b>	<b>12.5</b>	<b>20.6</b>	<b>24.5</b>	<b>6.9</b>	<b>13.4</b>	<b>27.0</b>	<b>34.8</b>

TABLE II: Comparisons between the proposed method and state-of-the-art methods in terms of MPJPE for long-term prediction on 5 action categories of the H3.6M dataset. We also report average performance. The best results are highlighted in bold.

Model	walking		eating		smoking		discussion		directions		Average	
	560	1000	560	1000	560	1000	560	1000	560	1000	560	1000
Residual Sup [36]	81.73	100.68	79.87	100.20	94.83	137.44	121.30	161.70	110.05	152.48	97.56	130.50
DMGNN [3]	73.36	95.82	58.11	86.66	50.85	72.15	81.90	138.32	110.06	115.75	74.85	101.74
Traj-GCN [2]	54.05	59.75	53.39	77.75	50.74	72.62	91.61	121.53	71.01	101.79	64.16	86.69
MSR-GCN [4]	52.72	63.04	52.54	77.11	49.45	71.64	88.59	117.59	71.18	100.59	62.89	86.00
S-DGCN [12]	47.9	55.6	51.2	77.1	45.3	70.3	85.3	110.5	72.5	110.3	60.44	84.76
PK-GCN [14]	42.5	47.0	57.9	69.7	33.3	60.2	75.8	112.0	74.7	101.9	56.84	78.16
DANet [37]	46.7	55.6	48.0	73.6	44.1	68.0	71.7	108.2	72.9	106.0	56.68	82.28
DPnet [15]	40.5	48.6	56.5	69.6	32.8	59.9	66.3	96.7	80.2	103.5	55.26	75.66
GA-MIN [13]	35.5	42.8	47.3	65.2	30.6	46.5	60.3	106.3	68.1	100.0	48.36	72.16
FMS-AM	<b>23.4</b>	<b>31.6</b>	<b>28.7</b>	<b>49.3</b>	<b>17.8</b>	<b>38.2</b>	<b>39.8</b>	<b>79.2</b>	<b>46.5</b>	<b>74.8</b>	<b>31.04</b>	<b>54.62</b>

the long-term predictions of the two models, in particular beyond 400ms, we observe that GA-MIN struggles to generate accurate estimations of the future motion while the proposed method achieves significant robustness. Furthermore, we observe that the predictions generated by the proposed FMS-AM for complex action classes such as soccer, where there are unique subject-specific and task-specific motions, are far superior to GA-MIN. Please refer to supplementary material for additional qualitative results.

#### D. Ablation Evaluations

We conducted a series of ablation studies to systematically analyse the impact of the individual innovations that our FMS-AM framework proposes. Several design choices contribute to the robustness of our model: i) the proposed feature factorisation strategy; ii) the multi-head knowledge retrieval scheme; iii) the proposed auxiliary memory stabilisation losses; and iv) the dynamic mask generation procedure. All of these experiments were conducted on the H3.6M dataset and for testing the ablation models we use the validation set of H3.6M.

1) *Effects of Feature Factorisation:* To study the effect of the proposed feature factorisation strategy we generated four ablation variants of the proposed FMS-AM model: i) S-AM-w/o [F,Mh]: a model with an auxiliary memory and proposed stabilisation losses, but without feature factorisation and the multi-head knowledge retrieval scheme, ii) MhS-AM - 2F v1, iii) MhS-AM - 2F v2, and iv) MhS-AM - 2F v3 are models with an auxiliary memory, the proposed multi-head knowledge retrieval scheme and proposed stabilisation losses. They also possess the ability to factorise features. However, they only factorise the features into two factors. Please refer to Tab. V for details regarding the two factors that each of these models consider.

Comparison results are shown in Tab. V. We observe that the feature factorisation procedure is an integral part of the proposed framework. In particular, we observe that the auxiliary memory struggles to reach significant robustness levels without the feature factorisation scheme. With the introduction of two factorised levels, we observe higher levels of robustness where subject and task-specific attributes seem to make a significant impact (see the rows corresponding to MhS-AM - 2F v1, MhS-AM - 2F v2, and MhS-AM - 2F v3). Furthermore, we observe that the proposed stabilisation losses are capable of better aiding the learning process when the extracted feature representation is factorised. These experiments demonstrate the utility of our feature factorisation procedure.

In addition to qualitative results, in Fig. 6 we visualise the 2D projections of the factorised subject-specific and task-specific embeddings. In this experiment we utilise 200 randomly chosen samples from the CMU-Mocap dataset and use t-SNE for 2D projection of the factorised features. From Fig. 6 it is clear that the proposed factorisation strategy has been able to clearly separate the feature space into unique subject-specific and task-specific sub-spaces.

2) *Effects of Multi-Head Knowledge Retrieval and Auxiliary Memory Stabilisation Losses:* The effectiveness of the proposed multi-head knowledge retrieval scheme and the aux-

iliary memory stabilisation losses are evaluated in this experiment. We generated five ablation variants of the proposed FMS-AM model: i) F w/o [AM, Mh, S]: a model using the proposed feature factorisation, but without the auxiliary memory, the proposed stabilisation losses and multi-head knowledge retrieval scheme; ii) F-AM w/o [Mh, S]: this model extends the model in i) adding an auxiliary memory; iii) FMh-AM w/o S: the proposed model without the stabilisation and diversity losses; iv) FMh-AM-S v1: the proposed model with the diversity loss, but without the stabilisation loss; and v) FMh-AM-S v2: the proposed model with the stabilisation loss, but without the diversity loss.

From the results in Tab. VI we can confirm that there is a utility with respect to incorporating both subject-specific and task-specific historical knowledge captured by the auxiliary memory into the prediction framework, as evidenced by the rows in Tab. VI corresponding to models F w/o [AM, Mh, S] and F-AM w/o [Mh, S]. Furthermore, the experimental results presented in Tab. VI confirm our hypothesis that both multi-head knowledge retrieval and the proposed stabilisation losses contribute to the robustness of our model. We refer the reader to the rows corresponding to F-AM w/o [Mh, S] and FMh-AM w/o S in Tab. VI, where we observe a significant increase in the accuracy with the introduction of multi-head knowledge retrieval. This is because by utilising multiple factorised information cues, the read operation of our auxiliary memory can retrieve multiple items of salient information to aid prediction. Furthermore, rows corresponding to FMh-AM w/o S and FMS-AM confirm that stabilisation losses are integral parts of our framework. Moreover, we observe that both diversity and stabilisation losses are pivotal for the convergence of the knowledge captured within the auxiliary memory. Therefore, using this experiment we can confirm the necessity of both the multi-head knowledge retrieval scheme and the auxiliary memory stabilisation losses within our framework.

3) *Effects of Dynamic Mask Generation Procedure:* To better establish the contributions of the proposed dynamic mask generation procedure we conducted an additional ablation experiment. Note that in the previous two ablation experiments in situations where factorisation of features is leveraged, we utilise a dynamic masking strategy. However, in this experiment, we utilise both static masks (shown in Fig. 2) and the dynamic masks generated using the proposed mask generation procedure. Specifically, three ablation variants of the proposed model were generated: i) FS-AM w/o [Mh, DM]: the proposed model without multi-head retrieval and dynamic masking; ii) FMS-AM w/o DM: the model of i) with multi-head retrieval but without dynamic masking; and iii) [FS-AM, DM] w/o Mh: the model of i) with dynamic masking but without multi-head retrieval.

The results presented in Tab. VII confirm the need for dynamic masking. In particular, the ablation model without both the multi-head retrieval mechanism and the dynamic masking procedure (see the row corresponding to FS-AM w/o [Mh, DM] in Tab. VII) significantly struggles to achieve good prediction results. Furthermore, comparing rows corresponding to FMS-AM w/o DM and [FS-AM, DM] w/o Mh models in Tab. VII we can confirm that multi-head retrieval

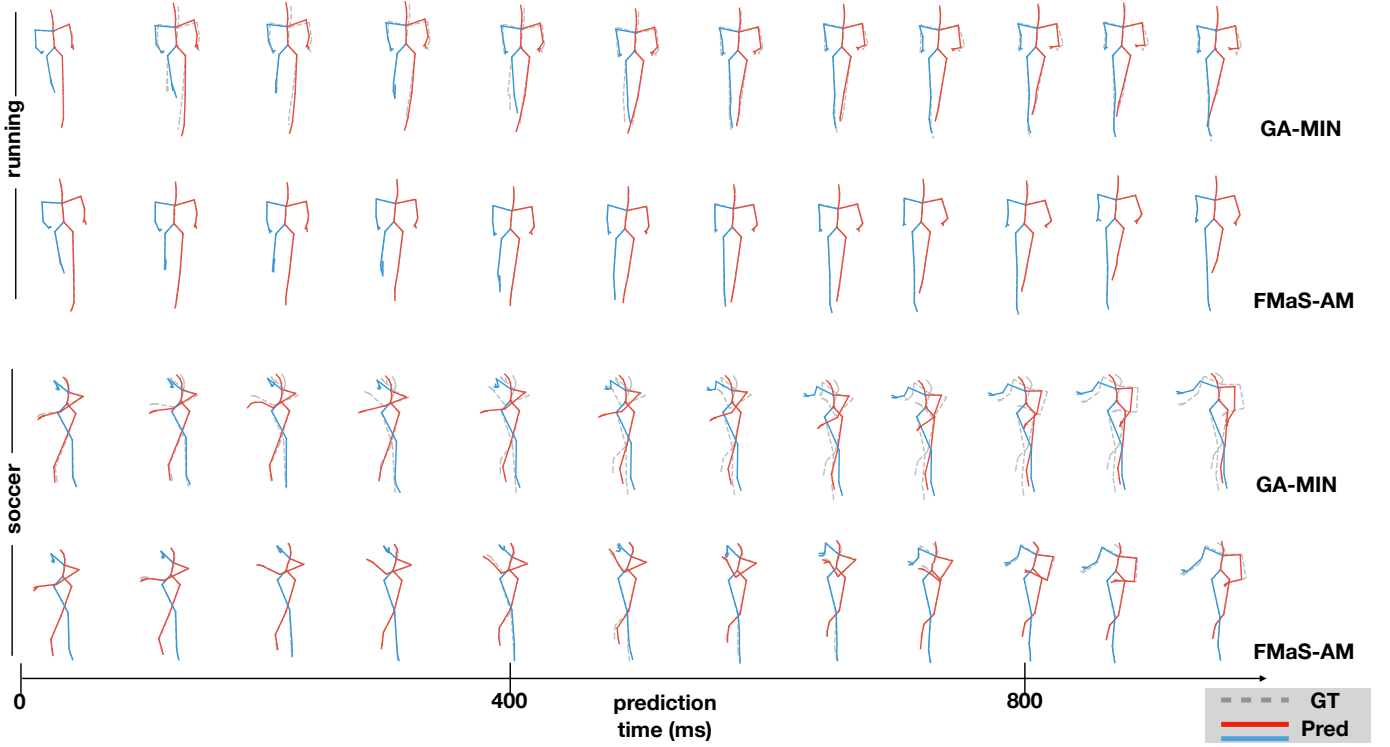


Fig. 5: Qualitative Results: A comparison between the existing state-of-the-art model, GA-MIN [13], and the proposed FMS-AM method for predicting long-term motion patterns for two action classes in the CMU-Mocap dataset.

TABLE III: Comparisons between the proposed method and state-of-the-art methods in terms of MPJPE for short-term predictions on 8 action categories of the CMU-Mocap dataset. We also report average performance across the action classes. The best results are highlighted in bold.

Model	basketball				basketball signal				directing traffic				jumping			
	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Residual Sup [36]	15.45	26.88	43.51	49.23	20.17	32.98	42.75	44.65	20.52	40.58	75.38	90.36	26.85	48.07	93.50	108.90
DMGNN [3]	15.57	28.72	59.01	73.05	5.03	9.28	20.21	26.23	10.21	20.90	41.55	52.28	31.97	54.32	96.66	119.92
Traj-GCN [2]	11.68	21.26	40.99	50.78	3.33	6.25	13.58	17.98	6.92	13.69	30.30	39.97	17.18	32.37	60.12	72.55
MSR-GCN [4]	10.28	18.94	37.68	47.03	3.03	5.68	12.35	16.26	5.92	12.09	28.36	38.04	14.99	28.66	55.86	62.93
DPnet [15]	10.7	17.8	38.4	49.5	2.6	4.4	10.0	13.4	5.9	11.8	26.6	33.5	12.4	28.3	70.2	89.2
GA-MIN [13]	10.3	19.8	40.3	51.8	2.5	4.6	10.5	15.3	<b>5.7</b>	10.8	27.2	33.4	14.2	28.2	71.8	91.1
FMS-AM	<b>6.2</b>	<b>14.6</b>	<b>26.2</b>	<b>34.1</b>	<b>2.4</b>	<b>3.2</b>	<b>6.8</b>	<b>10.6</b>	5.8	<b>8.4</b>	<b>19.8</b>	<b>26.4</b>	<b>9.3</b>	<b>16.3</b>	<b>56.2</b>	<b>79.2</b>

Model	running				soccer				walking				washwindow			
	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Residual Sup [36]	25.76	48.91	88.19	100.80	17.75	31.30	52.55	61.40	44.35	76.66	216.83	151.43	22.84	44.71	86.78	104.68
DMGNN [3]	17.42	26.82	38.27	40.08	14.86	25.29	52.21	26.23	9.57	15.53	26.03	30.37	7.93	14.68	33.34	44.24
Traj-GCN [2]	14.53	24.20	37.44	41.10	3.33	6.25	13.58	17.98	6.62	10.74	17.40	20.35	5.96	11.62	24.77	31.63
MSR-GCN [4]	12.84	20.42	30.58	34.42	3.03	5.68	12.35	16.26	6.31	10.30	17.64	21.12	5.49	11.07	25.05	32.51
DPnet [15]	16.7	18.4	19.6	25.1	9.0	17.1	35.8	48.7	5.8	9.0	17.2	21.4	4.5	<b>9.8</b>	27.3	36.7
GA-MIN [13]	17.5	22.3	22.1	26.1	9.8	18.3	39.0	49.4	5.2	8.9	16.2	18.2	4.5	9.9	27.8	35.2
FMS-AM	<b>10.1</b>	<b>12.6</b>	<b>15.3</b>	<b>18.8</b>	<b>6.7</b>	<b>11.2</b>	<b>26.8</b>	<b>31.4</b>	<b>3.9</b>	<b>6.1</b>	<b>11.8</b>	<b>14.9</b>	<b>4.4</b>	10.0	<b>18.1</b>	<b>21.7</b>

and dynamic masking complement each other. This is because depending on the context of the input partial activations and dynamic changes of the mask are possible, and by leveraging this dynamic mask the multi-head retrieval scheme can retrieve more informative content from the auxiliary memory. These observations strongly validate the importance of the proposed dynamic mask generation procedure.

#### E. Time Complexity

Our FMS-AM method contains 30.82M trainable parameters which is not a substantial increase of trainable parameters when compared with the state-of-the-art MGCN model [11]

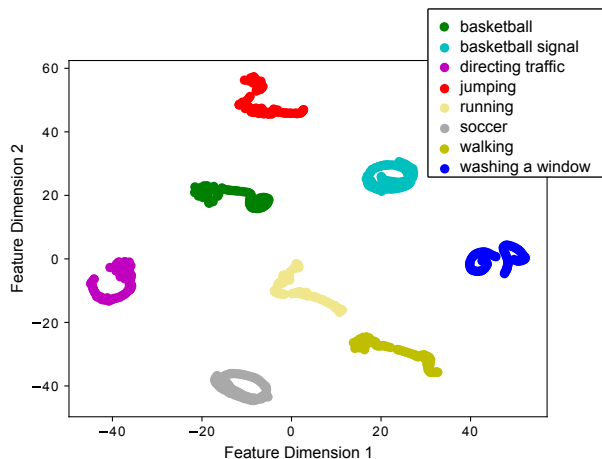
which has 26.51M trainable parameters when considering the significant performance increase that our FMS-AM method achieves. FMS-AM generates predictions (each of which has a prediction length of 1000 ms) for 100 input pose sequences in 5.85675 sec using a single NVIDIA A100 GPU. Note that this includes time taken for both feature extraction and the generation of model predictions.

#### V. CONCLUSION

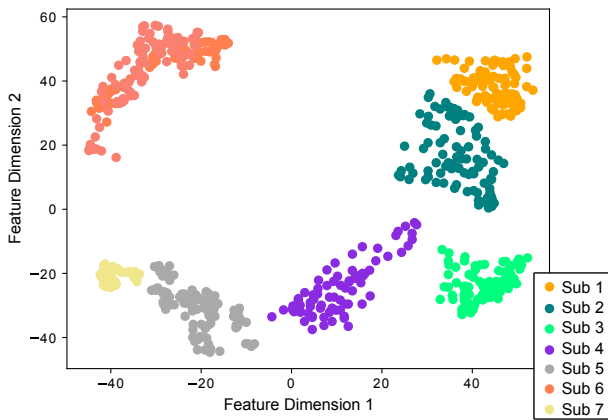
This paper presented a Factorised Multi-head retrieval and Stabilisation based Auxiliary Memory (FMS-AM) powered

TABLE IV: Comparisons between the proposed method and state-of-the-art methods in terms of average MPJPE for short-term and long-term predictions of the CMU-Mocap dataset. The best results are highlighted in bold. Note that ‘-’ represents the unavailability of baseline evaluations in that setting.

Model	Average					
	80	160	320	400	560	1000
DMGNN [3]	13.6	24.1	47.0	58.8	77.4	112.6
Traj-GCN [2]	11.2	19.1	36.3	-	45.8	95.7
S-DGCN [12]	7.6	14.3	29.0	36.6	50.9	80.1
PK-GCN [14]	9.4	17.1	32.8	40.3	52.2	79.3
DPnet [15]	8.4	14.5	30.6	39.7	-	91.3
GA-MIN [13]	8.7	15.4	31.9	-	40.1	84.5
FMS-AM	<b>6.1</b>	<b>10.3</b>	<b>22.6</b>	<b>29.6</b>	<b>32.4</b>	<b>52.5</b>



(a) Factorised Task Embeddings



(b) Factorised Subject Embeddings

Fig. 6: 2D projections obtained using t-SNE of the factorised subject-specific and task-specific features

framework for the accurate prediction of future human motions. We demonstrated that dynamic factorisation of the subject-specific and task-specific features plays a pivotal role in the effectiveness of the proposed architecture. Furthermore, our innovative multi-head knowledge retrieval scheme that leverages these factorised embeddings to generate multiple query operations is an integral part of our framework. The introduced loss functions guarantee that the knowledge captured within the auxiliary memory is not influenced by data imbalances or the diversity of the input data distributions. Extensive experiments were conducted on two public benchmarks: Human3.6M and CMU-Mocap, which demonstrated the ability of the proposed framework to outperform the current state-of-the-art algorithms by significant margins.

We observe two limitations of FMS-AM and suggest the following future research directions: (i) This study has only investigated the use of single auxiliary memory that stores subject-specific, task-specific, and auxiliary features together. The dynamic masking strategy is leveraged to query multiple representations out of the memory. In future works, we will investigate how separate topic-specific (i.e. subject, task, etc.) auxiliary memories can be incorporated to store these factorised features, which we believe will further improve the efficiency of knowledge retrieval. (ii) Furthermore, an investigation regarding dynamic auxiliary memory architectures such as graph-structured memory architectures is worthy of consideration.

## VI. ACKNOWLEDGEMENT

The research presented in this paper was supported by the Australian Research Council (ARC) Discovery grant DP200101942.

## REFERENCES

- [1] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, “Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 7, pp. 1325–1339, 2013.
- [2] W. Mao, M. Liu, M. Salzmann, and H. Li, “Learning trajectory dependencies for human motion prediction,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9489–9497.
- [3] M. Li, S. Chen, Y. Zhao, Y. Zhang, Y. Wang, and Q. Tian, “Dynamic multiscale graph neural networks for 3d skeleton based human motion prediction,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 214–223.
- [4] 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 IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11 467–11 476.
- [5] G. W. Taylor and G. E. Hinton, “Factored conditional restricted boltzmann machines for modeling motion style,” in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 1025–1032.
- [6] G. W. Taylor, G. E. Hinton, and S. Roweis, “Modeling human motion using binary latent variables,” *Advances in neural information processing systems*, vol. 19, 2006.
- [7] J. Wang, A. Hertzmann, and D. J. Fleet, “Gaussian process dynamical models,” *Advances in neural information processing systems*, vol. 18, 2005.
- [8] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik, “Recurrent network models for human dynamics,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4346–4354.
- [9] L.-Y. Gui, Y.-X. Wang, X. Liang, and J. M. Moura, “Adversarial geometry-aware human motion prediction,” in *Proceedings of the european conference on computer vision (ECCV)*, 2018, pp. 786–803.

TABLE V: Effect of the Proposed Feature Factorisation Scheme

Model	Auxiliary Memory	Factorisation			Multi-Head Access	Losses		Average					
		Subject	Task	Auxiliary		Diversity	Stabilisation	80	160	320	400	560	1000
S-AM w/o [F, Mh]	✓					✓	✓	9.5	23.2	48.8	58.4	57.4	78.7
MhS-AM - 2F v1	✓	✓		✓	✓	✓	✓	7.8	28.7	29.2	36.8	35.8	59.1
MhS-AM - 2F v2	✓		✓	✓	✓	✓	✓	7.5	17.3	28.4	36.8	35.7	58.9
MhS-AM - 2F v3	✓	✓	✓		✓	✓	✓	7.0	15.4	27.2	35.4	34.2	55.6
FMS-AM	✓	✓	✓	✓	✓	✓	✓	6.8	13.5	26.8	33.8	32.6	53.9

TABLE VI: Effects of the Proposed Multi Head Knowledge Retrieval Scheme and Auxiliary Memory Stabilisation Losses

Model	Auxiliary Memory	Factorisation			Multi Head Access	Losses		Average					
		Subject	Task	Auxiliary		Diversity	Stabilisation	80	160	320	400	560	1000
F w/o [AM, Mh, S]		✓		✓				10.0	24.1	48.2	58.4	64.3	85.8
F-AM w/o [Mh, S]	✓	✓	✓	✓				9.1	22.1	43.8	47.5	56.2	76.5
FMh-AM w/o S	✓	✓	✓	✓	✓			8.2	20.2	35.4	38.3	35.4	60.2
FMh-AM - S v1	✓	✓	✓	✓	✓	✓		7.4	15.8	28.1	36.4	34.2	56.8
FMh-AM - S v2	✓	✓	✓	✓	✓		✓	7.3	15.6	27.8	36.1	34.5	56.2
FMS-AM	✓	✓	✓	✓	✓	✓	✓	6.8	13.5	26.8	33.8	32.6	53.9

TABLE VII: Effects of the Proposed Dynamic Mask Generation Procedure

Model	Auxiliary Memory	Factorisation			Multi-Head Access	Losses		Dynamic Masking	Average					
		Subject	Task	Auxiliary		Diversity	Stabilisation		80	160	320	400	560	1000
FS-AM w/o [Mh, DM]	✓	✓	✓	✓		✓	✓		8.9	21.8	40.4	47.2	54.6	75.5
FMS-AM w/o DM	✓	✓	✓	✓	✓	✓	✓		8.5	18.2	34.4	37.6	38.3	64.5
[FS-AM, DM] w/o Mh	✓	✓	✓	✓	✓	✓	✓		8.8	20.4	38.9	45.4	50.5	73.1
FMS-AM	✓	✓	✓	✓	✓	✓	✓	✓	6.8	13.5	26.8	33.8	32.6	53.9

- [10] X. Guo and J. Choi, “Human motion prediction via learning local structure representations and temporal dependencies,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 2580–2587.
- [11] H. Zhou, C. Guo, H. Zhang, and Y. Wang, “Learning multiscale correlations for human motion prediction,” in *2021 IEEE International Conference on Development and Learning (ICDL)*. IEEE, 2021, pp. 1–7.
- [12] 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6437–6446.
- [13] J. Zhong and W. Cao, “Geometric algebra-based multiview interaction networks for 3d human motion prediction,” *Pattern Recognition*, p. 109427, 2023.
- [14] X. Sun, Q. Cui, H. Sun, B. Li, W. Li, and J. Lu, “Overlooked poses actually make sense: Distilling privileged knowledge for human motion prediction,” in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V*. Springer, 2022, pp. 678–694.
- [15] J. Tang, J. Zhang, R. Ding, B. Gu, and J. Yin, “Collaborative multi-dynamic pattern modeling for human motion prediction,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.
- [16] Y. Li, T. Yao, Y. Pan, and T. Mei, “Contextual transformer networks for visual recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.
- [17] H. Fan, Y. Yang, and M. Kankanhalli, “Point spatio-temporal transformer networks for point cloud video modeling,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 2, pp. 2181–2192, 2022.
- [18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, E. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [19] E. Aksan, M. Kaufmann, P. Cao, and O. Hilliges, “A spatio-temporal transformer for 3d human motion prediction,” in *2021 International Conference on 3D Vision (3DV)*. IEEE, 2021, pp. 565–574.
- [20] Y. Cai, L. Huang, Y. Wang, T.-J. Cham, J. Cai, J. Yuan, J. Liu, X. Yang, Y. Zhu, X. Shen *et al.*, “Learning progressive joint propagation for human motion prediction,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*. Springer, 2020, pp. 226–242.
- [21] D. Priyasad, A. Partovi, S. Sridharan, M. Kashefpoor, T. Fernando, S. Denman, C. Fookes, J. Tang, and D. Kaye, “Detecting heart failure through voice analysis using self-supervised mode-based memory fusion,” in *Proceedings of the 23rd INTERSPEECH Conference*. International Speech Communication Association, 2022, pp. 2848–2852.
- [22] D. Priyasad, T. Fernando, S. Denman, S. Sridharan, and C. Fookes, “Memory based fusion for multi-modal deep learning,” *Information Fusion*, vol. 67, pp. 136–146, 2021.
- [23] T. Fernando, C. Fookes, S. Denman, and S. Sridharan, “Detection of fake and fraudulent faces via neural memory networks,” *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1973–1988, 2020.
- [24] T. Fernando, S. Denman, D. Ahmed-Aristizabal, S. Sridharan, K. R. Laurens, P. Johnston, and C. Fookes, “Neural memory plasticity for medical anomaly detection,” *Neural Networks*, vol. 127, pp. 67–81, 2020.
- [25] T. Yang and A. B. Chan, “Visual tracking via dynamic memory networks,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 1, pp. 360–374, 2019.
- [26] D. Nguyen, P. Nguyen, H. Le, K. Do, S. Venkatesh, and T. Tran, “Memory-augmented theory of mind network,” *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.
- [27] C. C. Park, B. Kim, and G. Kim, “Towards personalized image captioning via multimodal memory networks,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 4, pp. 999–1012, 2018.
- [28] S. Park, S. Kim, S. Lee, H. Bae, and S. Yoon, “Quantized memory-augmented neural networks,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
- [29] M. Khademi, “Multimodal neural graph memory networks for visual question answering,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7177–7188.
- [30] H. Gammulle, S. Denman, S. Sridharan, and C. Fookes, “Forecasting future action sequences with neural memory networks,” *British Machine Vision Conference*, 2019.
- [31] M. Yang, Y. Li, P. Hu, J. Bai, J. Lv, and X. Peng, “Robust multi-view clustering with incomplete information,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 1055–1069, 2022.
- [32] F. Paredes-Vallés, K. Y. Scheper, and G. C. De Croon, “Unsupervised learning of a hierarchical spiking neural network for optical flow estimation: From events to global motion perception,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 8, pp. 2051–2064, 2019.
- [33] T. Fernando, S. Sridharan, S. Denman, and C. Fookes, “Split ‘n’ merge net: A dynamic masking network for multi-task attention,” *Pattern Recognition*, vol. 126, p. 108551, 2022.
- [34] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, “Memory augmented deep generative models for forecasting the next shot location in tennis,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 9, pp. 1785–1797, 2019.
- [35] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [36] J. Martinez, M. J. Black, and J. Romero, “On human motion prediction using recurrent neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2891–2900.

- [37] W. Cao, S. Li, and J. Zhong, “A dual attention model based on probabilistically mask for 3d human motion prediction,” *Neurocomputing*, vol. 493, pp. 106–118, 2022.