# Human Motion Prediction via Continual Prior Compensation

Jianwei Tang, Jian-Fang Hu, Tianming Liang, Xiaotong Lin, Jiangxin Sun, Wei-Shi Zheng, Jianhuang Lai

***Abstract*—Human Motion Prediction (HMP) aims to predict future human poses at different moments according to observed past motion sequences. Previous approaches mainly treated the prediction of different temporal moments as a single prediction task and learned the predictions of varied moments simultaneously, which would encounter a main limitation: the learning of short-term predictions (referring to "near-future" prediction) could be hindered by the predictions of long-term (referring to "far-future" prediction) motions. In this paper, we develop a novel temporal continual learning framework called Continual Prior Compensation (CPC) to progressively train HMP models, in which we divide the prediction task of motions corresponding to varied temporal moments into several subtasks and train the model in a multi-stage manner. To mitigate the prior information forgetting in the progressive training, we further introduce a learnable random variable Prior Compensation Factor (PCF) to explicitly measure the prior knowledge loss. We theoretically show that the PCF can be efficiently learned together with the model parameters by minimizing a reasonable upper bound of the objective function. The proposed CPC is further enhanced to estimate the prior information loss for each subtask and a new framework called Continual Prior Compensation++ (CPC++) with Fine-Grained Prior Compensation Factor (FGPCF) is finally developed. Our CPC and CPC++ frameworks are quite flexible and can be easily integrated with different HMP backbone models and adapted to various datasets and applications. Extensive experiments on three HMP benchmark datasets using multiple SOTA HMP backbones (PGBIG, siMLPe, MotionMixer, and LTD) demonstrate the effectiveness and flexibility of our frameworks.**

***Index Terms*—Human Motion Prediction, Continual Learning, Continual Prior Compensation.**

## I. Introduction

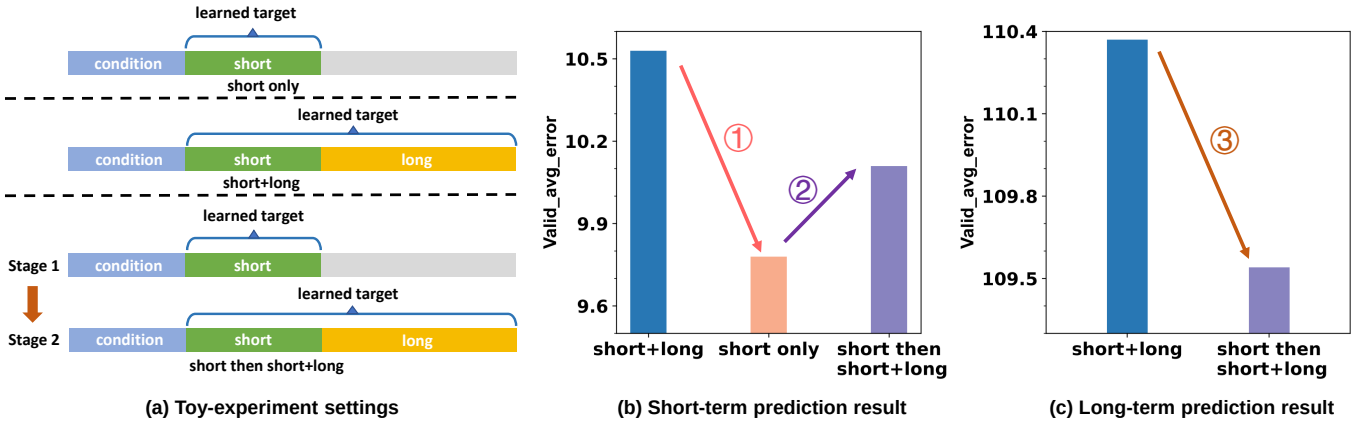
HUMAN Motion Prediction (HMP) aims to predict future poses at varied temporal moments based on the observed motion sequences. The accurate prediction of human motion plays a vital role in many applications, such as autonomous driving, human-robot interaction, and security monitoring, enabling the anticipation and mitigation of risks. This task is challenging due to its requirement for predicting multiple moments, including short-term predictions for the "near-future" and long-term predictions for the "far-future".

Previous approaches addressed this task by autoregressively forecasting using recurrent neural networks (RNNs) and transformer architectures [1]–[13], or parallelly generating all frames with graph convolution networks (GCNs) [14]–[21]. These methods employed the one-stage training strategy to acquire the model to learn both short and long-term prediction simultaneously. However, the long-term motion prediction is more challenging since the future motion can vary greatly (i.e., the prediction space is large), which would increase the uncertainty and ambiguity of future prediction. If we train short-term and long-term predictions together, the fitting of high-uncertainty long-term prediction will dominate the learning process, which hinders the learning of short-term predictions.

This motivates us to exploit proper training strategies to better learn the prediction for both short and long-term prediction. To this end, we conduct preliminary experiments with three settings: *short+long*, *short only* and *short then short+long*, as illustrated in Figure 1(a). We first observe that "short only" outperforms "short+long" on short-term prediction (Figure 1(b)), which implies that the joint learning of all timestamps is harmful to short-term prediction due to the fitting of high-uncertainty long-term prediction. This is intuitive and thus we can use a progressive learning approach, where the model is trained to predict increasing numbers of frames over multiple training stages, e.g., starting with 5 frames in the first stage and 10 frames in the second stage, and so on. However, we also observe that "short then short+long" performs worse than "short only" by a considerable margin for short-term prediction (Figure 1(b)), which demonstrates the joint learning of short-term and long-term prediction in subsequent stages results in knowledge forgetting for short-term prediction. At the same time, we observe that "short then short+long" outperforms "short+long" on long-term prediction in Figure 1(c), which implies the knowledge learned in short-term prediction can serve as prior facilitating the learning of "far-future" prediction.

Inspired by the above analysis, we propose the Continual Prior Compensation (CPC) framework, which is a multi-stage training framework that alleviates constraints posed by long-term prediction on short-term prediction and effectively utilizes prior information from short-term prediction. Specifically, we divide the future unobserved sequence into several segments and regard the prediction of each segment as a subtask. We define the training process as multiple stages and progressively increase the number of prediction segments across the stages. This allows the model to leverage the prior knowledge acquired from earlier stages for predicting the subsequent ones. Upon completion of each stage's training, the learned prior knowledge is saved in the model parameters. However, in this process, the learning in subsequent stages could gradually eliminate the prior knowledge learned from former stages. To overcome the prior forgetting problem, we further introduce the Prior

J. Tang, J.-F. Hu, T. Liang, X. Lin, J. Sun, W.-S. Zheng, and J. Lai are with the School of Computer Science, Sun Yat-sen University, Guangzhou, China. J.-F. Hu, W.-S. Zheng, and J. Lai are also with the Guangdong Province Key Laboratory of Information Security Technology and the Key Laboratory of Machine Intelligence and Advanced Computing (Sun Yat-sen University), Ministry of Education, China. The corresponding author is J.-F. Hu. E-mail: {tangjw7, liangtm, linxt29, sunjx5}@mail2.sysu.edu.cn, wszheng@ieee.org, {hujf5, stsljh}@mail.sysu.edu.cn.




(a) Toy-experiment settings (b) Short-term prediction result (c) Long-term prediction result

Fig. 1: Preliminary experiment results of three different prediction settings (lower values indicate better performance). (a) represents the experimental settings. (b) shows the short-term prediction results (predicting the 2-nd frame), while (c) illustrates the long-term prediction results (predicting the 25-th frame).

Compensation Factor (PCF), which is a learnable random variable that encodes the prior forgetting for all of the previous subtasks. However, directly learning the PCF and model parameters together is challenging. To this end, we further derive a reasonable upper bound of the objective function, which can be efficiently optimized.

In this work, we further introduce the Fine-Grained Prior Compensation Factor (FGPCF) and propose the Continual Prior Compensation++ (CPC++) framework to address the prior information forgetting problem more effectively. The FGPCF is defined to measure the prior information loss for each individual subtask. Similar to CPC, the enhanced CPC++ is also a multi-stage training approach, which intends to optimize the FGPCF and model parameters simultaneously. Our theoretical derivation shows that the CPC++ can be efficiently optimized by minimizing a reasonable upper bound function.

We evaluate our method by conducting experiments on three popular HMP benchmarks (Human3.6M [22], CMU-MoCap and 3DPW Dataset [23]) by integrating CPC and CPC++ with several HMP backbones (LTD [20], MotionMixer(MM) [24], siMLPe [25] and PGBIG [21]). Our results demonstrate that the proposed CPC and CPC++ training frameworks are flexible and can be easily integrated with various HMP backbones or adapted to different datasets. Additionally, the experimental results also indicate that the proposed PCF and FGPCF effectively mitigate the loss of prior knowledge, leading to improved performance in the HMP backbones.

In summary, our main contributions are threefold: 1) We identify certain limitations in existing HMP models and propose two novel multi-stage training strategies called Continual Prior Compensation and Continual Prior Compensation++ to enhance the training of the human motion prediction model. To the best of our knowledge, we are the first to develop such multi-stage training framework to progressively train human motion prediction models in a temporal continual learning manner. 2) We introduce the Prior Compensation Factor and Fine-Grained Prior Compensation Factor to tackle the forgetting problem of prior knowledge, which can be learned jointly with the prediction model parameters. 3) We theoretically derive an easily optimized and reasonable objective function for effective optimization. We also present an extensive experimental analysis of four backbone HMP models and three benchmark datasets to illustrate the effectiveness of the proposed approach.

A preliminary version of the current work was reported in [42], which uses a learnable random variable (PCF) to encode the prior forgetting for all of the previous subtasks. In this work, we have significantly extended our framework in the following three aspects. First, we introduce the Fine-Grained Prior Compensation Factor (FGPCF) containing multiple learnable random variables, each of which measures the prior information loss for the corresponding prediction subtask. Second, we propose the Continual Prior Compensation++ (CPC++) framework, which intends to optimize the FGPCF and model parameters simultaneously. And an efficient optimization algorithm is obtained through theoretical derivation. Third, we conduct a more comprehensive comparative analysis of our framework to demonstrate the advantages of utilizing the FGPCF for prior loss estimation and report improved results on three datasets.

## II. Related Work

Our work is closely related to one-stage training and multi-stage training approaches for human motion prediction, and continual learning, which have been extensively investigated in the community. In the following, we will provide a brief review of these works.

**One-stage training approaches for HMP.** Motivated by natural language processing, many researchers have adopted autoregressive models to process temporal sequences of human poses, which include RNNs, Long Short-Term Memory Networks (LSTMs) [26] and Transformer [27]. For instance, ERD [28] combined LSTMs with an encoder-decoder to model the temporal aspect, while Jain *et al.* [29] proposed Structural-RNN to capture spatiotemporal features of human motion. Martinez *et al.* [30] applied a sequence-to-sequence architecture for modeling the human motion structure. Aksan *et al.* [1] used

Transformer to predict future poses autoregressively. Sun *et al.* [12] designed a query-read process to retrieve motion dynamics from the memory bank. Lucas *et al.* [31] proposed a GPT-like [32] autoregressive method to generate human poses. However, autoregressive methods are difficult to train and suffer from error accumulation.

Some researchers employed parallel prediction methods to address HMP problem [14]–[21]. Many works ( [17], [33], [34]) used GCN to encode feature or to decode it, which associates different joints' information. Mao *et al.* [20] viewed a pose as a fully connected graph and used GCN to extract hidden information between any pair of joints. Martinez *et al.* [35] devised a transformer-based network to predict human poses. Sofianos *et al.* [36] proposed a method to extract spatiotemporal features using GCNs. And Ma *et al.* [21] tried to achieve better prediction results by progressively generating better initial guesses. Xu *et al.* [37] used multi-level spatial-temporal anchors to achieve diverse predictions. Wan et al. [38] proposed GGMotion to model human topology in groups which better leverages dynamics and kinematics priors. Ding et al. [39] proposed to capture temporal and spatial dependencies via a kinematic temporal convolutional network and spatial graph convolutional networks, respectively. Li et al. [40] constructed adaptive graph scattering across various body parts to capture motion dynamics. These work intended to facilitate motion prediction by developing stronger motion representations, which differs significantly from our objective.

**Multi-stage training approaches for HMP.** Some researchers proposed multi-stage prediction methods to handle this task. Yuan *et al.* [41] utilized a two-stage conditional Variational Autoencoder (cVAE) [42] model for diverse human motion prediction. Barquero *et al.* [43] introduced the BeLFusion model, a two-stage latent diffusion model for the diverse HMP task. They utilized a two-stage learning process, where the first stage learns the VAE generation process, and the second stage focuses on enhancing the diversity of sampling. However, it is worth noting that these models were all acquired to learn both short-term and long-term prediction simultaneously during the training process. It is significantly different from our multi-stage training framework which progressively trains the short-term and long-term predictions in different stages.

**Continual learning.** Although Deep Neural Networks (DNNs) have demonstrated impressive performance on specific tasks, their limitations in handling diverse tasks hinder their broader applicability. The most severe problem is the catastrophic forgetting problem when simply applying the DNNs to multiple tasks. Therefore, some researchers introduced the concept of Continual Learning (CL) [44] to DNNs to ensure that models retain the knowledge of previous tasks while learning new tasks. Kirkpatrick *et al.* [45] proposed the Elastic Weight Consolidation (EWC) method to overcome the catastrophic forgetting problem and improve the performance of multi-task problems. Shin *et al.* [46] introduced a method that addresses catastrophic forgetting in sequential learning scenarios by using a generative model to replay data from past tasks during the training of new tasks. It is important to note that the traditional CL approaches do not account for temporal correlation and cannot leverage data from previous tasks.

## III. Continual Prior Compensation

In this part, we introduce the details of the proposed Continual Prior Compensation (CPC) framework. We first provide a problem formulation in Section III-A. Section III-B introduces the formal modeling and derivation. Finally, the optimization strategy is presented in Section III-C.

### A. Problem Formulation

Human motion prediction aims to predict future motion sequences conditioned on the previously observed motion sequences. Formally, let us use $\mathbf{X}_{1:T_h} = [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_{T_h}] \in \mathbb{R}^{J\times D\times T_h}$ to denote the observed motion sequence of length $T_h$ where $\mathbf{X}_i$ indicates motion at temporal location $i$. Note that $J$ is the number of joints for each pose, and $D$ is the dimension of coordinates. $\mathbf{X}_{T_h+1:T_h+T_p} = [\mathbf{X}_{T_h+1}, \mathbf{X}_{T_h+2}, \cdots, \mathbf{X}_{T_h+T_p}] \in \mathbb{R}^{J\times D\times T_p}$ represents the motion sequence to be predicted. $T_p$ is the length of the predicted sequence. It can be regarded as a composition of multiple sequential prediction subtasks, which involves predicting motions at varied future moments.

With the above denotations, the multiple timestamp prediction problem (i.e., HMP prediction) can be formulated as solving the following optimization problem:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} P(\mathbf{X}_{T_h+1:T_h+T_p}|\mathbf{X}_{1:T_h};\boldsymbol{\theta}), \tag{1}$$

which means that our target is to find the optimal model that maximizes Equation (1). Here, $\boldsymbol{\theta}$ is the model parameters to be learned.

Previous approaches mainly formulated models to learn both short and long-term prediction simultaneously in a one-stage manner during the training process. However, one-stage training methods would encounter a main limitation: the learning of short-term predictions (referring to "near-future" prediction) could be hindered by the predictions of long-term (referring to "far-future" prediction) motions. In this work, we propose to conduct the model training in a multi-stage learning manner, in which the model is progressively trained by fitting ongoing prediction motion sequences. To this end, we partition the entire prediction interval into several consecutive segments and progressively tune the model to additionally fit new motion segments, which have been pre-trained with previous segments.

Specifically, we initially split the future sequence into $K$ segments with temporal boundaries $T_1, T_2, \cdots, T_K$, where $T_K = T_h + T_p$. And we denote the prediction of segment $k$ as task $Z_k$, which can be expressed as follows:

- Task $Z_1$ : $\mathbf{X}_{1:T_h} \rightarrow \mathbf{X}_{T_h+1:T_1}$
- Task $Z_2$ : $\mathbf{X}_{1:T_h} \rightarrow \mathbf{X}_{T_1+1:T_2}$
  ...
- Task $Z_K$ : $\mathbf{X}_{1:T_h} \rightarrow \mathbf{X}_{T_{K-1}+1:T_K}$

To be specific, the target of task $Z_1$ is to predict $\mathbf{X}_{T_h+1:T_1}$ conditioned on $\mathbf{X}_{1:T_h}$, and task $Z_k$ aims to predict $\mathbf{X}_{T_{k-1}+1:T_k}$ with $\mathbf{X}_{1:T_h}$ as condition. Therefore, based on the Bayesian formulation, Equation (1) can be expressed as:

$$\begin{aligned} P(Z_1 Z_2 \cdots Z_K;\boldsymbol{\theta}) = P(Z_K|Z_1 Z_2 \cdots Z_{K-1};\boldsymbol{\theta}) \\ P(Z_{K-1}|Z_1 Z_2 \cdots Z_{K-2};\boldsymbol{\theta}) \cdots P(Z_1;\boldsymbol{\theta}). \end{aligned} \tag{2}$$

Our target is to maximize the probability defined in Equation (2) which means finding an optimal model to accomplish all tasks. In the following, we denote "$Z_1 Z_2 \cdots Z_k$" as "$Z_{1:k}$" for simplicity.

In this way, we can train the model in a multi-stage manner, which facilitates the utilization of prior information from the previous tasks for predicting the subsequent tasks. Since the model is trained to optimize the prediction tasks $Z_{1:k-1}$ in training stage $S_{k-1}$, the knowledge of tasks $Z_{1:k-1}$ can be implicitly involved in its well-trained parameters $\boldsymbol{\theta}^*_{k-1}$. Initializing $\boldsymbol{\theta}_k$ as $\boldsymbol{\theta}^*_{k-1}$ can exploit the prior knowledge learned in previous tasks to assist the prediction of the next task. However, the change of the optimization objective in different training stages could also bring about the knowledge-forgetting problem due to the changing of parameters $\boldsymbol{\theta}_k$.

### *B. Objective of Continual Prior Compensation*

Here, we introduce the Prior Compensation Factor (PCF), which explicitly measures the knowledge forgotten in the multi-stage training.

*1)* ***Prior Compensation Factor****:* We define the Prior Compensation Factor (PCF) as a random variable $\alpha_{Z_{1:k-1}\to Z_k}$, which estimates the extent of forgotten knowledge when utilizing prior knowledge from tasks $Z_{1:k-1}$ to predict task $Z_k$. Specifically, the PCF can be formulated as:

$$\begin{aligned}\alpha_{Z_{1:k-1}\to Z_k} = P(Z_k|Z_{1:k-1};\boldsymbol{\theta}) - P(Z_k|\hat{Z}_{1:k-1};\boldsymbol{\theta}),\\ k \in \{2,3,\cdots,K\}.\end{aligned} \tag{3}$$

Here, $\hat{Z}_{1:k-1}$ is regarded as the prior knowledge that is reserved and can be still provided for predicting task $Z_k$. So $\hat{Z}_{1:k-1}$ initially represents the prior knowledge reserved in $\boldsymbol{\theta}^*_{k-1}$ in every stage $S_k$ and would get corrupted gradually during training. Therefore, $P(Z_k|\hat{Z}_{1:k-1};\boldsymbol{\theta})$ indicates the learning prediction ability of a new task using corrupted prior knowledge. While $P(Z_k|Z_{1:k-1};\boldsymbol{\theta})$ represents the most ideal case, where the current prediction task $Z_k$ can fully leverage the prior information provided by previous prediction tasks $Z_{1:k-1}$. Consequently, the loss of the prior knowledge is non-negative, implying that $0 \le \alpha_{Z_{1:k-1}\to Z_k} \le P(Z_k|Z_{1:k-1};\boldsymbol{\theta}) - P(Z_k|\hat{Z}_{1:k-1};\boldsymbol{\theta}) \le 1 - P(Z_k|\hat{Z}_{1:k-1};\boldsymbol{\theta})$. The larger the value of $\alpha_{Z_{1:k-1}\to Z_k}$ is, the more prior knowledge is forgotten. Specifically, we can observe that $\alpha_{Z_{1:k-1}\to Z_k} = 0$ when $P(Z_k|Z_{1:k-1};\boldsymbol{\theta}) = P(Z_k|\hat{Z}_{1:k-1};\boldsymbol{\theta})$, which implies that all the prior knowledge of previous tasks is completely exploited although the $\boldsymbol{\theta}$ changes. It is worth noting that the value of $\alpha_{Z_{1:k-1}\to Z_k}$ could be varied for different training samples.

*2)* ***Optimization Objective****:* By substituting Equation (3) into Equation (2) and taking the negative logarithm, we can obtain:

$$\begin{aligned}-\ln P(Z_{1:k};\boldsymbol{\theta}) = &-\ln P(Z_1;\boldsymbol{\theta})\\ &-\sum_{i=2}^{k}\ln(P(Z_i|\hat{Z}_{1:i-1};\boldsymbol{\theta}) + \alpha_{Z_{1:i-1}\to Z_i}).\end{aligned} \tag{4}$$

Our target turns to minimize $-\ln P(Z_{1:k};\boldsymbol{\theta})$ with respect to the model parameter $\boldsymbol{\theta}$ and prior compensation factors $\{\alpha_{Z_{1:i-1}\to Z_i}, i = 2,3,\cdots,k\}$. Optimizing Equation (4) directly is challenging due to the difficulty of determining the value of PCF during training. However, by applying Lemma 3.1 (details provided in the following), we can obtain an upper bound for $-\ln P(Z_{1:k};\boldsymbol{\theta})$, which can be expressed as:

$$\begin{aligned}UB = \sum_{i=2}^{k}&((1-\alpha_{Z_{1:i-1}\to Z_i})(-\ln P(Z_i|\hat{Z}_{1:i-1};\boldsymbol{\theta}))\\ &+ (1-\alpha_{Z_{1:i-1}\to Z_i})\ln(1-\alpha_{Z_{1:i-1}\to Z_i})\\ &+ \ln(1+\alpha_{Z_{1:i-1}\to Z_i})) - \ln P(Z_1;\boldsymbol{\theta}).\end{aligned} \tag{5}$$

Hence, we can turn to minimize the upper bound indicated by Equation (5). It is worth noting that in the optimization objective, $\alpha_{Z_{1:i-1}\to Z_i}$ serves as a factor to control the weights of different tasks. When the prior knowledge loss of previous tasks becomes severe, the weight of the current task's loss will be smaller, which mitigates the prior knowledge loss of previous tasks and thus compensates for the lost prior knowledge. We would like to point out that the largest difference between $-\ln P(Z_{1:k};\boldsymbol{\theta})$ and the upper bound $UB$ would not exceed $\ln(3/2) * (k-1)$ in the case of $P(Z_i|\hat{Z}_{1:i-1};\boldsymbol{\theta}) \ge 1/2, i \in \{2,3,\cdots,k\}$, which would be demonstrated in Lemma 3.2.

***Lemma 3.1.*** For $0 \le a \le 1-b$ and $0 < b \le 1$, the inequality $-\ln(a+b) \le (1-a)(-\ln b) + (1-a)\ln(1-a) + \ln(1+a)$ holds. The equality holds if and only if $a = 0$.

*Proof:* Let's consider the following function:

$$\begin{aligned}G(a) = &\ln(a+b) - (1-a)\ln b\\ &+ (1-a)\ln(1-a) + \ln(1+a).\end{aligned} \tag{6}$$

By defining $G_1(a) = \ln(a+b) - (1-a)\ln b$ and $G_2(a) = (1-a)\ln(1-a) + \ln(1+a)$, we can observe that $G(a) = G_1(a) + G_2(a)$.

Regarding $G_1(a)$, its derivative and second derivative can be computed as follows:

$$\begin{aligned}\dot{G}_1(a) &= \frac{1}{a+b} + \ln b\\ \ddot{G}_1(a) &= -\frac{1}{(a+b)^2}.\end{aligned} \tag{7}$$

For the case of $b \ge 1/e$, since $\ddot{G}_1(a) < 0$ and $0 \le a \le 1-b$, we can obtain that $\dot{G}_1(a) \ge \dot{G}_1(1-b) = 1 + \ln b \ge 0$. Therefore, $G_1(a) \ge G_1(0) = 0$. For the case of $0 < b < 1/e$, we can easily obtain that $\dot{G}_1(1-b) = 1+\ln b < 0$ and $\dot{G}_1(0) = 1/b + \ln b > 0$. Hence, by considering the monotonicity of $\dot{G}_1(a)$, we can conclude that there exists an $a_0 \in (0, 1-b)$ such that $\dot{G}_1(a) \ge 0$ for $a \in [0, a_0]$ and $\dot{G}_1(a) < 0$ for $a \in (a_0, 1-b]$. As a result, we can get $G_1(a) \ge \min(G_1(0), G_1(1-b))$. Since $G_1(0) = 0$ and $G_1(1-b) = -b\ln b \ge 0$, we can finally have $G_1(a) \ge 0$, the equality holds if only if $a = 0$.

In the following, we show that $G_2(a) \ge 0$ holds. The first derivative $\dot{G}_2(a)$ and second derivative $\ddot{G}_2(a)$ are given by:

$$\begin{aligned}\dot{G}_2(a) &= -\ln(1-a) - 1 + \frac{1}{1+a}\\ \ddot{G}_2(a) &= \frac{1}{1-a} - \frac{1}{(1+a)^2}.\end{aligned} \tag{8}$$

Since $\ddot{G}_2(a) \ge 0$ for any $a \in [0, 1-b]$, we can get $\dot{G}_2(a) \ge \dot{G}_2(0) = 0$ and thus $G_2(a) \ge G_2(0) = 0$.

With $G_1(a) \geq 0$ and $G_2(a) \geq 0$, we can obtain $G(a) = G_1(a) + G_2(a) \geq 0$ and conclude that $-\ln(a+b) \leq (1-a)(-\ln b) + (1-a)\ln(1-a) + \ln(1+a)$. Moreover, we can observe that equality holds if and only if $a = 0$.

***Lemma 3.2.*** The absolute difference between the target objective (Equation (4)) and the upper bound (Equation (5)) is not larger than $\ln(3/2) * (k-1)$ when $P(Z_i|\hat{Z}_{1:i-1};\boldsymbol{\theta}) \geq 1/2, i \in \{2,3,\cdots,k\}$. This bound is achieved when $P(Z_k|\hat{Z}_{1:k-1};\boldsymbol{\theta}) = 1/2$ and $\alpha_{Z_{1:i-1}\to Z_i} = 1/2, i \in \{2,3,\cdots,k\}$.

*Proof:* For simplicity, let's denote $\alpha_i = \alpha_{Z_{1:i-1}\to Z_i}$ and $p_i = P(Z_i|\hat{Z}_{1:i-1};\boldsymbol{\theta})$, where $p_i \in [1/2, 1], \alpha_i \in [0, 1-p_i], i \in \{2,3,\cdots,k\}$. Then the difference between the upper bound (Equation (5)) and the target objective (Equation (4)) can be calculated as:

$$\begin{aligned}\tau =& UB - (-\ln P(Z_{1:k};\boldsymbol{\theta})) \\ =& \sum_{i=2}^{k}(\ln(\alpha_i + p_i) - (1-\alpha_i)\ln p_i \\ &+ (1-\alpha_i)\ln(1-\alpha_i) + \ln(1+\alpha_i))).\end{aligned} \tag{9}$$

Note that each term in the summation of Equation (9) has the same form, we denote it as $T(\alpha)$, where

$$\begin{aligned}T(\alpha) =& \ln(\alpha+p) - (1-\alpha)\ln p \\ &+ (1-\alpha)\ln(1-\alpha) + \ln(1+\alpha).\end{aligned} \tag{10}$$

Then, we can turn to maximize $T(\alpha)$ in order to obtain the largest difference between the target objective and its upper bound.

The first derivative of $T(\alpha)$ can be calculated as:

$$\dot{T}(\alpha) = \frac{1}{\alpha+p} + \ln p - \ln(1-\alpha) - 1 + \frac{1}{1+\alpha}. \tag{11}$$

By defining $T_1(\alpha) = 1/(\alpha+p) + \ln p$ and $T_2(\alpha) = -\ln(1-\alpha) - 1 + 1/(1+\alpha)$, we can observe that $\dot{T}(\alpha) = T_1(\alpha) + T_2(\alpha)$. The first derivative of $T_2(\alpha)$ is:

$$\dot{T_2}(\alpha) = \frac{1}{1-\alpha} - \frac{1}{(1+\alpha)^2}. \tag{12}$$

Since $\dot{T_2}(\alpha) \geq 0$ for any $\alpha \in [0, 1-p]$, we get $T_2(\alpha) \geq T_2(0) = 0$. Furthermore, when $p \geq 1/2$, $T_1(\alpha) > 0$. Therefore, $\dot{T}(\alpha) = T_1(\alpha) + T_2(\alpha) > 0$, which means the maximum value of $T(\alpha)$ is:

$$\begin{aligned}T(1-p) &= -p\ln p + p\ln p + \ln(2-p) \\ &= \ln(2-p).\end{aligned} \tag{13}$$

Since $p \geq 1/2$ and $\ln(2-p)$ decreases with $p$, $T(1-p)$ will not larger than $\ln(2-1/2) = \ln(3/2)$. Particularly, when $p = 1/2$ and $\alpha = 1-p = 1/2$, this largest bound will be achieved. Hence, we can conclude that $T(\alpha) \leq \ln(3/2)$, and the equality holds when $p = \alpha = 1/2$.

Thus, as $\tau = \sum_{i=2}^{k} T(\alpha_i)$ in Equation (9), we obtain $\tau \leq \sum_{i=2}^{k} \ln(3/2) = \ln(3/2) * (k-1)$, which means the absolute difference between Equation (5) and Equation (4) will not be greater than $\ln(3/2) * (k-1)$. The equality holds if and only if $p_i = \alpha_i = 1/2, i \in \{2,3,\cdots,k\}$.

### C. Optimization Strategy

We train the model in a temporal continual learning (multi-stage) manner, in which a total of $K$ CPC stages are involved. In the first stage $S_1$, we aim to train the model for forecasting the motion in the foremost segment without estimating PCF. While for the stage $S_k, k \in \{2,3,\cdots,K\}$, we update our model for predicting motion segments of tasks $Z_{1:k}$ and measuring the PCF $\alpha_{Z_{1:k-1}\to Z_k}$. Once the model is completely trained, we turn to estimate the factors $\hat{\alpha}_{Z_{1:k-1}\to Z_k}$ for subsequent stages. This process is repeated until the final stage $S_K$ is finished. The algorithm flow is summarized in Algorithm 1. In the following, we elaborate on the learning process.

**Learning of initial stage $S_1$.** Following the implementations of previous methods [47], we can train the initial stage $S_1$ with mean suqared error (MSE) loss:

$$\mathcal{L}_1 = \sum_{i=T_h+1}^{T_1} \left\| \mathbf{X}_i - \hat{\mathbf{X}}_i \right\|^2, \tag{14}$$

where $\mathbf{X}_i$ and $\hat{\mathbf{X}}_i$ represent the ground truth and predicted motion of the $i$-th frames respectively.

**Learning of stage $S_k$.** In stage $S_k$ ($k \geq 2$), we need to update the model parameters $\boldsymbol{\theta}$ corresponding to tasks $Z_{1:k}$ and the PCF $\alpha_{Z_{1:k-1}\to Z_k}$. In practice, PCF can be calculated by adding an MLP head to the backbone model. According to Equation (5), the loss function in this stage can be calculated as follows:

$$\begin{aligned}\mathcal{L}_k =& (1-\alpha_{Z_{1:k-1}\to Z_k}) \sum_{i=T_{k-1}+1}^{T_k} \left\| \mathbf{X}_i - \hat{\mathbf{X}}_i \right\|^2 \\ &+ (1-\alpha_{Z_{1:k-1}\to Z_k})\ln(1-\alpha_{Z_{1:k-1}\to Z_k}) \\ &+ \ln(1+\alpha_{Z_{1:k-1}\to Z_k}) \\ &+ \sum_{j=2}^{k-1}(1-\hat{\alpha}_{Z_{1:j-1}\to Z_j}) \sum_{i=T_{j-1}+1}^{T_j} \left\| \mathbf{X}_i - \hat{\mathbf{X}}_i \right\|^2 + \mathcal{L}_1,\end{aligned} \tag{15}$$

where the parameters $\hat{\alpha}_{Z_1\to Z_2}, \cdots, \hat{\alpha}_{Z_{1:k-2}\to Z_{k-1}}$ are derter-mined in the learning of previous stages. Once the model parameters for stage $S_k$ are determined, we then calculate $\hat{\alpha}_{Z_{1:k-1}\to Z_k}$ as:

$$\hat{\alpha}_{Z_{1:k-1}\to Z_k} = \frac{1}{M}\sum_{m=1}^{M} \hat{\alpha}^m_{Z_{1:k-1}\to Z_k}, \tag{16}$$

where $M$ represents number of samples and $\hat{\alpha}^m_{Z_{1:k-1}\to Z_k}$ is PCF estimated for the $m$-th sample.

We continue the CPC training process by progressively predicting tasks $Z_{1:k+1}$ and updating the model parameter $\boldsymbol{\theta}$ as well as the PCF $\alpha_{Z_{1:k}\to Z_{k+1}}$. We repeat this CPC process until the final stage $S_K$ is achieved.

## IV. Continual Prior Compensation++

The Prior Compensation Factor in the CPC framework intends to encode the prior knowledge loss for all of the previous subtasks using a single random variable, which only provides a rough estimation of the prior knowledge loss. For example, when considering three tasks $Z_1, Z_2$, and $Z_3$, if task $Z_1$ experiences significant prior information loss while task

**Algorithm 1** Training procedure of proposed CPC framework.

**Require:** observed sequences $\mathbf{X}_{1:T_h}$, ground truth future sequences $\mathbf{X}_{T_h+1:T_h+T_p}$, model parameters $\boldsymbol{\theta}$, stage number $K$, training epoch for $k$-th stage $E_k$, learning rate $\lambda$, training sample number $M$.

**for** $i = 1$ to $E_1$ **do**
  $\hat{\mathbf{X}}_{T_h+1:T_1} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{X}_{1:T_h})$
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \lambda * \nabla_{\boldsymbol{\theta}} \mathcal{L}_1(\mathbf{X}_{T_h+1:T_1}, \hat{\mathbf{X}}_{T_h+1:T_1})$
**end for**
$\mathbf{B} = \emptyset$
**for** $k = 2$ to $K$ **do**
  **for** $i = 1$ to $E_k$ **do**
    $\hat{\mathbf{X}}_{T_h+1:T_k}, \alpha_{Z_{1:k-1} \to Z_k} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{X}_{1:T_h})$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \lambda * \nabla_{\boldsymbol{\theta}} \mathcal{L}_k(\mathbf{X}_{T_h+1:T_k}, \hat{\mathbf{X}}_{T_h+1:T_k}, \alpha_{Z_{1:k-1} \to Z_k}, \mathbf{B})$
  **end for**
  $\hat{\alpha}_{Z_{1:k-1} \to Z_k} = 0$
  **for** $m = 1$ to $M$ **do**
    $\alpha^m_{Z_{1:k-1} \to Z_k} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{X}^m_{1:T_h})$
    $\hat{\alpha}_{Z_{1:k-1} \to Z_k} = \hat{\alpha}_{Z_{1:k-1} \to Z_k} + \alpha^m_{Z_{1:k-1} \to Z_k}$
  **end for**
  $\hat{\alpha}_{Z_{1:k-1} \to Z_k} = \frac{1}{M} \hat{\alpha}_{Z_{1:k-1} \to Z_k}$
  $\mathbf{B} = \mathbf{B} \cup \{\hat{\alpha}_{Z_{1:k-1} \to Z_k}\}$
**end for**

$Z_2$ does not, the PCF $\alpha_{Z_{1:2} \to Z_3}$ remains small due to the influence of task $Z_2$. Consequently, the $1 - \alpha_{Z_{1:2} \to Z_3}$ will be large, preventing the model from effectively mitigating the prior forgetting problem of task $Z_1$. To handle this problem, we define the Fine-Grain Prior Compensation Factor (FGPCF), which contains multiple learnable random variables, each of which measures the prior information loss for the corresponding subtask. Based on the FGPCF, we formulate our Continual Prior Compensation++ (CPC++) framework, which intends to optimize the FGPCF and model parameters simultaneously in a multi-stage progressive training manner. We further derive a reasonable upper bound of the objective function for CPC++ through theoretical derivation, which can be efficiently optimized. In the following, we first provide a formal modeling and derivation of FGPCF as well as the optimization objective of CPC++ in Section IV-A. Section IV-B introduces the estimation of FGPCF. In Section IV-C, the optimization strategy is elaborated.

### A. Objective of Continual Prior Compensation++

*1)* ***Fine-Grained Prior Compensation Factor****:* To estimate the prior information loss for each task, we propose the Fine-Grain Prior Compensation Factor (FGPCF), which is defined as follows:

$$\alpha_i^{(k)} = \begin{cases} P(Z_{2:k}|Z_1;\boldsymbol{\theta}) - P(Z_{2:k}|\hat{Z}_1;\boldsymbol{\theta}), & i = 1, \\ P(Z_{i+1:k}|\hat{Z}_{1:i-1}Z_i;\boldsymbol{\theta}) - P(Z_{i+1:k}|\hat{Z}_{1:i};\boldsymbol{\theta}), & \\ \qquad\qquad i \in \{2,3,\cdots,k-1\}, \end{cases} \tag{17}$$

where $k \in \{2,3,\cdots,K\}$. $P(Z_{i+1:k}|\hat{Z}_{1:i-1}Z_i;\boldsymbol{\theta})$ indicates completing tasks $Z_{i+1:k}$ using corrupted prior information of tasks $Z_{1:i-1}$ and complete prior knowledge of task $Z_i$. And $P(Z_{i+1:k}|\hat{Z}_{1:i};\boldsymbol{\theta})$ represents completing tasks $Z_{i+1:k}$ with lost prior knowledge learned from tasks $Z_{1:i}$. Since the impact of prior information loss in previous tasks $Z_{1:i-1}$ is excluded using conditional probability, we can effectively measure the prior loss for the current task $Z_i$. Therefore, this approach enables a fine-grained assessment of information loss for each individual task, which leads to a more specific estimation of prior information loss and control of the training process.

*2)* ***Optimization Objective****:* Since the optimization function $-\ln P(Z_{1:k};\boldsymbol{\theta})$ is challenging to optimize directly, based on Lemma 3.3, we theoretically show that we can turn to minimize an upper bound of our optimization objective instead, which can be optimized easily. The upper bound is given as follows:

$$\begin{aligned} UB^{(k)} = &\sum_{i=2}^{k}(\prod_{j=1}^{i-1} A_j^{(k)})(-\ln P(Z_i|\hat{Z}_{1:i-1};\boldsymbol{\theta})) \\ &+ \sum_{i=2}^{k-1}(\prod_{j=1}^{i-1} A_j^{(k)})\Delta_i^{(k)} - \ln P(Z_1;\boldsymbol{\theta}) + \Delta_1^{(k)}, \end{aligned} \tag{18}$$

where $A_i^{(k)} = 1 - \alpha_i^{(k)}$, $\Delta_i^{(k)} = (1-\alpha_i^{(k)})\ln(1-\alpha_i^{(k)}) + \ln(1+\alpha_i^{(k)})$, $i \in \{1,2,\cdots,k-1\}$. In the following, we provide the description and proof of Lemma 3.3 in detail.

***Lemma 3.3.*** The inequality

$$\begin{aligned} -\ln P(Z_{1:k};\boldsymbol{\theta}) \le &\sum_{i=2}^{k}(\prod_{j=1}^{i-1} A_j^{(k)})(-\ln P(Z_i|\hat{Z}_{1:i-1};\boldsymbol{\theta})) \\ &+ \sum_{i=2}^{k-1}(\prod_{j=1}^{i-1} A_j^{(k)})\Delta_i^{(k)} - \ln P(Z_1;\boldsymbol{\theta}) + \Delta_1^{(k)} \end{aligned}$$

holds. The equality holds if and only if $A_i^{(k)} = 1$ and $\Delta_i^{(k)} = 0$, $i \in \{1,2,\cdots,k-1\}$.

*Proof:* We can rewrite $-\ln P(Z_{1:k};\boldsymbol{\theta})$ using conditional formular:

$$-\ln P(Z_{1:k};\boldsymbol{\theta}) = -\ln P(Z_{2:k}|Z_1;\boldsymbol{\theta}) - \ln P(Z_1;\boldsymbol{\theta}). \tag{19}$$

Using the definition from Equation (17) and Lemma 3.1, we can obtain:

$$\begin{aligned} -\ln P(Z_{1:k};\boldsymbol{\theta}) &= -\ln(P(Z_{2:k}|\hat{Z}_1;\boldsymbol{\theta}) + \alpha_1^{(k)}) - \ln P(Z_1;\boldsymbol{\theta}) \\ &\le -\ln P(Z_1;\boldsymbol{\theta}) + (1-\alpha_1^{(k)})(-\ln(P(Z_{2:k}|\hat{Z}_1;\boldsymbol{\theta})) \\ &\quad + (1-\alpha_1^{(k)})\ln(1-\alpha_1^{(k)}) + \ln(1+\alpha_1^{(k)}). \end{aligned} \tag{20}$$

For simplicity, let's denote $A_i^{(k)} = 1 - \alpha_i^{(k)}$, $\Delta_i^{(k)} = (1-\alpha_i^{(k)})\ln(1-\alpha_i^{(k)}) + \ln(1+\alpha_i^{(k)})$, $i \in \{1,2,\cdots,k-1\}$. By repeatedly utilizing Equation (17) and Lemma 3.1, we can

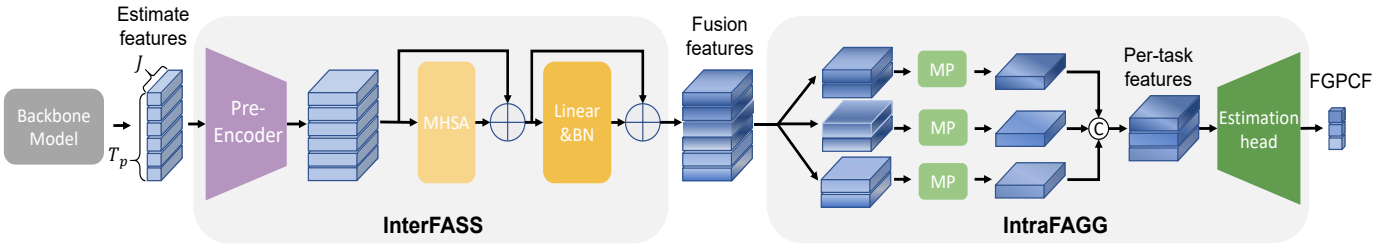



Fig. 2: Illustration of Prior Loss Estimation Module. InterFASS is the Inter-Task Forgetting Assessment module, and IntraFAGG represents the Intra-Task Forgetting Aggregation module. MP denotes the max-pooling layer.

obtain:

$$\begin{aligned}
&A_1^{(k)}(-\ln P(Z_{2:k}|\hat{Z}_1;\boldsymbol{\theta})) + \Delta_1^{(k)} - \ln P(Z_1;\boldsymbol{\theta}) \\
=&A_1^{(k)}(-\ln P(Z_{3:k}|\hat{Z}_1 Z_2;\boldsymbol{\theta}) - \ln P(Z_2|\hat{Z}_1;\boldsymbol{\theta})) \\
&+ \Delta_1^{(k)} - \ln P(Z_1;\boldsymbol{\theta}) \\
=&A_1^{(k)}(-\ln(P(Z_{3:k}|\hat{Z}_{1:2};\boldsymbol{\theta}) + \alpha_2^{(k)})) + \Delta_1^{(k)} \\
&+ A_1^{(k)}(-\ln P(Z_2|\hat{Z}_1;\boldsymbol{\theta})) - \ln P(Z_1;\boldsymbol{\theta}) \\
\leq&A_1^{(k)}A_2^{(k)}(-\ln P(Z_{3:k}|\hat{Z}_{1:2};\boldsymbol{\theta})) + A_1^{(k)}\Delta_2^{(k)} + \Delta_1^{(k)} \\
&+ A_1^{(k)}(-\ln P(Z_2|\hat{Z}_1;\boldsymbol{\theta})) - \ln P(Z_1;\boldsymbol{\theta}) \\
&\cdots \\
\leq&\sum_{i=2}^{k}(\prod_{j=1}^{i-1} A_j^{(k)})(-\ln P(Z_i|\hat{Z}_{1:i-1};\boldsymbol{\theta})) \\
&+ \sum_{i=2}^{k-1}(\prod_{j=1}^{i-1} A_j^{(k)})\Delta_i^{(k)} - \ln P(Z_1;\boldsymbol{\theta}) + \Delta_1^{(k)}.
\end{aligned} \tag{21}$$

When $A_i^{(k)} = 1$ and $\Delta_i^{(k)} = 0$, $i \in \{1, 2, \cdots, k-1\}$, the equality holds.

### B. Estimation of FGPCF

Here, we develop a Prior Loss Estimation Module (PLEM) to explicitly estimate the prior information loss. The PLEM consists of an Inter-Task Forgetting ASSessment (InterFASS) module and an Intra-Task Forgetting AGGregation (IntraFAGG) module, as illustrated in Figure 2. Specifically, consider a HMP backbone model with an output of $\boldsymbol{M}_{res} \in \mathbb{R}^{T\times N\times 3}$, where $T$ and $N$ denotes the motion length and the number of joints, respectively. We extend the model output with an additional dimension, generating an output of $\boldsymbol{M}_{res} \in \mathbb{R}^{T\times N\times(3+1)}$. We then extract features of the last dimension, with a size of $\boldsymbol{M}_{in} \in \mathbb{R}^{T\times N\times 1}$, and feed it into the InterFASS and IntraFAGG module, producing an output of $\boldsymbol{M}_{\alpha} \in \mathbb{R}^{K\times 1}$, which forms the estimation for FGPCF. The details of each module are elaborated in the following.

**Inter-Task Forgetting Assessment Module.** Firstly, the model employs a pre-encoder to encode the input features into a high-dimensional space, facilitating the subsequent estimation of the forgetting extent of prior information for each task. Considering that the FGPCF to be calculated is closely related to all the earlier tasks, we employ a multi-head self-attention mechanism (MHSA) to interact with the features of each task. To extract the information forgotten at each time step, attention is applied to each time step among all tasks. Subsequently, the output of MHSA is fed into a linear mapping and batch normalization.

**Intra-Task Forgetting Aggregation Module.** With the features after interaction through InterFASS provided, we then employ intra-task max-pooling to combine the features at each time step for various tasks. The combination features are then fed into an estimation head which contains a multilayer perceptron (MLP) and a sigmoid mapping. Finally, we can obtain an estimate of prior information loss (i.e., FGPCF) for each task on the given sample, which has a value from 0 to 1.

### C. Optimization Strategy

Similar to the CPC framework, CPC++ follows a multi-stage progressive training paradigm, in which a total of $K$ CPC++ stages are involved. In the first stage $S_1$, we aim to train the model for predicting the motion in the foremost segment without estimating FGPCF. While for the stage $S_k, k \in \{2, 3, \cdots, K\}$, we update our model to predict motion segments for task $Z_{1:k}$ and simultaneously learn the FGPCF $\alpha_i^{(k)}, i \in \{1, 2, \cdots, k-1\}$. This process is repeated until the final stage $S_K$ is achieved.

**Learning of initial stage $S_1$.** We train the initial stage $S_1$ with MSE loss:

$$\mathcal{L}_1 = \sum_{i=T_h+1}^{T_1} \left\| \mathbf{X}_i - \hat{\mathbf{X}}_i \right\|^2, \tag{22}$$

where $\mathbf{X}_i$ and $\hat{\mathbf{X}}_i$ represent the ground truth and predicted motion of the $i$-th frames respectively.

**Learning of stage $S_k$.** In stage $S_k$ ($k \geq 2$), we need to update the model parameters $\boldsymbol{\theta}$ corresponding to tasks $Z_{1:k}$ and estimate the FGPCF $\alpha_i^{(k)}, i \in \{1, 2, \cdots, k-1\}$. According to Equation (18), the loss function in this stage can be calculated as follows:

$$\begin{aligned}
\mathcal{L}_k = &\sum_{i=2}^{k}(\prod_{j=1}^{i-1} A_j^{(k)}) \sum_{v=T_{i-1}+1}^{T_i} \left\| \mathbf{X}_v - \hat{\mathbf{X}}_v \right\|^2 \\
&+ \sum_{i=2}^{k-1}(\prod_{j=1}^{i-1} A_j^{(k)})\Delta_i^{(k)} + \mathcal{L}_1 + \Delta_1^{(k)}.
\end{aligned} \tag{23}$$

We continue the CPC++ training process by completing tasks $Z_{1:k+1}$ and updating the model parameter $\boldsymbol{\theta}$ as well as FGPCF

TABLE I: Results on Human3.6M, CMU-MoCap and 3DPW using PGBIG as baseline. A lower value means better performance.

| Dataset | Method | 80ms | 160ms | 320ms | 400ms | 560ms | 1000ms |
|---|---|---|---|---|---|---|---|
| Human3.6M | PGBIG | 10.3 | 22.7 | 47.4 | 58.5 | 76.9 | 110.3 |
| | PGBIG&CPC | 9.4 (-0.9) | 21.3 (-1.4) | 45.7 (-1.7) | 56.8 (-1.7) | 75.4 (-1.5) | **108.8** (-1.5) |
| | PGBIG&CPC++ | **9.1** (-1.2) | **20.9** (-1.8) | **45.5** (-1.9) | **56.6** (-1.9) | **75.4** (-1.5) | 109.2 (-1.1) |
| CMU-MoCap | PGBIG | 7.6 | 14.3 | 29.0 | 36.6 | 50.9 | 80.1 |
| | PGBIG&CPC | **7.5** (-0.1) | 14.3 (-0.0) | 28.3 (-0.7) | 35.4 (-1.2) | 48.6 (-2.3) | 78.4 (-1.7) |
| | PGBIG&CPC++ | 7.6 (-0.0) | **14.2** (-0.1) | **27.9** (-1.1) | **34.9** (-1.7) | **48.0** (-2.9) | **76.8** (-3.3) |
| 3DPW | PGBIG | 13.7 | 23.2 | 47.3 | 58.3 | 76.1 | 104.1 |
| | PGBIG&CPC | 7.2 (-6.6) | 16.3 (-6.9) | 37.3 (-10.0) | 47.1 (-11.2) | **64.1** (-12.0) | **96.0** (-8.1) |
| | PGBIG&CPC++ | **7.1** (-6.7) | **16.2** (-7.0) | **37.1** (-10.2) | **47.0** (-11.3) | 64.5 (-11.6) | 96.3 (-7.8) |

$\alpha_i^{(k+1)}, i \in \{1, 2, \cdots, k\}$. The algorithm flow is summarized in Algorithm 2, where the $\mathbf{B}^{(k)} = \{\alpha_1^{(k)}, \alpha_2^{(k)}, \cdots, \alpha_{k-1}^{(k)}\}$.

**Algorithm 2** Training procedure of proposed CPC++ framework.

**Require:** observed sequences $\mathbf{X}_{1:T_h}$, ground truth future sequences $\mathbf{X}_{T_h+1:T_h+T_p}$, model parameters $\boldsymbol{\theta}$, stage number $K$, training epoch for $k$-th stage $E_k$, learning rate $\lambda$.
**for** $i = 1$ to $E_1$ **do**
  $\hat{\mathbf{X}}_{T_h+1:T_1} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{X}_{1:T_h})$
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \lambda * \nabla_{\boldsymbol{\theta}} \mathcal{L}_1(\mathbf{X}_{T_h+1:T_1}, \hat{\mathbf{X}}_{T_h+1:T_1})$
**end for**
**for** $k = 2$ to $K$ **do**
  **for** $i = 1$ to $E_k$ **do**
    $\hat{\mathbf{X}}_{T_h+1:T_k}, \mathbf{B}^{(k)} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{X}_{1:T_h})$,
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \lambda * \nabla_{\boldsymbol{\theta}} \mathcal{L}_k(\mathbf{X}_{T_h+1:T_k}, \hat{\mathbf{X}}_{T_h+1:T_k}, \mathbf{B}^{(k)})$
  **end for**
**end for**

## V. Experiments

### A. Experimental Setup

**Datasets.** We validate our framework on three benchmark datasets: Human3.6M [22], CMU-MoCap, 3DPW [23]. Human3.6M [22] is a large dataset containing 3.6 million 3D human pose data. 15 types of actions performed by 7 actors (S1, S5, S6, S7, S8, S9 and S11) are included in this dataset. Each actor is represented by a skeleton of 32 joints. However, following the data preprocessing method proposed in [20], [21], we only use 22 joints in the experiments. The global rotations and translations of poses are removed, and the frame rate is downsampled from 50 fps to 25 fps. We use actors S5 and S11 for testing and validation while conducting training on the remaining sections of the dataset. CMU-MoCap is a smaller dataset with 8 different action categories. The global rotations and translations of the poses are also removed. Following the data preprocessing methods in [20], [21], we use 25 joints to indicate human poses. 3DPW [23] is a challenging dataset that contains human motion data captured from both indoor and outdoor scenes. Poses in this dataset are represented in 3D space, with each pose containing 26 joints. However, only 23 of these joints are used, as the other three are redundant.

**Evaluation Metrics.** Following the benchmark protocols, we use the Mean Per Joint Position Error (MPJPE) in millimeters (ms) as our evaluation metric for 3D coordinate errors. We follow [20], [21] to report both short-term (80, 160, 320 and 400ms) and long-term predictions (560 and 1000ms). The performance is better if this metric is smaller.

**Implementation Details.** Following [20], [21], [24], we set the input length to 10 frames and the predictive output to 25 frames for Human3.6M and CMU-Mocap datasets, respectively. For the 3DPW dataset, we predict 30 frames conditioned on the observation of the preceding 10 frames. We choose PGBIG [21] as our baseline model by default. To learn the PCF, we add an extra dimension to the output of the backbone model and calculate PCF through an MLP network whose hidden dimension is set to 512. For estimating the FGPCF, we utilize the PLEM. The self-attention mechanism has four heads, and the hidden dimension of PLEM is configured as 64. We partitioned the future sequences into three segments with lengths of 3, 9, and 13. The training process was conducted on an NVIDIA RTX 3090 GPU for 120 epochs, allocating 50, 90, and 120 epochs for each stage. It should be noted that the inference process is the same as that of the baseline models.

**Backbones.** We apply our method on the following backbone approaches: LTD [20], MotionMixer (MM) [24], siMLPe [25] and the current state-of-the-art PGBIG [21] and report the experimental results on three benchmark datasets. LTD and PGBIG are GCN-based models. MotionMixer and siMLPe are MLP-based models. All these methods have released their code publicly. We employ their pre-trained models or re-train their models using the suggested hyper-parameters for a fair comparison. And we also exactly follow the metric they used to evaluate the results.

### B. Main Experimental Results

**Quantitative results.** Table I presents experimental results of the baseline model PGBIG with and without our training strategy (CPC and CPC++) on the Human3.6M, CMU-MoCap and 3DPW datasets. As shown, our frameworks outperform the corresponding baseline model by a considerable margin across all three datasets. Specifically, we can observe that the improvement in long-term prediction (e.g., 1000 ms) is greater compared to short-term prediction (e.g., 80 ms), indicating that the prior knowledge provided by short-term prediction is more crucial for challenging long-term prediction tasks. The results on the Human3.6M dataset show that our CPC framework decreases the prediction error by an average of 1.5 (52.9 vs. 54.4). Additionally, we can obtain a 1.7 (52.7 vs 54.4) error reduction using CPC++. When using our CPC strategy to train the PGBIG [21], we obtain about a 1.0 (35.4 vs. 36.4 in terms of average MPJPE) performance improvement on the CMU-Mocap dataset. Furthermore, when the CPC++ strategy is employed, we can achieve an additional improvement of 0.5. It is worth noting that our framework outperforms PGBIG by a margin of 9.1 (44.7 vs. 53.8 in terms of average MPJPE) on the 3DPW dataset. We attribute this to our multi-stage training framework, which effectively utilizes more comprehensive prior knowledge for subsequent predictions in this challenging dataset with smaller sizes and more complex motions. However, it also makes the difficult for the model with more parameters to learn the more specific FGPCF using this challenging dataset, which leads to the comparable performance between CPC++ and CPC on the 3DPW dataset.

**Qualitative results.** Figure 3 provides some visualization examples of predicted motions, qualitatively illustrating that our framework achieves more accurate results than the baseline model PGBIG. Specifically, in the "Directions" action, the person maintains an upright position throughout the sequence. In contrast, PGBIG exhibits a bent posture during long-term predictions, while our method accurately predicts positions closer to the ground truth. In the "Takingphoto" action, the person initially bends and then stands upright. PGBIG maintains the bent posture throughout long-term predictions, whereas our method accurately predicts posture changes. In the "Purchases" action, we observe that PGBIG's predicted results gradually bend, which differs from the ground truth. Furthermore, PGBIG's predicted results for the "Posing" action show an unnatural posture, where the person stands upright but with shoulders hunched forward and hands hanging vertically downward. It is evident that our method demonstrates an improvement in long-term prediction. This improvement is attributed to our proposed framework's ability to effectively leverage prior knowledge from short-term predictions, which provides long-term predictions with more comprehensive clues.

**Visualization of PCF and FGPCF.** In our CPC and CPC++ frameworks, we have employed the PCF and FGPCF to mitigate the loss of prior knowledge. In Figure 4, we show the value of them in different stages and different action types.

The PCF value progressively increases with each training stage for all action types. However, the trends in change for different actions are not uniform. For example, the value of "greeting" is smaller than that of "walkingdog" at the first stage but becomes larger at stage 5. This difference can be attributed to the static nature of the "greeting" action in the short term, where the action pattern undergoes minimal change. Consequently, the forgetting problem of prior knowledge learned in the initial stages is less severe, leading to a smaller PCF value. Conversely, the "greeting" action manifests in various forms over long-term sequences, resulting in different action patterns and a more severe forgetting problem. Additionally, we observe that the PCF value of "walking" is higher than that of "waiting" at stage 1 but lower at stage 5. This suggests that "walking" exhibits more movement in short-term sequences. As a result, when the periodic action pattern is learned in the earlier stages, the loss of prior knowledge is less pronounced in stage 5. In contrast, "waiting" remains static initially but undergoes posture changes in long-term sequences. By exactly examining the PCF/FGPCF results in Figure 4, we find that the learned FGPCF values increase with stage progresses for all action types, which means that the prior-information loss induced by multi-stage continual training becomes more severe. These results confirm the effectiveness of our approach in learning fine-grained prior compensation factors for mitigating the prior information forgetting in the multi-stage continual training. Furthermore, we can also observe that both the PCF and FGPCF values vary across actions. In particular, low-dynamic actions (e.g., sitting, smoking) exhibit consistently smaller PCF/FGPCF values. In contrast, for the actions with diverse and complex dynamics (e.g., walking dog and greeting), we observe that the learned PCF/FGPCF values are much larger in all stages. These observations suggest that exploring dynamic-specific prior information compensation mechanisms for motion prediction, by explicitly considering dynamic similarities across actions, could be beneficial and promising.

### C. Ablation Studies

We conduct several ablation experiments to further verify the effectiveness of our proposed framework.

**Integration with different baseline models.** We first study the flexibility by applying the CPC and CPC++ frameworks to progressively train different baseline models. Table II shows the quantitative comparisons of prediction results of different baseline models on the Human3.6M dataset. As shown, the performance of both short-term and long-term prediction improves when employing CPC and CPC++ in these baseline models. To be specific, we achieve a 3.6 performance improvement on average by applying CPC to LTD and a 4.1 performance improvement when using CPC++. Likewise, for the baseline models MM, CPC leads to a 1.1 performance enhancement, while CPC++ leads to a 1.6 performance enhancement. As for siMLPe, CPC also enhances prediction accuracy by approximately 1.2 and CPC++ improves the system performance by 1.5. These experimental results demonstrate that our proposed frameworks can effectively leverage prior knowledge to enhance the performance of both short and long-term prediction. Furthermore, these also validate the flexibility of applying CPC and CPC++ to various HMP baseline models, enhancing their performance.

TABLE II: Results of other baseline models on Human3.6M. A lower value means better performance.

| Method | Framework | 80ms | 160ms | 320ms | 400ms | 560ms | 1000ms |
|---|---|---|---|---|---|---|---|
| LTD | baseline | 12.7 | 26.1 | 52.3 | 63.5 | 81.6 | 114.3 |
| | CPC | 10.8 (-1.9) | 23.0 (-3.1) | 48.2 (-4.1) | 59.3 (-4.2) | 77.5 (-4.1) | 111.2 (-3.1) |
| | CPC++ | **10.2** (-2.5) | **22.3** (-3.8) | **47.3** (-5.0) | **58.5** (-5.0) | **77.2** (-4.4) | **111.0** (-3.3) |
| MM | baseline | 12.7 | 26.4 | 53.4 | 65.0 | 83.6 | 117.6 |
| | CPC | 11.5 (-1.2) | 25.0 (-1.4) | 52.0 (-1.4) | 63.7 (-1.3) | 82.8 (-0.8) | 117.1 (-0.5) |
| | CPC++ | **10.8** (-1.9) | **24.3** (-2.1) | **51.4** (-2.0) | **63.1** (-1.9) | **82.4** (-1.2) | **117.0** (-0.6) |
| siMLPe | baseline | 10.7 | 23.9 | 50.7 | 62.6 | 82.0 | 116.0 |
| | CPC | 10.0 (-0.7) | 22.9 (-1.0) | 49.4 (-1.3) | 61.2 (-1.4) | 80.6 (-1.4) | 114.6 (-1.4) |
| | CPC++ | **10.0** (-0.7) | **22.8** (-1.1) | **48.9** (-1.8) | **60.6** (-2.0) | **80.2** (-1.8) | **114.2** (-1.8) |

**Effect of PCF and FGPCF.** In this paper, we have introduced the prior compensation factor and fine-grained prior compensation factor to mitigate the prior forgetting problem during the multi-stage training process. Here, we investigate the benefits of them. In our experiments, we first report the results of the baseline "Without PCF", which trains in a multi-stage manner without using the PCF. We also report the results of using PCF "With PCF" and using FGPCF "With FGPCF".

The results presented in Figure 5 show that introducing the PCF alleviates the performance degradation from stage $S_1$ to stage $S_3$ of task $Z_1$'s predictions. Specifically, without PCF, the prediction error of task $Z_1$ increases by 0.83, whereas with PCF, it only increases by 0.27. This result suggests that PCF can effectively alleviate the prior forgetting issue. As a result, task $Z_1$ can offer more comprehensive priors for tasks $Z_2$ and $Z_3$, leading to better prediction performance. To be specific, the prediction error for task $Z_2$ at the end of stage $S_2$ is 45.33 without PCF, but it reduces to 44.43 when PCF is used. Similarly, for task $Z_3$ in stage $S_3$, the error decreases from 92.80 to 91.37 by using PCF. Furthermore, the results in Figure 5 demonstrate that FGPCF further mitigates the loss of prior knowledge. The prediction error of task $Z_1$ only increases by 0.14, significantly smaller than that of the baseline model. Therefore, we obtain better performance of tasks $Z_2$ and $Z_3$ by utilizing more comprehensive prior knowledge. These promising experimental results confirm that the proposed PCF and FGPCF help mitigate the prior forgetting problem.

**Evaluation on Prior Loss Estimation Module.** In our CPC++ framework, we have proposed the prior loss estimation module to better estimate the FGPCF, which contains the inter-task forgetting assessment and intra-task forgetting aggregation modules. Here, we conduct experiments to validate the effectiveness of these modules. In our experiments, we first remove both modules and obtain the baseline, "w/o both". Then, the result of adding InterFASS is reported as "with InterFASS". Furthermore, we test the result by only adding IntraFAGG, labeled as "with IntraFAGG". Finally, we use both modules, which is referred to as "with both".

The results are presented in Table III. We can observe that without the InterFASS and IntraFAGG, the prediction error is 65.7. When InterFASS is employed, the error decreases to 65.0, demonstrating the importance of accurately estimating the extent of prior knowledge loss. The prediction error is reduced to 65.5 by using IntraFAGG, which indicates that IntraFAGG is beneficial for the estimation of prior knowledge loss. With both modules added, the error further drops to 64.9.

TABLE III: Results of our method with/without InterFASS and IntraFAGG modules.

| Method | InterFASS | IntraFAGG | Avg↓ |
|---|---|---|---|
| w/o both | | | 65.7 |
| with InterFASS | ✓ | | 65.0 |
| with IntraFAGG | | ✓ | 65.5 |
| with both | ✓ | ✓ | **64.9** |

**Evaluation on different implementations.** In our CPC framework, we introduce the prior compensation factor to address the issue of prior forgetting. This factor is adaptively determined based on the backbone output. To demonstrate the importance of incorporating PCF, learned during the training process, we have conducted a series of experiments. In our experiments, we compare the results of four different implementations. PGBIG is the baseline model. "w/o $\alpha$" means that we only divide the training process into several stages to train each task without using the PCF. "HC" represents using a hand-crafted coefficient that changes its values similar to our PCF at each epoch. Specifically, in stage $S_1$, the value of $\alpha$ is set to 1. In stage $S_2$, $\alpha$ is initially set to 0.1 and increased by 0.05 at each epoch until reaching 0.5, where it remains constant. The same pattern applies to stage $S_3$.

The results are presented in Figure 6. In each subfigure, we show the results of task $Z_k$ on the validation set which is obtained at stage $S_k$. In stage $S_1$, where no prior information is available, the results of "w/o $\alpha$", "HC", and "Ours" are identical, but superior to the baseline. This validates the effectiveness of decomposing multiple-moment predictions, as it alleviates the constraint of long-term predictions on short-term predictions and enhances the model's ability to learn short-term predictions. In the following stages, the performance of "w/o $\alpha$" is worse than "Ours", which indicates that the prior

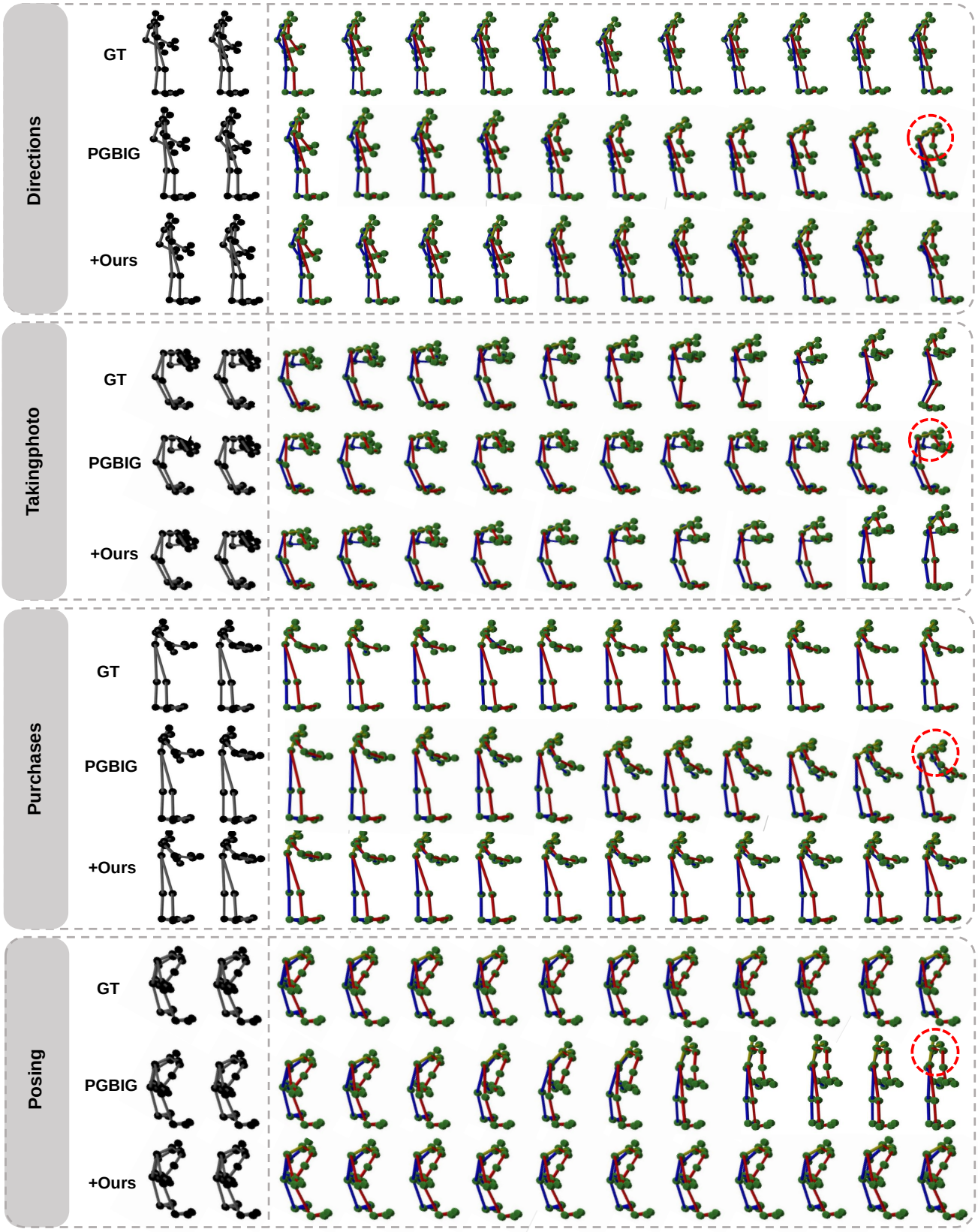



Fig. 3: Some visualized results from the Human3.6M dataset are presented. We display four action categories: "Directions", "Takingphoto", "Purchases", and "Posing". The observed frames are shown in black, while the colorful motion sequences represent the prediction results. "GT" denotes the ground truth results and "PGBIG" is the baseline. "+Ours" represents the results of using the proposed CPC framework. As demonstrated, our method produces the best future motion sequences. And we highlight the different parts in PGBIG's results with red dashed circles.

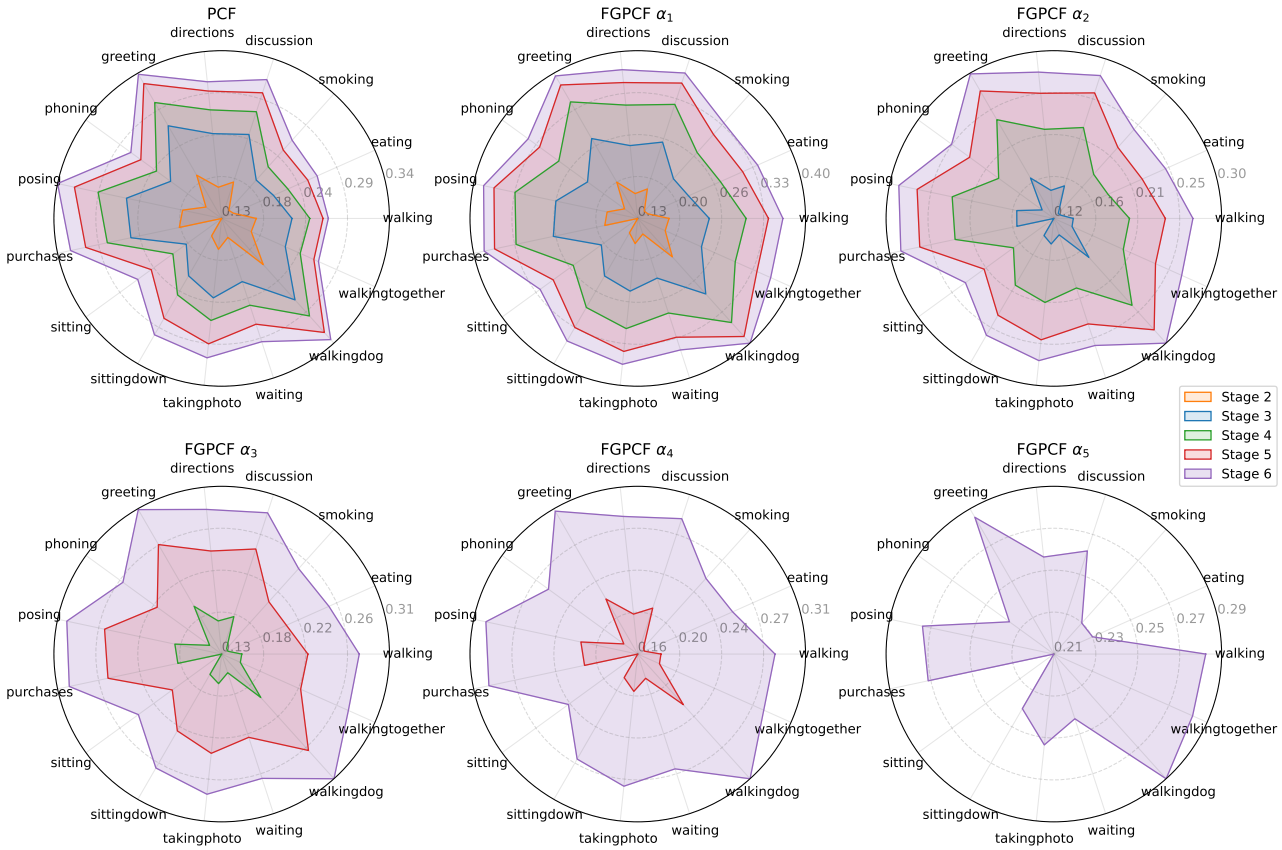



Fig. 4: Visualized results of PCF and FGPCF. Each subplot is represented in polar coordinates, where the angular coordinate represents different action categories, and the radial distance represents values. The drawn circular area represents the results of a certain stage.

| Method | Stage \ Task | z1 | z2 | z3 |
|---|---|---|---|---|
| Without PCF | s1 | 9.03 | | |
| | s2 | 9.44 | 45.33 | |
| | s3 | 9.86 | 45.70 | 92.80 |
| With PCF | s1 | 9.03 | | |
| | s2 | 9.10 | 44.43 | |
| | s3 | 9.30 | 44.62 | 91.37 |
| With FGPCF | s1 | 9.03 | | |
| | s2 | 9.10 | 44.43 | |
| | s3 | 9.17 | 44.51 | 91.35 |

Fig. 5: The average error of different tasks at the end of each stage. A lower value means better performance.

knowledge exploited by our framework benefits the prediction model training. Moreover, as the training period progressed, the performance gap became larger. However, the method without PCF can still achieve better performance than "PGBIG", indicating that the training model in a multi-stage manner can also exploit some useful prior information for prediction. We also note that the performance of "Ours" is much better than "HC", which conducts temporal continual learning with fixed and manually defined PCF. It demonstrates that joint training PCF and model parameters is beneficial.

**Evaluation on the number of tasks.** In most of our implementations, our CPC framework divides the future prediction into three tasks. Here, we study the influence of the number of tasks. Specifically, 1 task represents the training of short and long-term prediction together. In the 2 tasks setting, we partition the future sequences into two segments with lengths of 3 and 22. As for 5 tasks, we design prediction lengths of 3, 4, 5, 6 and 7 for each task. We split the future sequences into eight segments with lengths of 3, 3, 3, 3, 3, 3, 3 and 4 for the 8-tasks setting. The detailed comparison results are shown in Table IV. As shown, the model's performance improves as the number of tasks gets larger from 1 to 3, and it remains stable when the number of tasks becomes larger than 3. It demonstrates that the proposed framework for task division is beneficial for human motion prediction.

TABLE IV: The average error of different numbers of tasks.

| number of tasks | 1 | 2 | 3 | 5 | 8 |
|---|---|---|---|---|---|
| avg error↓ | 66.95 | 66.02 | **65.00** | 65.05 | 65.03 |

**Effect of the overlapping vs. non-overlapping segments.**

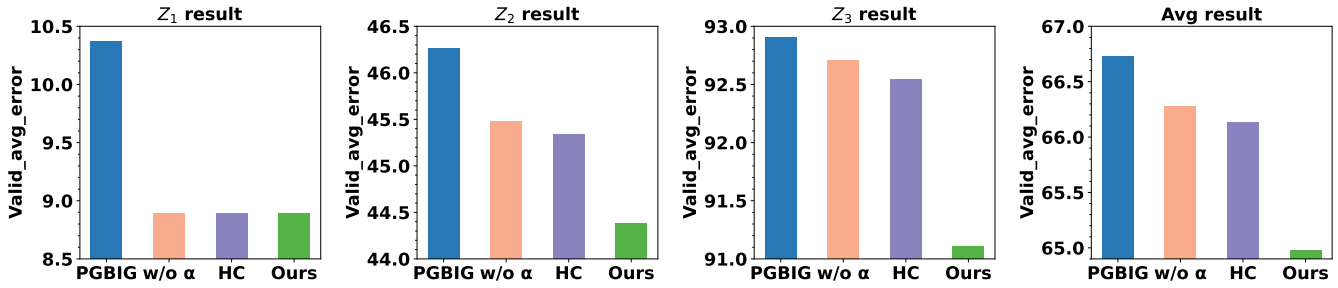



Fig. 6: Comparison of different approaches. PGBIG is a baseline model that is trained without multi-stage. "w/o $\alpha$" represents training multi-stage process without PCF. "HC" means using a manually designed coefficient. "Ours" is a multi-stage training process with PCF. A lower value means better performance.

TABLE V: Comparison results (MPJPE) of our method with different overlaps. X-overlap denotes the X-frame overlap between adjacent segments.

| Method | 80ms | 160ms | 320ms | 400ms | 560ms | 1000ms |
|---|---|---|---|---|---|---|
| PGBIG (Baseline) | 10.3 | 22.7 | 47.4 | 58.5 | 76.9 | 110.3 |
| 6-overlap | 9.9 | 21.9 | 46.5 | 57.6 | 76.5 | 110.4 |
| 3-overlap | 9.8 | 21.8 | 46.5 | 57.6 | 76.3 | 110.3 |
| no-overlap | **9.1** | **20.9** | **45.5** | **56.6** | **75.4** | **109.2** |

TABLE VI: Comparison results (MPJPE) of implementing multi-stage training with different strategies (single head vs. multi-head).

| Method | 80ms | 160ms | 320ms | 400ms | 560ms | 1000ms |
|---|---|---|---|---|---|---|
| PGBIG | 10.3 | 22.7 | 47.4 | 58.5 | 76.9 | 110.3 |
| +multi-head | 10.0 | 22.0 | 47.0 | 58.0 | 77.1 | 111.1 |
| +single shared-head | 9.4 | 21.3 | 45.7 | 56.8 | 75.4 | 108.8 |

In our progressive training framework, we split motions into non-overlapping segments and perform temporal continual learning across segments with prior compensation. To verify the effect of this design, we also implemented our approach using overlapping segments. The detailed results are presented in Table V. As expected, non-overlapping training yields the best performance, significantly outperforming implementations with overlapping segments, demonstrating the effectiveness of our proposed continual learning framework with prior compensation.

**Evalution on the single PCF/FGPCF prediction head across stages.** In our framework, we develop a single shared head to predict PCF/FGPCF across stages. Note that employing K output heads to generate K sub-predictions is also a feasible way to implement multi-stage training, in which each head mainly handles the prediction of a certain stage. However, this design faces scalability issues in long-term motion prediction since the number of head would grow with the total motion length T (e.g., K=[T/$\Delta$], where $\Delta$ represents length of motion predicted by each head), causing parameter bloat, higher memory/compute, and training instability. Moreover, the predictions outputted by multi-head decoder could encounter a termporal consistency issue, especially when the snippts are not overlapped. To emprically verify the effectivess of our single shared head, we also implement our multi-stage training with multi-head decoders and combined it with SOTA model PGBIG. The detailed results on Human3.6M dataset are presented in Table VI. As shown, our method with single shared-head obtains the best prediction performances and outperforms the multi-head variant in both short-term and long-term motion prediction, demonstrating the effectiveness of our approach in leveraging short-term priors to progressively train models across stages. We can also observe that performing multi-stage training with multi-head decoder improves short-term prediction (80–400ms) over PGBIG but slightly degrades long-term prediction (500–1000ms) accuracy, indicating that short-term priors are not effectively exploited for long-term motion prediction in this design.

**Analysis on the training time.** In Table VII, we provide the detailed training time of different methods with our progressive training strategy. As shown, our three-stage training strategy only slightly increases the total training time (e.g., from 16.3h to 19.5h for the PGBIG backbone) while significantly improving the overall performance from 64.9 to 67.0. The slight increase in training time can be attributed to the fact that learning the short-term prediction is much easier than optimizing over the entire sequence (8.1h vs. 16.3h), while the training in the subsequent stages can converge faster by leveraging the prior knowledge pretrained in previous stages. It is worth noting that our method is only used in the training stage, the inference time remains identical to that of the baseline model.

TABLE VII: Total training time comparison. The training procedure is stopped when the loss variation remains below 1e-5 within 5 epochs.

| Method | Avg. err. | stage 1 | stage 2 | stage 3 | total |
|---|---|---|---|---|---|
| PGBIG | 67.0 | 16.3 h | - | - | 16.3 h |
| PGBIG+Ours | 64.9 | 8.1 h | 6.0 h | 5.4 h | 19.5 h |
| LTD | 70.2 | 8.4 h | - | - | 8.4 h |
| LTD+Ours | 67.3 | 5.3 h | 3.0 h | 3.1 h | 11.4 h |

**Discussion on the assumption $P(Z_i|\hat{Z}_{1:i-1};\theta) \geq 1/2$.** In **Lemma 3.2**, we have derived the maximum difference between target objective and the upper bound (UB) under the assumption $P(Z_i|\hat{Z}_{1:i-1};\theta) \geq 1/2$. To validate the plausibility of this assumption, we have carefully examined the samples

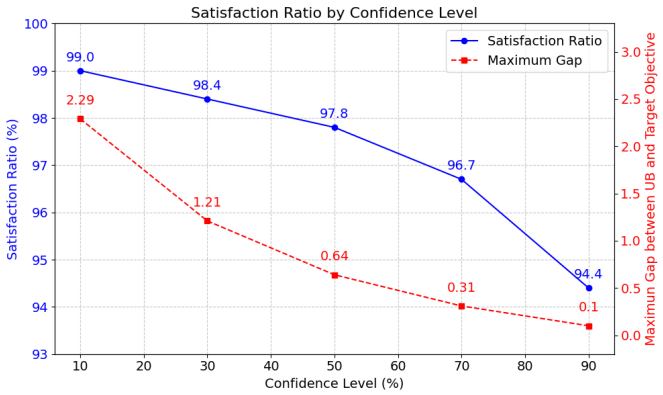



Fig. 7: The ratio of samples and the upper bound of maximun gap between UB and target objective when the confidence value is larger than a certain level.

in our experiments on the Human3.6M dataset with PGBIG as baseline, and present Figure 7 to demonstrate the ratio of samples and maximum gap between UB and target objective when the confidence value is larger than a certain level. We empirically find that a relatively large number of samples can satisfy the condition/assumption. For instance, 97.8% of the samples satisfy the condition $P(Z_i|\hat{Z}_{1:i-1};\theta) >= 1/2$. When the confidence level decreases, the number of samples meeting the condition increases, while the maximum gap between UB and target objective becomes larger accordingly. For non-satisfying cases, we cannot mathematically derive an exact formulation for the maximum gap between UB and target objective, but we empirically examine the non-satisfying samples and find that their gaps are still in a reasonable range (from 0.64 to 2.29). And more importantly, such samples account for only 2.2% of the total.

**Discussion on the training strategy of our method and Transformer.** Both our method and the Transformer involve processing sequences of varying lengths during training. Specifically, transformer processes all the sequences of varied lengths simultaneously in a single stage. In contrast, our method follows multi-stage training strategy, which processes the sequences of a certain length in each training stage. Here, we conduct experiments on the Human3.6M dataset and compare our results with POTR [48], which is trained following the standard Transformer paradigm. To ensure a fair comparison, we use the same EAE metric (in radians, ranging from 0 to $\pi$) as POTR for evaluation. The detailed results are presented in Table VIII. As shown, applying our temporal continual training strategy to POTR achieves better performance than the standard Transformer paradigm and the performance gain is much larger for long-term motion prediction (e.g., 400ms-1000ms) than short-term motion prediction (e.g., 80ms-160ms), demonstrating the effectiveness of our progressive multi-stage training paradigm to explore prior knowledge for facilitating human motion prediction.

## VI. Conclusion

In this work, we proposed to progressively train the human motion prediction model in the temporal continual learning framework. Specifically, two different multi-stage training approaches (i.e., continual prior compensation and continual prior compensation++) are developed to progressively train the human motion prediction model. We introduce the prior compensation factor and fine-grained prior compensation factor to explicitly mitigate the information-forgetting problem that occurs in multi-stage model training. Furthermore, we theoretically show that the PCF and FGPCF can be efficiently learned together with the model parameters by minimizing a reasonable upper bound of the objective function. Extensive experiments demonstrated our framework's effectiveness and flexibility. We believe our work has value for not only human motion prediction but also for more general prediction tasks and backbone models [49]–[52], which forms one of our future work.

TABLE VIII: Experimental results on Transformer-based model POTR with different training strategies.

| Method | 80ms | 160ms | 320ms | 400ms | 560ms | 1000ms |
|---|---|---|---|---|---|---|
| Transformer | 0.235 | 0.581 | 0.990 | 1.143 | 1.362 | 1.826 |
| Ours | 0.230 | 0.553 | 0.921 | 1.051 | 1.266 | 1.729 |
| Gain | 0.005 | 0.028 | 0.069 | 0.092 | 0.096 | 0.097 |

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62476296), Guangdong Natural Science Funds Project (2023B1515040025, 2022B1111010002), Guangdong NSF for Distinguished Young Scholar (2022B1515020009), Guangdong Provincial Key Laboratoryof Information Security Technology (2023B1212060026).

## References

[1] 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.

[2] H.-k. Chiu, E. Adeli, B. Wang, D.-A. Huang, and J. C. Niebles, "Action-agnostic human pose forecasting," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019, pp. 1423–1432.

[3] E. Corona, A. Pumarola, G. Alenya, and F. Moreno-Noguer, "Context-aware human motion prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6992–7001.

[4] P. Ghosh, J. Song, E. Aksan, and O. Hilliges, "Learning human motion models for long-term predictions," in *2017 International Conference on 3D Vision (3DV)*. IEEE, 2017, pp. 458–466.

[5] A. Gopalakrishnan, A. Mali, D. Kifer, L. Giles, and A. G. Ororbia, "A neural temporal model for human motion prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 116–12 125.

[6] 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.

[7] J. Tang, J. Wang, and J.-F. Hu, "Predicting human poses via recurrent attention network," *Visual Intelligence*, vol. 1, no. 1, p. 18, Aug 2023. [Online]. Available: https://doi.org/10.1007/s44267-023-00020-z

[8] L.-Y. Gui, Y.-X. Wang, D. Ramanan, and J. M. Moura, "Few-shot human motion prediction via meta-learning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 432–450.

[9] 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.

[10] Z. Liu, S. Wu, S. Jin, Q. Liu, S. Lu, R. Zimmermann, and L. Cheng, "Towards natural and accurate future motion prediction of humans and animals," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 004–10 012.

[11] H.-F. Sang, Z.-Z. Chen, and D.-K. He, "Human motion prediction based on attention mechanism," *Multimedia Tools and Applications*, vol. 79, no. 9, pp. 5529–5544, 2020.

[12] J. Sun, Z. Lin, X. Han, J.-F. Hu, J. Xu, and W.-S. Zheng, "Action-guided 3d human motion prediction," *Advances in Neural Information Processing Systems*, vol. 34, pp. 30 169–30 180, 2021.

[13] J. Tang, H. Yang, T. Chen, and J.-F. Hu, "Stochastic human motion prediction with memory of action transition and action characteristic," in *Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference*, 2025, pp. 1883–1893.

[14] E. Aksan, M. Kaufmann, and O. Hilliges, "Structured prediction helps 3d human motion modelling," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7144–7153.

[15] Q. Cui and H. Sun, "Towards accurate 3d human motion prediction from incomplete observations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4801–4810.

[16] Q. Cui, H. Sun, and F. Yang, "Learning dynamic relationships for 3d human motion prediction," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 6519–6527.

[17] 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.

[18] J. Liu and J. Yin, "Multi-grained trajectory graph convolutional networks for habit-unrelated human motion prediction," *arXiv preprint arXiv:2012.12558*, 2020.

[19] W. Mao, M. Liu, and M. Salzmann, "History repeats itself: Human motion prediction via motion attention," in *European Conference on Computer Vision*. Springer, 2020, pp. 474–489.

[20] 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.

[21] 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.

[22] 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.

[23] T. Von Marcard, R. Henschel, M. J. Black, B. Rosenhahn, and G. Pons-Moll, "Recovering accurate 3d human pose in the wild using imus and a moving camera," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 601–617.

[24] A. Bouazizi, A. Holzbock, U. Kressel, K. Dietmayer, and V. Belagiannis, "Motionmixer: Mlp-based 3d human body pose forecasting," *arXiv preprint arXiv:2207.00499*, 2022.

[25] 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 IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 4809–4819.

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[28] 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.

[29] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-rnn: Deep learning on spatio-temporal graphs," in *Proceedings of the ieee conference on computer vision and pattern recognition*, 2016, pp. 5308–5317.

[30] 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.

[31] T. Lucas, F. Baradel, P. Weinzaepfel, and G. Rogez, "Posegpt: Quantization-based 3d human motion generation and forecasting," in *European Conference on Computer Vision*. Springer, 2022, pp. 417–435.

[32] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018.

[33] T. Lebailly, S. Kiciroglu, M. Salzmann, P. Fua, and W. Wang, "Motion prediction using temporal inception module," in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[34] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Symbiotic graph neural networks for 3d skeleton-based human action recognition and motion prediction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 6, pp. 3316–3333, 2021.

[35] A. Martínez-González, M. Villamizar, and J.-M. Odobez, "Pose transformers (potr): Human motion prediction with non-autoregressive transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2276–2284.

[36] T. Sofianos, A. Sampieri, L. Franco, and F. Galasso, "Space-time-separable graph convolutional network for pose forecasting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11 209–11 218.

[37] S. Xu, Y.-X. Wang, and L.-Y. Gui, "Diverse human motion prediction guided by multi-level spatial-temporal anchors," in *European Conference on Computer Vision*. Springer, 2022, pp. 251–269.

[38] S. Wan, "Ggmotion: Group graph dynamics-kinematics networks for human motion prediction," *arXiv preprint arXiv:2507.07515*, 2025.

[39] R. Ding, K. Qu, and J. Tang, "Ksof: Leveraging kinematics and spatio-temporal optimal fusion for human motion prediction," *Pattern Recognition*, vol. 161, p. 111206, 2025.

[40] M. Li, S. Chen, Z. Zhang, L. Xie, Q. Tian, and Y. Zhang, "Skeleton-parted graph scattering networks for 3d human motion prediction," in *European conference on computer vision*. Springer, 2022, pp. 18–36.

[41] Y. Yuan and K. Kitani, "Dlow: Diversifying latent flows for diverse human motion prediction," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*. Springer, 2020, pp. 346–364.

[42] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[43] G. Barquero, S. Escalera, and C. Palmero, "Belfusion: Latent diffusion for behavior-driven human motion prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 2317–2327.

[44] M. B. Ring, "Child: A first step towards continual learning," *Machine Learning*, vol. 28, no. 1, pp. 77–104, 1997.

[45] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[46] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," *Advances in neural information processing systems*, vol. 30, 2017.

[47] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7482–7491.

[48] Á. Martínez-González, M. Villamizar, and J. Odobez, "Pose transformers (POTR): human motion prediction with non-autoregressive transformers," *CoRR*, vol. abs/2109.07531, 2021. [Online]. Available: https://arxiv.org/abs/2109.07531

[49] J. Sun, J. Xie, J.-F. Hu, Z. Lin, J. Lai, W. Zeng, and W.-s. Zheng, "Predicting future instance segmentation with contextual pyramid convlstms," in *Proceedings of the 27th acm international conference on multimedia*, 2019, pp. 2043–2051.

[50] Z. Lin, J. Sun, J.-F. Hu, Q. Yu, J.-H. Lai, and W.-S. Zheng, "Predictive feature learning for future segmentation prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7365–7374.

[51] J.-F. Hu, J. Sun, Z. Lin, J.-H. Lai, W. Zeng, and W.-S. Zheng, "Apanet: Auto-path aggregation for future instance segmentation prediction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3386–3403, 2021.

[52] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, "Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 14 424–14 432.

## BIOGRAPHY SECTION

**Jianwei Tang** received the bachelor's degree in computer science from Sun Yat-sen University in 2022. He is now a M.S. student in the School of Computer Science and Engineering in Sun Yat-sen University. His research interests include 3D human motion analysis and continual learning.

**Jian-Fang Hu** is now an associate professor with Sun Yat-sen University. He received the Ph.D. and B.S. degrees from the School of Mathematics, Sun Yat-Sen University, Guangzhou, China, in 2016 and 2010, respectively. His research interests include human-object interaction modeling, 3D face modeling, and RGB-D action recognition. He has published several scientific papers in the international conferences and journals including ICCV, CVPR, NeurIPS, ECCV, and IEEE TPAMI.

**Tianming Liang** received the B.S. degree from Dalian University of Technology (DUT), and the M.S. degree from Harbin Institute of Technology (HIT). He is currently working toward the Ph.D. degree with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China. His research interests include video understanding and vision-language learning.

**Xiaotong Lin** received the bachelor's degree in computer science from Sun Yat-sen University in 2022. She is now a M.S. student in the School of Computer Science and Engineering in Sun Yat-sen University. Her research interests include trajectory prediction and 3D human motion.

**Jiangxin Sun** received the bachelor's degree in computer science from Sun Yat-sen University in 2020. He is now a M.S. student in the School of Computer Science and Engineering in Sun Yat-sen University. His research interests include instance segmentation and 3D human motion.

**Dr. Wei-Shi Zheng** is now a full Professor with Sun Yat-sen University. His research interests include person/object association and activity understanding, and the related weakly supervised/unsupervised and continuous learning machine learning algorithms. He has now published more than 200 papers, including more than 150 publications in main journals (TPAMI, IJCV, TIP) and top conferences (ICCV, CVPR, SIGGRAPH, ECCV, NeurIPS). He has ever served as area chairs of ICCV, CVPR, ECCV, BMVC, NeurIPS and etc. He is associate editors/on the editorial board of IEEE-TPAMI, Artificial Intelligence Journal, Pattern Recognition. He has ever joined Microsoft Research Asia Young Faculty Visiting Programme. He is a Cheung Kong Scholar Distinguished Professor, a recipient of the Excellent Young Scientists Fund of the National Natural Science Foundation of China, and a recipient of the Royal Society-Newton Advanced Fellowship of the United Kingdom.

**Jianhuang Lai** (Senior Member, IEEE) received the Ph.D. degree in mathematics from Sun Yat-sen University, China, in 1999. In 1989, he joined Sun Yat-sen University as an Assistant Professor, where he is currently Professor with the School of Computer science and Engineering. His current research interests include the areas of computer vision, pattern recognition, and its applications. He has published over 250 scientific papers in the international journals and conferences on image processing and pattern recognition. He serves as the Deputy Director of the Image and Graphics Association of China.