MoML: Online Meta Adaptation for 3D Human Motion Prediction

Xiaoning Sun[1], Huaijiang Sun[1], Bin Li[2], Dong Wei[1]✉, Weiqing Li[1], Jianfeng Lu[1]
[1]Nanjing University of Science and Technology, China
[2]Tianjin AiForward Science and Technology Co., Ltd., China
{sunxiaoning,sunhuaijiang,csdwei}@njust.edu.cn, libin@aiforward.com

Abstract

In the academic field, the research on human motion prediction tasks mainly focuses on exploiting the observed information to forecast human movements accurately in the near future horizon. However, a significant gap appears when it comes to the application field, as current models are all trained offline, with fixed parameters that are inherently suboptimal to handle the complex yet ever-changing nature of human behaviors. To bridge this gap, in this paper, we introduce the task of online meta adaptation for human motion prediction, based on the insight that finding "smart weights" capable of swift adjustments to suit different motion contexts along the time is a key to improving predictive accuracy. We propose MoML, which ingeniously borrows the bilevel optimization spirit of model-agnostic meta-learning, to transform previous predictive mistakes into strong inductive biases to guide online adaptation. This is achieved by our MoAdapter blocks that can learn error information by facilitating efficient adaptation via a few gradient steps, which fine-tunes our meta-learned "smart" initialization produced by the generic predictor. Considering real-time requirements in practice, we further propose Fast-MoML, a more efficient variant of MoML that features a closed-form solution instead of conventional gradient update. Experimental results show that our approach can effectively bring many existing offline motion prediction models online, and improves their predictive accuracy.

1. Introduction

3D human motion prediction is aimed at forecasting a future motion sequence accurately based on the historical observation. As a core technology in computer vision and robotics, it has been widely used in applications such as human-robot collaboration [27, 40] and autonomous driving [6, 34].

Academically, current mainstream works on this task are formulated as an offline problem. They are dedicated to exploiting spatial correlations of body-joints [9, 26, 30] and temporal information of sequences [32, 38, 54], with the goal of learning a generic predictor that can handle various motions over the entire data distribution. Meanwhile, they only model the prescribed *observation-target* window with static data, with samples less than one second. However, real motions are streaming data that dynamically arrive, with diverse movements and may keep changing continuously over time, e.g., leading to concept drift [41]. Existing offline-trained predictors use frozen parameters to tackle the complexity and variability of human motions, which is inherently suboptimal for real-world applications.

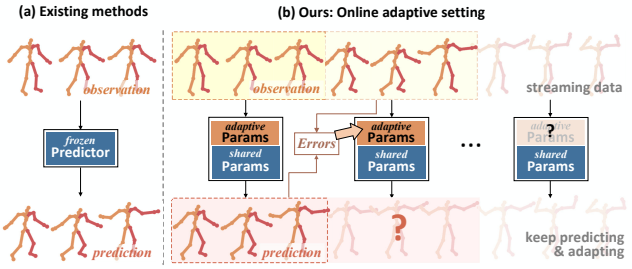

Figure 1. Existing methods predicts a short future given the observation, with frozen parameters handling all scenarios. We present a new paradigm for this task, in which: (1) motions are dynamically arriving as streaming data, instead of previous static setting; (2) we propose online meta adaptation approach, with adaptive parameters fitting the ever-changing nature of human motions; (3) as new data appears, we exploit the information in previous predictive mistakes as the driving force for online adaptation.

To bridge this gap, we draw inspiration from the human mindsets. Humans have derived basic predictive ability through their growth and learning, while making mistakes is still inevitable. A remarkable attribute of human intelligence is the ability to quickly adjust his/her thinking to suit the new situation and avoid making further mistakes during prediction. In contrast, for existing deep learning-based motion predictors, the dynamic relations between model parameters and testing errors are never discussed.

In this paper, we introduce the task of online meta adaptation for human motion prediction, based on the insight that finding "smart" weights capable of swift parameter ad-

justments along the time can effectively benefits the predictive accuracy during inference. We propose an online meta adaptation approach named MoML, which transforms the recently-produced predictive mistakes into strong inductive biases as the driving force for online adaptation, to ensure a closer alignment with the temporary context and thereby improved prediction performance then. We customize the bilevel optimization spirit of model-agnostic meta-learning (MAML) algorithm [11] to train with inner loops and outer loops. The inner loops conduct a few gradient steps to learn the error information, which constrain parameters to optimize into a *context-specific* configuration w.r.t. the temporary motion status, while the outer loops enable predictors with a generic parameter setting over various motion status via meta-updates.

To achieve this, we design MoAdapter blocks that serve as the adaptive parameters to realize online adaptation. During meta-training, we introduce the notion of temporary prediction loss, which reflects the predictive errors that just happened, and guides the learning of inner loops by supervising MoAdapters towards the optimal parameters under the temporary context. Our meta-loss, however, is a general prediction loss used to optimize the "smart" initialization over the entire motion distribution as a generic predictor in outer loops. Therefore, when performing online adaptation along the direction of time, we only update MoAdapters based on each temporary prediction loss and keep the generic parameters fixed, which exactly emulates the *flexible adaptability* and the *basic predictability* possessed in human thinking. The separation of adaptive and generic parameters in MoML allows us to discard the entire model update strategy in vanilla MAML. This ensures stable training as well as avoiding excessive time cost for adaptation during inference. Additionally, considering the real-time requirements for human motion prediction in practice, we further propose a more efficient variant named Fast-MoML, which features only a one-layer motion embedding as MoAdapter. Rather than performing gradient update, it conducts adaptation with the direct calculation of a closed-form solution, and appears less time-consuming.

In summary, our contributions are as follows:

- We are the first to address online adaptive human motion prediction, which introduces recent predictive mistakes as the driving force for adaptation along the time, to suit the ever-changing motion contexts.
- We propose MoML, an online meta adaptation approach that utilizes MoAdapters to capture error information for adaptation towards context-specific weights, by operating a few gradient steps from the meta-learned initialization. Fast-MoML is further developed for efficient adaptation.
- We empirically show that our approach can bring many existing offline-trained predictors online, and help constantly yield improved prediction performance.

Notably, so far, meta-learning has only been employed in a few studies on human motion prediction tasks. They are mainly aimed at few-shot learning for novel/unseen motion categories [10, 14, 55], which uses small samples from certain unseen category to adjust parameters for this specific category. Essentially, all of them are still under the offline setting to tackle the static distributional discrepancy. *Orthogonal* to them, we argue that cultivating the adaptability of predictors (1) in the time direction and (2) with an online manner for streaming data could bring a fresh perspective to improve both *predictive accuracy* and *model practicability*.

2. Related Work

2.1. Human Motion Prediction

Current mainstream academical research on human motion prediction primarily focuses on two aspects: capturing spatial correlations of body-joints and extracting temporal dependencies of motion sequences, to predict a short, near future based on observations. Concretely, RNNs [7, 21], LSTMs [12, 32], GRUs [38, 39], or Discrete Cosine Transformation (DCT) [30, 31] are employed to learn temporal information. Recently, by exploring the natural graph structure in human body, GNN-based [25, 26] and GCN-based [9, 28–31, 42, 58] are proposed to depict spatial correlations. Meanwhile, recent architectures like Transformer [1, 5, 48, 52] and MLP-Mixer [4, 49] are also involved. Additionally, [53] presents a motion prediction network that is theoretically equivariant under Euclidean transformations and can achieve state-of-the-art performance. The new [52] designs auxiliary denoising branch and masking prediction branch to assist the main prediction, to exploit spatio-temporal dependencies more comprehensively. However, all these models are trained offline. When applied to real world where motions are of an ever-changing nature, it is neither possible nor optimal for the fixed parameters to make accurate predictions constantly. On the other hand, these works are all restricted to the static data within the short prescribed research window, and never consider the dynamic arrival of streaming motion data. Although [43] takes the latter issue into account, the offline-trained manner still hinders its ability to fully exploit error patterns during inference. Motivated by the above, we introduce the online meta adaptation paradigm for human motion prediction, to take a step from the academic field towards applications.

2.2. Meta-Learning

Known as learning-to-learn, meta-learning is used to realize quick model adaptation to novel data or tasks. Our work borrows the bilevel optimization spirit from model-agnostic meta-learning (MAML) [11], one of the most representative and influential algorithms in meta-learning, which consists of inner loops and outer loops of training process, to ob-

tain a meta-learner that can learn task-specific parameters via a few gradient updates. [35] proposes Reptile, a simplified version of MAML by replacing second-order gradient calculation with first-order.

Online Meta Adaptation. Meta-learning for online adaptation has been discussed in various time series-related tasks in computer vision, such as human mesh recovery in videos [13], video depth estimation [24, 56, 57], video object segmentation [51] and video semantic segmentation [36, 46]. The adaptation in these works focuses more on the domain shift problem caused by the distributional discrepancy between training data and testing data (i.e., between different sequences). However, our adaptation focuses more on an *orthogonal* perspective, motivated by the inherent ever-changing nature of 3D human motion in the time direction (along the sequence). In other words, we address concept drift, rather than domain shift.

Meta-Learning in Human Motion Prediction. To date, there are only a few works in this task concerning meta-learning, which are mainly designed for few-shot learning over unseen motion categories [10, 14, 55], wherein [14] employs gradient-based, [55] memory-based and [10] graph-based meta-learning paradigms. To the best of our knowledge, the remaining two works [8, 33] aim to handle out-of-distribution problems caused by new/unseen human subjects with unique properties, such as motion style, rhythm or personal preferences. All of them are different from our setting that realizes online meta adaptation along the time to better fit the various motion contexts.

3. Method

In this section, we provide formal descriptions of human motion prediction under online meta adaptation setting, and introduce the working mechanism of our proposed MoML.

3.1. Problem Formulation & Overview

Preliminary. Currently, mainstream human motion prediction works (like [26, 29, 30]) have formed a basic routine as follows. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]$ be the observed motion sequence, the goal is to predict the future motion sequence $\hat{\mathbf{Y}} = [\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \cdots, \hat{\mathbf{y}}_T]$ as accurately as possible towards the target $\mathbf{Y}$, with each human pose $\mathbf{x}_i$, $\hat{\mathbf{y}}_i$ or $\mathbf{y}_i \in \mathbb{R}^{J\times 3}$ denoted by 3D body joint position coordinate. J is for joint number. N and T mean the frame numbers of observation and prediction target, respectively.

Online Meta Adaptation Setup. As we are aimed at addressing online meta adaptation along the time, the above prescribed research time window $(N+T)$ is not fully applicable. Here we introduce our formulation. We define the conventional goal of predicting T frames based on the observed N frames as a sub-task $\mathcal{S}_s$, which involves the (*observed, target*) motion pair $(\mathbf{X}, \mathbf{Y})_s$, i.e., $(\mathbf{x}_{1:N}, \mathbf{y}_{1:T})_s$. To realize *online* motion prediction over long-range horizon, we need to implement all these sub-tasks that are stacked along the time, i.e., $\mathcal{S} = [\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_s, \cdots]$. Every sub-task is intended for predicting the next T poses compared to its adjacent previous sub-task.

Our ultimate goal, i.e., *online meta adaptation* for motion prediction, is to learn from the mistakes made in previous sub-task $\mathcal{S}_{s-1}$, using this error information to adapt model parameters for closer alignment with the temporary context, and therefore improve the prediction performance of current sub-task $\mathcal{S}_s$. Formally, in our setting, each task is defined as $\mathcal{T}_\tau = \{\mathcal{D}^{spt}, \mathcal{D}^{qry}, \mathcal{L}^{tmp}\}_\tau$ that contains support data, query data and our temporary prediction loss. We draw adjacent sub-task pair $(\mathcal{S}_{s-1}, \mathcal{S}_s)$ from $\mathcal{S}$, with the former $\mathcal{S}_{s-1}$ as $\mathcal{D}^{spt}$ and the latter $\mathcal{S}_s$ as $\mathcal{D}^{qry}$. Specifically, $\mathcal{D}^{spt}_\tau$ is used to learn the error information guided by $\mathcal{L}^{tmp}_\tau$, so that model parameters can be adapted to better fit the temporary context τ; the prediction performance of the updated parameters is then evaluated by $\mathcal{D}^{qry}_\tau$ to ensure the effectiveness of adaptation towards this context (i.e., task).

During training, we aim to find the "smart" initialization that close to every task-specific parameter configuration in the parameter space. Therefore, during inference where each $\mathcal{S}_s$ is executed one by one for online motion prediction, the meta-learned initialization can conduct quick adaptation according to the context it confronts, and thereby solving the distributional changing in the time direction.

MoML Overview. Our approach of MoML (online Motion Meta adaptation) consists of two components: the adaptive parameter θ that keeps adjusting to fit each temporary context over time, and the generic parameter ϕ shared across all prediction scenarios. The consecutive adaptations along the prediction process can be regarded as performing swift fine-tuning of θ based on recent predictive mistake under ϕ initialization. To achieve this, we propose MoAdapters as θ, and integrate them into the model hidden layers. By learning the temporary error information, they adapt to θ_τ for context τ as *context-specific* weights, and therefore enable improved predictive accuracy. Our generic ϕ can be constructed with many existing end-to-end motion predictors, in the seek of a smart initialization that determines where is the best starting point for adaptive operation for every temporary context. We visualize the our overview in Figure 2.

3.2. MoAdapter

To accommodate the ever-changing nature of human motion, we design MoAdapters to realize online adaptation to fit diverse motion contexts, which essentially integrate the recent error information for the subsequent prediction rectification. We propose two types of MoAdapter shown in Figure 3. The first type *FC-MoAdapter* is modified from the adapter architectures in transfer learning works [15, 17] in NLP, which can produce the adapter representation at each insertion layer l. Given the input of MoAdapter at this layer

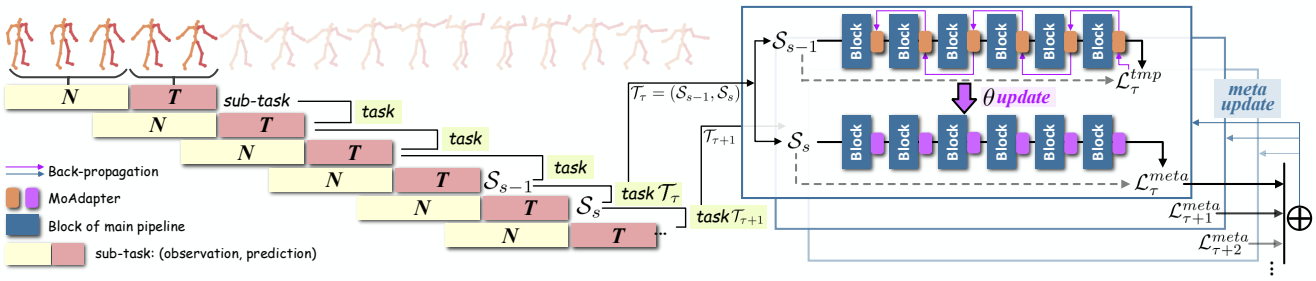

Figure 2. A sub-task is defined as predicting T frames based on N frames. To keep predicting along the time, we perform the subsequent sub-tasks, with each T frames further. In our setting, every adjacent sub-task pair is a task $\mathcal{T}_\tau = (\mathcal{S}_{s-1}, \mathcal{S}_s)$ with a temporary context τ. To drive MoAdapters (i.e., θ) to fit context τ, we feed $\mathcal{S}_{s-1}$ into the model, and the produced temporary prediction loss $\mathcal{L}_\tau^{tmp}$ contains error information that depicts how the model deviates from current context. Using it to optimize MoAdapters θ towards the context-specific θ_τ forms our inner loops. With the updated MoAdapters, the current model can be evaluated by feeding $\mathcal{S}_s$ and calculating the prediction loss $\mathcal{L}_\tau^{meta}$. By back-propagating multiple contexts of $\mathcal{L}^{meta}$ over main pipeline blocks (i.e., ϕ), we can ensure the shared parameter to obtain generic weights for all scenarios that serves as the optimal starting points for different adaptation operations. This forms our outer loops.

as $\mathbf{H}^l \in \mathbb{R}^{d_s \times d_t}$, with d_s and d_t the spatial and temporal dimensions, respectively, the adapter representation $\mathbf{Z}^l$ can be expressed as:

$$\mathbf{Z}^l = \mathbf{W}_2^l(\sigma(\mathbf{W}_1^l \mathsf{LN}(\mathbf{H}^l))) + \mathbf{H}^l, \tag{1}$$

where σ is the activation function that is set to $gelu(\cdot)$ [16], and $\mathsf{LN}(\cdot)$ indicates layer normalization [2]. We regard $\mathbf{W}_1^l$ and $\mathbf{W}_2^l$ as the adaptive parameters of each MoAdapter. If L MoAdapters are utilized totally, then $\theta = \{\mathbf{W}_1, \mathbf{W}_2\}_{1:L}$. Meanwhile, we design *GC-MoAdapter* as the second type that specially considers the graph structure of human motion data, which employs the graph convolution operation [23, 45] among the explicit body-joints or the latent body-joint features. Similarly, given the MoAdapter input as $\mathbf{H}^l \in \mathbb{R}^{d_s \times d_t}$, the adapter representation $\mathbf{Z}^l$ can be expressed as:

$$\mathbf{Z}^l = \mathbf{W}_3^l(\mathsf{GraphConv}(\mathsf{LN}(\mathbf{H}^l))) + \mathbf{H}^l. \tag{2}$$

Particularly, $\mathsf{GraphConv}(\cdot)$ learns the graph connectivity by modeling a fully-connected graph with d_s nodes, with the trainable weighted adjacency matrix $\mathbf{A}^l \in \mathbb{R}^{d_s \times d_s}$. Taking $\mathbf{Z}_{gc\_in}^l \in \mathbb{R}^{d_s \times d_t}$ as input, $\mathsf{GraphConv}(\cdot)$ outputs:

$$\mathbf{Z}_{gc\_out}^l = \varphi(\mathbf{A}^l \mathbf{Z}_{gc\_in}^l \mathbf{W}_{gc}^l), \tag{3}$$

where the activation function φ is $tanh(\cdot)$ attached with batch normalization [19], and the trainable weights $\mathbf{W}_{gc}^l \in \mathbb{R}^{d_t \times d_t'}$. We define $\theta = \{\mathbf{A}, \mathbf{W}_{gc}, \mathbf{W}_3\}_{1:L}$ if L MoAdapters participate in online adaptation. The separation of θ from the generic ϕ allows for adjustments over limited parameters w.r.t. different contexts instead of updating the entire model, which can ensure stable training and efficient online adaptation during inference.

3.3. Meta-optimization

To enable online adaptability to suit every motion context along the time, we should learn a model that can learn

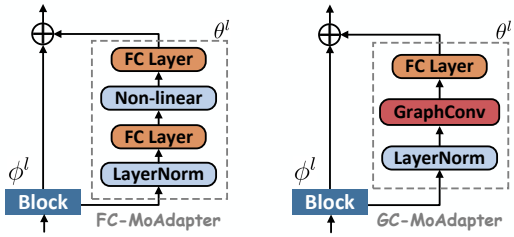

Figure 3. Architectures of the two proposed MoAdapters. Either of them is integrated into the main pipeline by attaching at the end of each network block at layer l with residual connections.

MoAdapters guided by temporary error information, i.e., learn-to-learn. We customize the bilevel optimization spirit in model-agnostic meta-learning (MAML), where the optimal θ for context τ (i.e., θ_τ^*) is obtained by MoAdapter adaptation steps, and the generic parameters ϕ shared across all scenarios are learned via meta-update. Recall our task definition $\mathcal{T}_\tau = \{\mathcal{D}^{spt}, \mathcal{D}^{qry}, \mathcal{L}^{tmp}\}_\tau = \{\mathcal{S}_{s-1}, \mathcal{S}_s, \mathcal{L}_\tau^{tmp}\}$ in Section 3.1. During the inner loops, θ is optimized by implementing sub-task $\mathcal{S}_{s-1}$ via the entire model parameterized as $\{\theta, \phi\}$, and conducting back-propagation on $\mathcal{L}_\tau^{tmp}(\theta, \phi; \mathcal{S}_{s-1})$. To be specific, the model predicts $\hat{\mathbf{Y}}_{s-1}$ based on the observed $\mathbf{X}_{s-1}$, and calculate the predictive mistake

$$\begin{aligned}\mathcal{L}_\tau^{tmp}(\theta, \phi; \mathcal{S}_{s-1}) &= \|\hat{\mathbf{Y}}_{s-1} - \mathbf{Y}_{s-1}\|_F^2 \\ &= \frac{1}{T}\sum_{t=1}^{T} \|\hat{\mathbf{y}}_{s-1,t} - \mathbf{y}_{s-1,t}\|_2^2\end{aligned} \tag{4}$$

as temporary prediction loss to optimize θ, where $\mathbf{Y}_{s-1} \in \mathbb{R}^{J \times 3 \times T}$ is the prediction target of sub-task $\mathcal{S}_{s-1}$ corresponding to the context τ, and $\mathbf{y}_{s-1,t}$ denotes the t-th frame of the entire target sequence. One or few times of such optimization steps can be operated to produce the optimal θ_τ^*.

Algorithm 1: MoML training procedure

Require: Meta-dataset $\mathcal{D}^{tr}$
Require: Hyper-parameters α, β, γ
Randomly initialize θ and ϕ
while *max interation* **do**
  Sample batch of tasks $\mathcal{T}_\tau$ composed of adjacent sub-task pair $(\mathcal{S}_{s-1}, \mathcal{S}_s)$, where $\forall \mathcal{S}_s \sim \mathcal{D}^{tr}$
  for *all* $\mathcal{T}_\tau$ **do**
    Initialize $\theta_\tau = \theta$
    for *number of adaptation steps* **do**
      Compute temporary prediction loss $\mathcal{L}_\tau^{tmp}(\theta, \phi; \mathcal{S}_{s-1})$ using Eq (4)
      Update MoAdapters:
      $\theta_\tau \leftarrow \theta_\tau - \alpha \nabla_{\theta_\tau} \mathcal{L}_\tau^{tmp}(\theta_\tau, \phi; \mathcal{S}_{s-1})$
    end
    Compute $\mathcal{L}_\tau^{meta}(\theta_\tau, \phi; \mathcal{S}_s)$ using Eq (5)
  end
  Update shared parameters ϕ by performing:
  $\phi \leftarrow \phi - \beta \nabla_\phi \sum_\tau \mathcal{L}_\tau^{meta}(\theta_\tau, \phi; \mathcal{S}_s)$
  over all the contexts in this batch
  $\beta \leftarrow \gamma \cdot \beta$
end

During outer loops, we implement sub-task $\mathcal{S}_s$ that predicts $\hat{\mathbf{Y}}_s$ given observed $\mathbf{X}_s$, and optimize ϕ guided by:

$$\mathcal{L}_\tau^{meta}(\phi; \mathcal{S}_s) = \|\hat{\mathbf{Y}}_s - \mathbf{Y}_s\|_F^2 = \frac{1}{T}\sum_{t=1}^{T} \|\hat{\mathbf{y}}_{s,t} - \mathbf{y}_{s,t}\|_2^2, \quad (5)$$

with the adapted θ_τ fixed. The meta-update is performed across different $\mathcal{L}_\tau^{meta}$ for multiple motion contexts, to obtain an optimal starting point ϕ^* that can be quickly adapted to all prediction scenarios. The formal training procedure is shown in Algorithm 1, and the formulation of MoML can be expressed as:

$$\begin{aligned} \phi^* &= \arg\min_{\phi} \sum\nolimits_\tau \mathcal{L}_\tau^{meta}(\theta_\tau^*, \phi; \mathcal{S}_s) \\ s.t. \quad \theta_\tau^* &= \arg\min_{\theta} \mathcal{L}_\tau^{tmp}(\theta, \phi; \mathcal{S}_{s-1}) \end{aligned} \quad (6)$$

As a result, with the meta-learned ϕ^* as the generic initialization, we use the temporary prediction loss over the former $\mathcal{S}_{s-1}$ in every task, to help θ transit into θ_τ^* to suit its context, and improve the performance of the latter $\mathcal{S}_s$. Note that, both of the two losses are essentially prediction losses, but the inner $\mathcal{L}_\tau^{tmp}$ drives θ towards specific context, and the outer $\mathcal{L}_\tau^{meta}$ serves as meta-loss to find the optimal initialization for the entire data distribution.

3.4. Fast-MoML

Apart from the above gradient-based optimization framework, considering the real-time requirements of online motion prediction, we present a more efficient version of MoML, named Fast-MoML. We specify that the adaptive parameters θ only exist as the last layer of the entire model, with all the other parameters as ϕ, which means that only the last layer is involved in inner loops. Motivated by [3, 50], we discover that such last-layer optimization with MSE loss can be regarded as a simple ridge regression problem with an analytic solution instead of the complex and time-consuming gradient-based optimization. Suppose an L-layer network denoted as ϕ_{net}, which outputs $\mathbf{H}_{s-1}^L$ during the sub-task $\mathcal{S}_{s-1}$. With $\theta = \{\mathbf{W}^L\}$, the solution of inner loop optimization can be expressed as:

$$\begin{aligned} \mathbf{W}_\tau^{L^*} &= \arg\min_{\mathbf{W}} \|\mathbf{H}_{s-1}^L \mathbf{W} - \mathbf{Y}_{s-1}\|^2 + \lambda \|\mathbf{W}\|^2 \\ &= ((\mathbf{H}_{s-1}^L)^T \mathbf{H}_{s-1}^L + \lambda \mathbf{I})^{-1} (\mathbf{H}_{s-1}^L)^T \mathbf{Y}_{s-1} \end{aligned} \quad (7)$$

for each adaptation. When θ is adapted to θ_τ^* as $\mathbf{W}_\tau^{L^*}$, the prediction process in the subsequent $\mathcal{S}_s$ can be improved as $\hat{\mathbf{Y}}_s = \mathbf{H}_s^L \mathbf{W}_\tau^{L^*}$. λ serves as the regularization parameter in ridge regression, which is trainable and belongs to ϕ that participates meta-update, i.e., $\phi = \{\phi_{net}, \lambda\}$. Similar to [3], we also involve a bias term by appending a scalar 1 to $\mathbf{H}^L$ during the calculation. As the above closed-form solution is differentiable, gradient-based optimization only happens to ϕ. In inference, we can directly conduct matrix multiplication to realize fast adaptation for every context over time.

4. Experiments

In this section, we evaluate our MoML approach over the task of human motion prediction with online meta adaptation. Following [26, 30], we train on Human3.6M [20], CMU-Mocap and 3DPW [47] datasets. The experiments are conducted by modifying three baselines into online meta adaptive setting, and we provide numerical results, visualizations, with ablation studies for full analysis.

4.1. Datasets

Human3.6M is the most influential and representative dataset for human motion prediction, which contains seven actors performing 15 types of actions, such as *walking*, *eating*, *smoking* and *discussion*. Motion sequences are downsampled into 25 Hz to form our motion data. Each human body is denoted by 32 body-joints pre-processed into 3D coordinates. We follow the mainstream [4, 26, 30, 32] and consider 22 of these joints. Subjects 1 (S1), S6, S7-S9 are for training, S5 and S11 are for testing and validation.

CMU-Mocap denotes each body by 28 joints as 3D coordinates. Following [26, 30], we leave 25 joints for experiments and use the same protocol to split training and testing sets. Similar to them, we include 8 categories: *basketball*, *basketball signal*, *directing traffic*, *jumping*, *running*, *soccer*, *walking* and *washing window*.

	walking				eating				smoking				discussion				directions			
millisecond (ms)	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Res. sup [32]	29.4	50.8	76.0	81.5	16.8	30.6	56.9	68.7	23.0	42.6	70.1	82.7	32.9	61.2	90.9	96.2	35.4	57.3	76.3	87.7
DMGNN [25]	17.3	30.7	54.6	65.2	11.0	21.4	36.2	43.9	9.0	17.6	32.1	40.3	17.3	34.8	61.0	69.8	13.1	24.6	64.7	81.9
MSR [9]	12.2	22.7	38.6	45.2	8.4	17.1	33.0	40.4	8.0	16.3	31.3	38.2	12.0	26.8	57.1	69.7	8.6	19.7	43.3	53.8
LTD [30]	12.3	23.0	39.8	46.1	8.4	16.9	33.2	40.7	8.0	16.2	31.9	38.9	12.5	27.4	58.5	71.7	9.0	19.9	43.4	53.7
LTD-FC	10.9	**19.5**	37.3	**42.2**	7.8	15.7	**31.6**	38.4	7.2	**14.2**	**28.5**	36.9	11.2	25.0	55.8	68.9	8.2	**17.5**	41.7	51.0
LTD-GC	**10.5**	**19.5**	**36.9**	42.7	**7.0**	**15.1**	**31.6**	**38.0**	**6.9**	14.5	29.0	**36.6**	**11.0**	**24.5**	**55.4**	**68.7**	**8.0**	17.6	**39.8**	**50.6**
MotionMixer [4]	10.8	22.4	36.5	42.4	7.7	14.0	27.3	36.1	**7.1**	14.0	29.1	36.8	10.2	22.5	51.0	64.1	8.3	18.1	43.8	53.4
MotionMixer-FC	9.9	**20.7**	**34.0**	**40.1**	6.8	12.8	**25.4**	**34.7**	7.4	14.1	**27.3**	**34.5**	9.4	**19.7**	**48.6**	**61.4**	**7.8**	**17.2**	**41.5**	**51.3**
MotionMixer-GC	**9.6**	21.4	34.7	41.0	**6.2**	**12.5**	26.1	35.4	7.5	**13.8**	27.9	34.6	**9.3**	20.8	49.5	62.5	8.5	17.6	42.6	51.7
SPGSN [26]	10.1	19.4	34.8	41.5	**7.1**	14.9	30.5	37.9	6.7	13.8	28.0	34.6	10.4	23.8	53.6	67.1	7.4	17.2	39.8	50.3
SPGSN-FC	**9.2**	18.1	32.9	40.0	6.5	**14.1**	**28.1**	36.8	6.7	12.7	**26.1**	**32.9**	**9.1**	**21.7**	**51.0**	65.2	7.3	16.3	**37.0**	**48.0**
SPGSN-GC	9.3	**17.5**	**32.2**	**39.8**	7.3	14.3	28.2	**36.4**	**6.3**	**12.5**	26.3	33.6	9.3	21.9	51.9	**63.9**	**7.1**	**16.2**	37.4	47.2

Table 1. Comparisons of MPJPE errors of 5 typical activities in Human3.6M between baselines without/with our MoML approach. The suffix of FC or GC indicates integrating FC-MoAdapters or GC-MoAdapters, respectively. Lower errors are highlighted in **bold**, where our designs gain improvement in most cases. As we perform each sub-task one by one along the time to predict with streaming motion data, each error of ours is calculated by averaging errors of all sub-tasks at the corresponding timestamp.

3DPW is a more challenging dataset containing 51k frames of both indoor and outdoor human activities. Each body is represented by 3D coordinate of 23 joints, and the frame rate is 30 Hz. We follow [26, 30] to split training, testing and validation sets as official suggestion.

Evaluation Metric. Following standard human motion prediction works [4, 9, 26, 29, 30], we evaluate our approach by Mean Per Joint Position Error (MPJPE) on 3D human joint coordinates. It calculates the average L_2-norm on the discrepancies between the prediction and corresponding ground truth over all body-joints.

4.2. Baselines

We provide comparisons with mainstream Res. sup [32], DMGNN [25], LTD [30], MSR [9], SPGSN [26] and MotionMixer [4]. As our goal is to use MoML to bring offline-trained baselines online, we choose the following baselines for our online meta adaptive modification.

LTD [30] is a typical (also the first) GCN-based model for human motion prediction, which designs stacked graph convolutional blocks with skip connections to model the spatial correlations of body-joints. Discrete Cosine Transformation (DCT) is employed to extract temporal information.

SPGSN [26] is a reformative graph-based model with cascaded graph scattering blocks to capture fine representation of both spatial and spectrum features. Within each block, human body are divided into separated body-parts to conduct their respective graph modeling, which are then fused with the full-body modeling.

MotionMixer [4] is an efficient motion predictor features the recent MLP-Mixer architecture [44]. It customizes spatial mixing blocks and temporal mixing blocks, and stacks them with squeeze-and-excitation [18] enhanced.

Implementation Details. Our experiments are conducted under Pytorch [37] framework with Adam optimizer [22] on a single NVIDIA RTX 3090. We train all three baselines with MoML for 50 epochs. As the network architecture and other experimental details are different for each model, we leave them in supplementary for further introduce.

Fairness Discussion. Motion data is very long, containing multiple sequences with hundreds to thousands of frames. Existing offline-trained predictors are evaluated by short-term and long-term prediction, which stand for predicting 400ms and 1000ms of motions, respectively. They cut short samples from the long sequence data regardless of their streaming structure, and test on every single sample. For our online adaptive setting, we treat every 400ms-prediction as a sub-task and predict with adaptive parameters to suit each motion context along the time. We evaluate the effectiveness of MoML by averaging the predictive errors of all sub-tasks at certain timestamps. In other words, we all go through the entire dataset, so the comparison is fair.

4.3. Results

Human3.6M. Table 1 provides the prediction performance of 5 typical activities in Human3.6M, which shows comparison of MPJPE errors between baselines without/with MoML approach. Values in bold indicate the lowest error among baseline and two types of baseline+MoML. Results of the remaining activities in Human3.6M are shown in Table 2. Both designs of MoML improve the predictive accuracy in most cases. We also observe that failures may happen to motions with relatively static status, such as *sitting* and *waiting* where the contexts barely changed, and updating parameters in this scenario may tend to be unnecessary.

In Figure 4, we present visualized comparisons of two cases on *discussion* and *walking dog*. We draw motion contents in eight seconds, where significant errors produced by baselines are highlighted with red boxes, and the benefits brought by our MoML are highlighted with green. For example, in the bottom sub-figure, the person is walking fast and appears to be dragged by a dog from the side. The base-

	greeting				phoning				posing				purchases				sitting			
millisecond (ms)	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
LTD [30]	18.7	38.7	77.7	93.4	**10.2**	21.0	42.5	52.3	13.7	29.9	66.6	84.1	15.6	32.8	65.7	79.3	**10.6**	21.9	**46.3**	**57.9**
LTD-FC	18.0	36.9	75.5	91.1	10.8	21.2	**42.1**	51.4	14.0	28.5	**65.5**	84.3	15.2	31.2	64.0	77.1	11.0	22.6	46.7	59.0
LTD-GC	**16.9**	**36.1**	**74.9**	**90.6**	10.6	**20.4**	42.5	**51.2**	**13.3**	**28.4**	65.7	**81.8**	**15.1**	**30.6**	**63.7**	**76.8**	10.9	**21.0**	46.5	58.3
MotionMixer [4]	12.8	33.4	62.3	82.2	10.0	20.1	37.4	51.1	**11.7**	23.3	62.4	79.5	14.6	31.3	62.8	76.1	**10.0**	20.9	43.7	54.5
MotionMixer-FC	**12.4**	**33.0**	**61.1**	**80.8**	9.9	**19.4**	36.6	**48.9**	12.2	**23.0**	**60.8**	**77.5**	**13.8**	**30.2**	**60.4**	75.7	10.2	**20.7**	**43.3**	**53.8**
MotionMixer-GC	13.6	33.6	61.7	81.3	**9.7**	19.9	**36.1**	49.4	12.0	23.3	61.3	77.7	**13.8**	30.9	60.6	**75.4**	10.6	21.0	43.6	54.2
SPGSN [26]	14.6	32.6	70.6	86.4	8.7	18.3	38.7	48.5	10.7	25.3	59.9	76.5	12.8	28.6	61.0	74.4	**9.3**	**19.4**	**42.3**	53.6
SPGSN-FC	13.8	**31.1**	69.6	84.1	**8.5**	**17.1**	**37.8**	**48.0**	10.8	**24.4**	**57.1**	74.7	**12.2**	**27.8**	**58.8**	**72.3**	10.2	21.1	42.3	**53.2**
SPGSN-GC	**12.6**	31.8	**68.9**	**84.0**	8.7	18.0	38.9	49.1	**10.2**	25.2	57.8	**74.2**	12.9	28.2	59.5	72.8	9.8	20.8	42.5	53.8

	sittingdown				takingphoto				waiting				walkingdog				walkingtogether			
millisecond (ms)	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
LTD [30]	16.1	31.1	61.5	75.5	9.9	20.9	45.0	56.6	11.4	24.0	50.1	61.5	23.4	46.2	83.5	96.0	10.5	21.0	38.5	45.2
LTD-FC	15.8	30.7	60.6	74.1	8.6	19.0	44.3	54.8	11.8	24.1	**49.2**	59.1	21.9	43.4	80.7	94.1	9.6	19.7	36.7	43.0
LTD-GC	**15.5**	**30.3**	**60.4**	**73.8**	**8.2**	**18.7**	**42.8**	**53.2**	**11.2**	**23.7**	50.0	**58.2**	**20.5**	**43.0**	**79.8**	**93.6**	**8.9**	**19.2**	**36.1**	**42.4**
MotionMixer [4]	12.0	31.4	61.4	74.5	9.0	18.9	41.0	51.6	**10.2**	21.1	45.2	56.4	20.5	42.8	75.6	87.8	10.5	20.6	38.7	43.5
MotionMixer-FC	**11.4**	**30.9**	**60.2**	**72.6**	**7.8**	**18.0**	**38.5**	49.2	10.3	**20.2**	**44.1**	**55.2**	**18.4**	**40.1**	**73.0**	**84.2**	**8.9**	**18.3**	**36.2**	**40.9**
MotionMixer-GC	12.8	31.2	60.8	72.7	8.1	18.3	38.8	**48.8**	11.1	20.6	45.0	56.9	19.7	41.1	73.3	85.0	9.6	18.7	36.9	41.4
SPGSN [26]	14.2	27.7	56.8	70.7	8.8	18.9	41.5	52.7	**9.2**	19.8	43.1	54.1	18.2	37.3	71.3	84.2	8.9	18.2	33.8	40.9
SPGSN-FC	**14.1**	**26.9**	**55.2**	68.3	7.7	18.2	39.2	50.6	9.5	**19.4**	**42.2**	53.3	**16.8**	35.8	68.5	82.0	**8.1**	16.9	32.0	38.9
SPGSN-GC	**14.1**	27.1	55.4	**68.2**	**7.5**	**17.8**	**38.7**	**48.9**	9.6	20.5	42.4	**52.8**	17.1	**36.0**	**68.0**	**81.3**	8.2	**16.5**	**31.6**	**38.5**

Table 2. Comparisons of MPJPE errors of remaining activities in Human3.6M between baselines without/with our MoML approach.

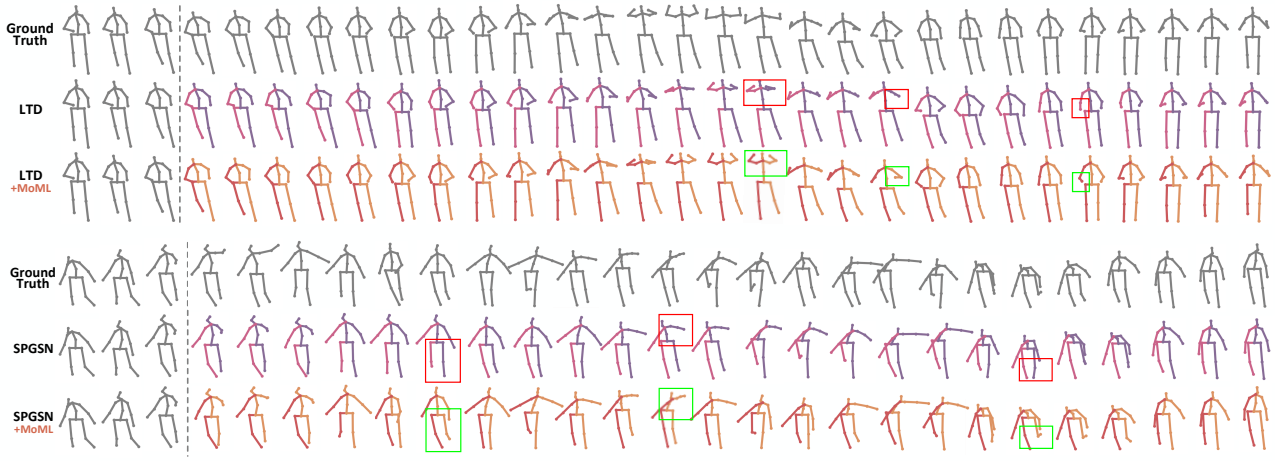

Figure 4. Two visualized cases on *discussion* (top) and *walking dog* (bottom). We choose LTD [30] and SPGSN [26] as baselines for comparison. In each case, we draw motion contents in eight seconds. With the online adaptation operation by MoML, we can produce predictions with higher accuracy, indicated by the improved performance marked in green boxes.

line fails to predict accurately as the motions are complex and quickly changing, while with our MoML, although we still cannot produce the exact true poses, many predictions have already obtained the right motion tendencies.

CMU and 3DPW. We also provide experimental results on this two datasets in Table 3, where the average performance of MoML-trained models also surpasses baselines. As the results of MotionMixer [4] are not available here, we only leave LTD [30] and SPGSN [26] for comparison.

Fast-MoML. Considering the practical needs of predicting motions instantly, we present the predictive errors of our Fast-MoML on Human3.6M in Figure 5, and compare them with the average performance of FC/GC-based MoML. We choose the 400ms timestamp in each sub-task as testpoint. From the figure, Fast-MoML indeed obtains a certain degree of adaptability, but the expressions of single-layer embedding are limited when faced with varied motions. Nevertheless, the time-saving property is still undeniable, and we leave the related experiments in supplementary.

4.4. Ablation Study

Performance on streaming sub-tasks. In Figure 6, we draw MPJPE error of each sub-task along the time, to evaluate the improvement brought by MoML, as well as the stability of such benefit. Notably, to let the original baselines to perform with streaming data, we just need to use these frozen models to predict every sub-task in turn and calculate the corresponding error. Both FC/GC-based MoML

millisecond (ms)	CMU-Mocap Average 80	160	320	400	3DPW Average 100	200	400
Res. Sup [32]	24.2	43.8	72.4	88.9	102.3	113	174
DMGNN [25]	14.1	24.4	45.9	56.5	17.8	37.1	70.4
MSR [9]	8.7	15.8	30.6	38.1	15.7	33.5	65.0
LTD [30]	9.9	18.0	33.5	40.9	16.3	35.6	67.5
LTD-FC	9.6	16.9	32.4	38.7	15.5	33.8	66.1
LTD-GC	**9.4**	**16.6**	**32.0**	**38.2**	**15.2**	**33.2**	**65.3**
SPGSN [26]	8.3	14.8	28.8	37.0	15.4	32.9	64.5
SPGSN-FC	8.0	**13.7**	**27.1**	**35.1**	**14.3**	30.6	**61.2**
SPGSN-GC	**7.8**	13.9	27.4	35.2	14.7	**30.5**	61.6

Table 3. Comparisons of MPJPE average errors on CMU-Mocap and 3DPW without/with our MoML approach. With our approach, the prediction performance is improved.

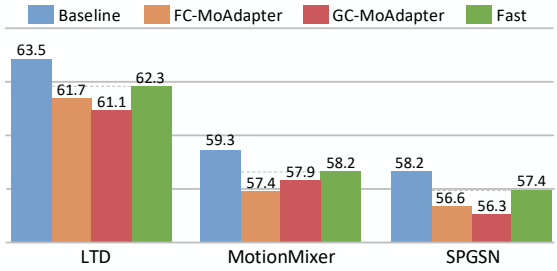

Figure 5. Comparisons of predictive errors at 400ms testpoints among two types of MoML and Fast-MoML on Human3.6M.

and Fast-MoML produce lower predictive errors, and can stably keep this improvement over a long horizon. Note that there exists no improvement for the first sub-task of prediction, as no recent errors can be used for adaptation.

MoML vs. MAML. As we conduct adaptation over selective parameters, rather than the entire model like vanilla MAML, we further investigate the performance of employing MAML in regard to our task. In Table 4, we present results of baselines with (1) *baseline+MAML*: using MAML to train the original baselines and update all parameters during inference; (2) *baseline-FC/GC+MAML*: integrating our FC/GC-MoAdapters into the main backbone, and operating MAML over the entire network; (3) *baseline-LL+MAML*: attaching the last-layer structure to baselines like Fast-MoML, but operating MAML over the entire network; (4) *only-LL+grad*: Using gradient-based optimization instead of closed-form solution as the inner loops of Fast-MoML.

From the table, MAML also exhibits certain superiority compared to the offline-trained manner, but there is no obvious advantage compared to our MoML, as the instability brought by large number of parameter update may limit the adaptive performance. Moreover, MAML can be cumbersome and time-consuming when applied to existing predictors, making it less suitable to handle the streaming motion data. In our supplementary, we provide more discussion concerning inference time, along with hyperparameter settings and concrete meta design for each baseline in detail.

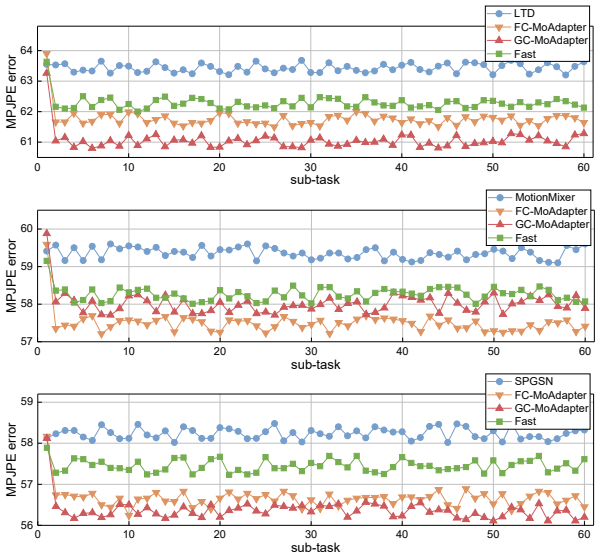

Figure 6. Performance on streaming sub-tasks on Human3.6M. We draw 60 sub-tasks of MPJPE errors at 400ms testpoints, where our three designs of MoML help improve the predictive accuracy of baselines constantly.

method	LTD [30]	MotionMixer [4]	SPGSN [26]
baseline	63.52	59.33	58.22
baseline+MAML	61.98	58.09	57.34
baseline-FC+MAML	62.51	57.83	56.59
baseline-GC+MAML	61.90	60.04	57.23
baseline-LL+MAML	62.92	58.14	57.30
only-LL+grad	62.37	58.28	57.58

Table 4. Different networks trained with vanilla MAML-based approach, and gradient-based last-layer optimization.

5. Conclusion

In this paper, we address the problem of online adaptive human motion prediction with streaming motion data. We introduce an online meta adaptation approach named MoML, which cultivates model adaptability in the time direction, to suit the inherent complexity and ever-changing nature of human behaviors. We propose two types of MoAdapters that incorporate recent error information as guidance, to perform swift parameter adjustments towards a closer alignment with recent temporary motion context. The bilevel optimization structure is customized to learn "smart" initialization as the optimal starting point for online adaptation across various contexts. Fast-MoML is further developed, featuring a last-layer motion embedding with closed-form solution for time saving. Experiments show that our MoML can bring different existing offline-trained predictors online, and constantly benefit the predictive accuracy.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China (NO. 62176125, 61772272).

References

[1] Emre Aksan, Manuel Kaufmann, Peng Cao, and Otmar Hilliges. A spatio-temporal transformer for 3d human motion prediction. In *International Conference on 3D Vision (3DV)*, pages 565–574. IEEE, 2021. 2

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[3] Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *ICLR*, 2019. 5

[4] Arij Bouazizi, Adrian Holzbock, Ulrich Kressel, Klaus Dietmayer, and Vasileios Belagiannis. Motionmixer: Mlp-based 3d human body pose forecasting. In *IJCAI*, 2022. 2, 5, 6, 7, 8

[5] Yujun Cai, Lin Huang, Yiwei Wang, Tat-Jen Cham, Jianfei Cai, Junsong Yuan, Jun Liu, Xu Yang, Yiheng Zhu, Xiaohui Shen, et al. Learning progressive joint propagation for human motion prediction. In *ECCV*, pages 226–242. Springer, 2020. 2

[6] Fanta Camara, Nicola Bellotto, Serhan Cosar, Florian Weber, Dimitris Nathanael, Matthias Althoff, Jingyuan Wu, Johannes Ruenz, André Dietrich, Gustav Markkula, et al. Pedestrian models for autonomous driving part ii: high-level models of human behavior. *IEEE Transactions on Intelligent Transportation Systems*, 22(9):5453–5472, 2020. 1

[7] Enric Corona, Albert Pumarola, Guillem Alenyà, and Francesc Moreno-Noguer. Context-aware human motion prediction. In *CVPR*, pages 6992–7001, 2020. 2

[8] Qiongjie Cui, Huaijiang Sun, Jianfeng Lu, Bin Li, and Weiqing Li. Meta-auxiliary learning for adaptive human pose prediction. In *AAAI*, 2023. 3

[9] Lingwei Dang, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. Msr-gcn: Multi-scale residual graph convolution networks for human motion prediction. In *ICCV*, pages 11467–11476, 2021. 1, 2, 6, 8

[10] Rafael Rego Drumond, Lukas Brinkmeyer, and Lars Schmidt-Thieme. Few-shot human motion prediction for heterogeneous sensors. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 551–563. Springer, 2023. 2, 3

[11] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pages 1126–1135. PMLR, 2017. 2

[12] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *ICCV*, pages 4346–4354, 2015. 2

[13] Shanyan Guan, Jingwei Xu, Yunbo Wang, Bingbing Ni, and Xiaokang Yang. Bilevel online adaptation for out-of-domain human mesh reconstruction. In *CVPR*, pages 10472–10481, 2021. 3

[14] Liang-Yan Gui, Yu-Xiong Wang, Deva Ramanan, and José MF Moura. Few-shot human motion prediction via meta-learning. In *ECCV*, pages 432–450, 2018. 2, 3

[15] Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. In *ICLR*, 2022. 3

[16] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 4

[17] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *ICML*, pages 2790–2799. PMLR, 2019. 3

[18] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, pages 7132–7141, 2018. 6

[19] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456. PMLR, 2015. 4

[20] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE TPAMI*, 36(7):1325–1339, 2013. 5

[21] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *CVPR*, pages 5308–5317, 2016. 2

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[23] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017. 4

[24] Yevhen Kuznietsov, Marc Proesmans, and Luc Van Gool. Comoda: Continuous monocular depth adaptation using past experiences. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2907–2917, 2021. 3

[25] Maosen Li, Siheng Chen, Yangheng Zhao, Ya Zhang, Yanfeng Wang, and Qi Tian. Dynamic multiscale graph neural networks for 3d skeleton based human motion prediction. In *CVPR*, pages 214–223, 2020. 2, 6, 8

[26] Maosen Li, Siheng Chen, Zijing Zhang, Lingxi Xie, Qi Tian, and Ya Zhang. Skeleton-parted graph scattering networks for 3d human motion prediction. In *ECCV*, pages 18–36. Springer, 2022. 1, 2, 3, 5, 6, 7, 8

[27] Hongyi Liu and Lihui Wang. Human motion prediction for human-robot collaboration. *Journal of Manufacturing Systems*, 44:287–294, 2017. 1

[28] Zhenguang Liu, Pengxiang Su, Shuang Wu, Xuanjing Shen, Haipeng Chen, Yanbin Hao, and Meng Wang. Motion prediction using trajectory cues. In *ICCV*, pages 13299–13308, 2021. 2

[29] Tiezheng Ma, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. Progressively generating better initial guesses towards next stages for high-quality human motion prediction. In *CVPR*, pages 6437–6446, 2022. 3, 6

[30] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Learning trajectory dependencies for human motion prediction. In *ICCV*, pages 9489–9497, 2019. 1, 2, 3, 5, 6, 7, 8

[31] Wei Mao, Miaomiao Liu, and Mathieu Salzmann. History repeats itself: Human motion prediction via motion attention. In *ECCV*, pages 474–489. Springer, 2020. 2

[32] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *CVPR*, pages 2891–2900, 2017. 1, 2, 5, 6, 8

[33] Hee-Seung Moon and Jiwon Seo. Fast user adaptation for human motion prediction in physical human–robot interaction. *IEEE Robotics and Automation Letters*, 7(1):120–127, 2021. 3

[34] Mahyar Najibi, Jingwei Ji, Yin Zhou, Charles R Qi, Xinchen Yan, Scott Ettinger, and Dragomir Anguelov. Motion inspired unsupervised perception and prediction in autonomous driving. In *ECCV*, pages 424–443. Springer, 2022. 1

[35] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018. 3

[36] Theodoros Panagiotakopoulos, Pier Luigi Dovesi, Linus Härenstam-Nielsen, and Matteo Poggi. Online domain adaptation for semantic segmentation in ever-changing conditions. In *ECCV*, pages 128–146. Springer, 2022. 3

[37] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 6

[38] Dario Pavllo, Christoph Feichtenhofer, Michael Auli, and David Grangier. Modeling human motion with quaternion-based neural networks. *IJCV*, 128(4):855–872, 2020. 1, 2

[39] Tim Salzmann, Marco Pavone, and Markus Ryll. Motron: Multimodal probabilistic human motion forecasting. In *CVPR*, pages 6457–6466, 2022. 2

[40] Alessio Sampieri, Guido Maria D'Amely di Melendugno, Andrea Avogaro, Federico Cunico, Francesco Setti, Geri Skenderi, Marco Cristani, and Fabio Galasso. Pose forecasting in industrial human-robot collaboration. In *ECCV*, pages 51–69. Springer, 2022. 1

[41] Pekka Siirtola and Juha Röning. Feature relevance analysis to explain concept drift-a case study in human activity recognition. In *Adjunct Proceedings of the 2022 ACM International Joint Conference on Pervasive and Ubiquitous Computing and the 2022 ACM International Symposium on Wearable Computers*, pages 386–391, 2022. 1

[42] Theodoros Sofianos, Alessio Sampieri, Luca Franco, and Fabio Galasso. Space-time-separable graph convolutional network for pose forecasting. In *ICCV*, pages 11209–11218, 2021. 2

[43] Xiaoning Sun, Huaijiang Sun, Bin Li, Dong Wei, Weiqing Li, and Jianfeng Lu. Defeenet: Consecutive 3d human motion prediction with deviation feedback. In *CVPR*, pages 5527–5536, 2023. 2

[44] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *NeurIPS*, 34: 24261–24272, 2021. 6

[45] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018. 4

[46] Riccardo Volpi, Pau De Jorge, Diane Larlus, and Gabriela Csurka. On the road to online adaptation for semantic image segmentation. In *CVPR*, pages 19184–19195, 2022. 3

[47] Timo Von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *ECCV*, pages 601–617, 2018. 5

[48] Dong Wei, Huaijiang Sun, Bin Li, Jianfeng Lu, Weiqing Li, Xiaoning Sun, and Shengxiang Hu. Human joint kinematics diffusion-refinement for stochastic motion prediction. In *AAAI*, pages 6110–6118, 2023. 2

[49] Dong Wei, Huaijiang Sun, Bin Li, Xiaoning Sun, Shengxiang Hu, Weiqing Li, and Jianfeng Lu. Nerm: Learning neural representations for high-framerate human motion synthesis. In *ICLR*, 2024. 2

[50] Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. Learning deep time-index models for time series forecasting. In *International Conference on Machine Learning*, pages 37217–37237. PMLR, 2023. 5

[51] Huaxin Xiao, Bingyi Kang, Yu Liu, Maojun Zhang, and Jiashi Feng. Online meta adaptation for fast video object segmentation. *IEEE TPAMI*, 42(5):1205–1217, 2019. 3

[52] Chenxin Xu, Robby T Tan, Yuhong Tan, Siheng Chen, Xinchao Wang, and Yanfeng Wang. Auxiliary tasks benefit 3d skeleton-based human motion prediction. In *ICCV*, pages 9509–9520, 2023. 2

[53] Chenxin Xu, Robby T Tan, Yuhong Tan, Siheng Chen, Yu Guang Wang, Xinchao Wang, and Yanfeng Wang. Eqmotion: Equivariant multi-agent motion prediction with invariant interaction reasoning. In *CVPR*, pages 1410–1420, 2023. 2

[54] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018. 1

[55] Chuanqi Zang, Mingtao Pei, and Yu Kong. Few-shot human motion prediction via learning novel motion dynamics. In *IJCAI*, pages 846–852, 2021. 2, 3

[56] Zhenyu Zhang, Stéphane Lathuiliere, Andrea Pilzer, Nicu Sebe, Elisa Ricci, and Jian Yang. Online adaptation through meta-learning for stereo depth estimation. *arXiv preprint arXiv:1904.08462*, 2019. 3

[57] Zhenyu Zhang, Stephane Lathuiliere, Elisa Ricci, Nicu Sebe, Yan Yan, and Jian Yang. Online depth learning against forgetting in monocular videos. In *CVPR*, pages 4494–4503, 2020. 3

[58] Chongyang Zhong, Lei Hu, Zihao Zhang, Yongjing Ye, and Shihong Xia. Spatio-temporal gating-adjacency gcn for human motion prediction. In *CVPR*, pages 6447–6456, 2022. 2