MoML: Online Meta Adaptation for 3D Human Motion Prediction

Supplementary Material

1. More Details of MoAdapters

Given the output of certain main pipeline block at the l-th layer as $\mathbf{H}^l \in \mathbb{R}^{d_s \times d_t}$, with d_s and d_t the dimensions of pipeline intermediate spatial and temporal feature. For FC-MoAdapters, the dimensions of the two FC layers compile with the temporal dimension of $\mathbf{H}^l$, expressed as $\mathbf{W}_1^l \in \mathbb{R}^{d_t \times d_t}$ and $\mathbf{W}_2^l \in \mathbb{R}^{d_t \times d_t}$. For GC-MoAdapters, we set the dimension of trainable weights $\mathbf{W}_{gc}^l$ in $\mathsf{GraphConv}(\cdot)$ to $d_t \times d_t{}'$, where $d_t{}' = 256$ is applied to all three baselines. Then the following FC layer (i.e., $\mathbf{W}_3^l$) maps it into the backbone temporal dimension d_t.

Specially, for Fast-MoML, as the last layer (the L-th layer) of the original pipeline outputs $\mathbf{H}^L \in \mathbb{R}^{d_s^{ta} \times d_t^{ta}}$ with small target dimension, directly attaching our motion embedding $\mathbf{W}^L$ behind it may not be sufficient to express the adaptive information. In this case, we modify the original last layer to let it output $\mathbf{H}^L \in \mathbb{R}^{d_s^{ta} \times d_t}$, and then use our $\mathbf{W}^L$ to map it into the final dimension $d_s^{ta} \times d_t^{ta}$.

The blocks of the three baselines are stacked as follows. LTD [7] contains 12 blocks of graph convolutional layers with residual connections, plus two individual graph convolutional layers (one at the beginning and one at the end). MotionMixer [1] contains 3 blocks of spatio-temporal mixing layers, plus two individual fully-connected layers (one at the beginning and one at the end). SPGSN [6] contains 10 blocks of graph scattering layers, with each one followed by an MLP structure. In accordance with them, we apply 12, 3 and 10 MoAdapters for each baselines, respectively.

2. Details of Loss Function

In the Eq (4) and Eq (5) of our main paper, we use $\frac{1}{T}\sum_{t=1}^{T} \|\hat{\mathbf{y}}_{s,t} - \mathbf{y}_{s,t}\|_2^2$ to depict the prediction loss of the s-th sub-task, with $\hat{\mathbf{y}}_{s,t}$ and $\mathbf{y}_{s,t}$ indicating the t-th predicted and real frame. This is only a general expression, and here we present the concrete function for each baseline.

For MoML on LTD [7] and SPGSN [6], we follow the two baselines to predict both the observed part (N frames) and target part (T frames), and formulate the loss as:

$$\mathcal{L}_{\text{(LTD/SPGSN)}} = \frac{1}{N+T} \sum_{t=-N}^{T} \|\hat{\mathbf{y}}_{s,t} - \mathbf{y}_{s,t}\|_2^2. \qquad (1)$$

As the two baselines use Discrete Cosine Transformation (DCT) to process motion data to extract temporal information, they calculate the loss over the $(N+T)$ horizon rather than T, which can bring additional signal to learn to predict DCT coefficients that represent the entire sequence. We also adopt the same strategy in line with them.

For MoML on MotionMixer [1], we follow the baseline to use the joint position displacement Δ between two adjacent frames (i.e., velocity), rather than the joint position of each frame, with the loss formulated as

$$\mathcal{L}_{\text{(MotionMixer)}} = \frac{1}{T} \sum_{t=1}^{T} \|\Delta\hat{\mathbf{y}}_{s,t} - \Delta\mathbf{y}_{s,t}\|_2^2. \qquad (2)$$

3. More Implementation Details

To train baselines with MoML, we set the batch size to 16, 50, and 16 for LTD [7], MotionMixer [1] and SPGSN [6]. The adaptive operation is conducted via 5 gradient steps by FC/GC MoAdapters. We provide comparisons on different gradient step number u in Figure 1 to further analyze the effects brought by its changes. During meta-training, the inner learning rate $\alpha = 0.01$ is for all methods. The outer learning rate β is initialized as 0.0005, 0.01 and 0.001 in align with the three baselines. For LTD and SPGSN, a 0.96 decay is performed every two epochs; for MotionMixer, the learning rate is decayed by a factor of 0.1 every 10 epochs. Fast-MoML shares the same β as the above, and no α is needed due to the closed-form solution.

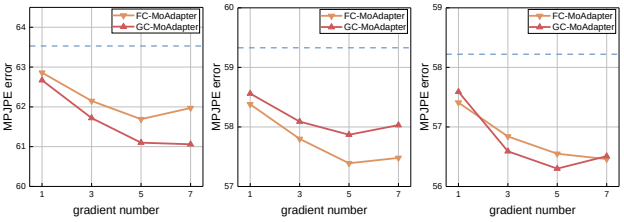

Figure 1. Comparisons on different numbers u of gradient updates on Human3.6M. Left, middle and right sub-figures show the online adaptive prediction performance modified from LTD, MotionMixer and SPGSN, respectively. Blue dashed lines indicate offline-trained performance of these baselines. Generally, $u = 5$ can help produce the lowest MPJPE errors in most cases, while the larger u would not necessarily bring more benefits.

For the ablations on vanilla MAML in our main paper Sec 4.4, the outer learning rate β is the same as MoML, but a lower inner learning rate $\alpha = 0.005$ is adopted to stabilize the training process. For the ablation on gradient-based Fast-MoML, α and β are same as in standard MoML.

4. More Experiments

4.1. Full Results on CMU-Mocap

We provide MPJPE errors of all actions in CMU-Mocap in Table 1. From the table, our MoML can effectively bring

	Basketball				Basketball Signal				Directing Traffic				Jumping			
millisecond (ms)	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Res. sup [8]	15.5	26.9	43.5	49.2	20.2	33.0	42.8	44.7	20.5	40.6	75.4	90.4	26.9	48.1	93.5	108.9
DMGNN [5]	15.6	28.7	59.0	73.1	5.0	9.3	20.2	26.2	10.2	20.9	41.6	52.3	32.0	54.3	96.7	119.9
MSR [2]	10.3	18.9	37.7	47.0	3.0	5.7	12.4	16.3	5.9	12.1	28.4	38.0	15.0	28.7	55.9	69.1
LTD [7]	11.7	21.3	41.0	50.8	3.3	6.3	**13.6**	18.0	6.9	13.7	30.3	40.0	17.2	32.4	60.1	72.6
LTD-FC	11.3	20.1	39.9	48.6	**3.2**	5.5	13.8	17.7	6.3	12.6	28.1	37.8	16.7	31.2	58.5	69.5
LTD-GC	**11.1**	**19.8**	**39.5**	**48.3**	**3.2**	**5.3**	13.8	**17.5**	**6.1**	**12.0**	**27.9**	**37.3**	**16.3**	**30.8**	**58.2**	**69.0**
SPGSN [6]	10.2	18.5	38.2	48.7	**2.9**	**5.3**	**11.3**	15.0	5.5	11.2	25.5	37.1	14.9	28.2	56.7	71.2
SPGSN-FC	9.8	**17.6**	**35.4**	**46.3**	3.3	5.8	12.6	14.7	5.3	**10.4**	**23.1**	**35.3**	14.1	**26.2**	**53.5**	**66.9**
SPGSN-GC	**9.7**	17.7	35.8	46.6	3.5	5.5	13.9	**14.5**	**5.1**	10.8	23.2	35.5	**13.6**	26.5	54.0	67.0

	Running				Soccer				Walking				Washwindow			
millisecond (ms)	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Res. sup [8]	25.8	48.9	88.2	100.8	17.8	31.3	52.6	61.4	44.4	76.7	126.8	151.4	22.8	44.7	56.8	104.7
DMGNN [5]	17.4	26.8	38.3	40.1	14.9	25.3	52.2	65.4	9.6	15.5	26.0	30.4	7.9	14.7	33.3	44.2
MSR [2]	12.8	20.4	30.6	34.4	10.9	19.5	37.1	46.4	6.3	10.3	17.6	21.1	5.5	11.1	25.1	32.5
LTD [7]	**14.5**	24.2	37.4	41.1	13.3	24.0	43.8	53.2	6.6	10.7	17.4	20.4	6.0	11.6	24.8	31.6
LTD-FC	14.6	23.3	36.6	39.7	12.7	22.8	42.1	48.4	6.1	9.6	16.8	**19.6**	5.6	10.4	23.0	27.9
LTD-GC	14.8	**23.1**	**35.9**	**38.4**	**12.4**	**21.9**	**41.7**	**47.8**	**6.0**	**9.2**	**16.2**	19.7	**5.4**	**10.3**	**22.6**	**27.5**
SPGSN [6]	10.8	16.7	26.1	30.1	10.9	19.0	35.1	45.2	6.3	10.2	16.3	20.2	4.9	9.4	21.5	28.4
SPGSN-FC	10.0	**14.5**	24.4	**27.8**	10.5	**17.0**	33.7	44.3	6.0	**9.0**	**14.6**	18.1	4.8	**8.8**	19.6	27.5
SPGSN-GC	**9.9**	14.6	**24.2**	29.0	**10.4**	17.3	**33.4**	**44.0**	**5.7**	9.3	14.8	**17.4**	**4.5**	9.1	**19.8**	**27.3**

Table 1. Comparisons of MPJPE errors on CMU-Mocap between baselines without/with our MoML approach.

	walking			eating			smoking			discussion		
	LTD	MotMix	SPGSN	LTD	MotMix	SPGSN	LTD	MotMix	SPGSN	LTD	MotMix	SPGSN
baseline	48.08	44.36	43.97	42.8	38.21	39.87	40.04	39.81	37.54	73.59	65.93	69.23
FC	45.87	**42.22**	42.44	41.21	**37.52**	38.96	38.96	**37.76**	**36.25**	71.84	**64.54**	68.68
GC	**45.45**	43.38	**42.01**	**41.01**	38.39	**38.75**	**38.89**	37.99	36.94	**71.65**	65.74	**66.81**

Table 2. Performance on unseen categories between baselines without/with our MoML approach. *MotMix* is short for *MotionMixer*.

baselines into online adaptive setting and yield improved performance in most cases.

4.2. MoML for Unseen Categories

Inspired by [4] that involves few-shot learning paradigm to predict human motions of unseen motion categories, we also analyze the compatibility of MoML in this scenario. Following [4], we exclude 4 classes of Human3.6M (*walking*, *eating*, *smoking* and *discussion*) from meta-training dataset $\mathcal{M}^{tr}$, and only train MoML with the remaining 11 classes. Motions from these 4 categories are regarded as unseen and used to evaluate the adaptability of our approach during meta-testing. Specifically, we draw multiple consecutive sub-tasks from certain novel category as novel streaming data, and adapt θ to suit each temporary novel context along the time. Note that, for fair comparison, we additionally re-train baselines on the 11 classes and directly evaluate their performance on the unseen 4 classes. Results are shown in Table 2, where our MoML also exhibits some adaptability w.r.t. different unseen motion categories.

4.3. Running Time

To verify the efficiency of MoML, we compare the running time of offline-trained baselines to the corresponding online adaptive modifications, including MoML with FC/GC-based MoAdapters and Fast-MoML. Meanwhile, to show the superiority of MoML against vanilla MAML [3] that update the entire model during adaptation. For gradient-based methods, we adopt 5 times of gradient update. Shown in Figure 2, our MoML achieves improvement on predictive accuracy while inevitably becoming more time-consuming, but is still significantly faster than conventional MAML.

4.4. More Visualizations

We additionally visualize another case *greeting*, shown in Figure 3. The motion appears varied that changes from waving right hand, putting hand down, waving left hand, putting hand down, to raising both hands, which is difficult to predict accurately. With MoML, we achieve more correct motion tendencies than offline-trained baseline.

5. Limitations

In real-world applications, the observation may not be clean and may involve occlusions or noises. How to deal with incomplete observed human poses remains a challenge for future study. On the other hand, model update during inference inevitably leads to more time consumption, how to maintain the online adaptive performance as well as considering time costs requires further research.

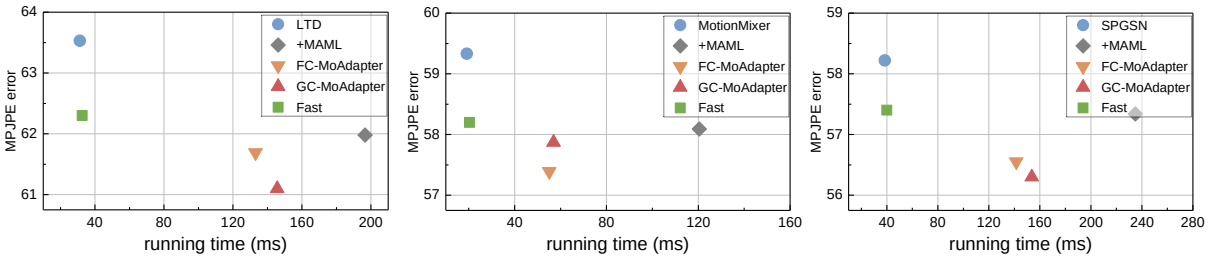

Figure 2. Comparisons of running time and MPJPE errors. *+MAML* means updating the entire model.

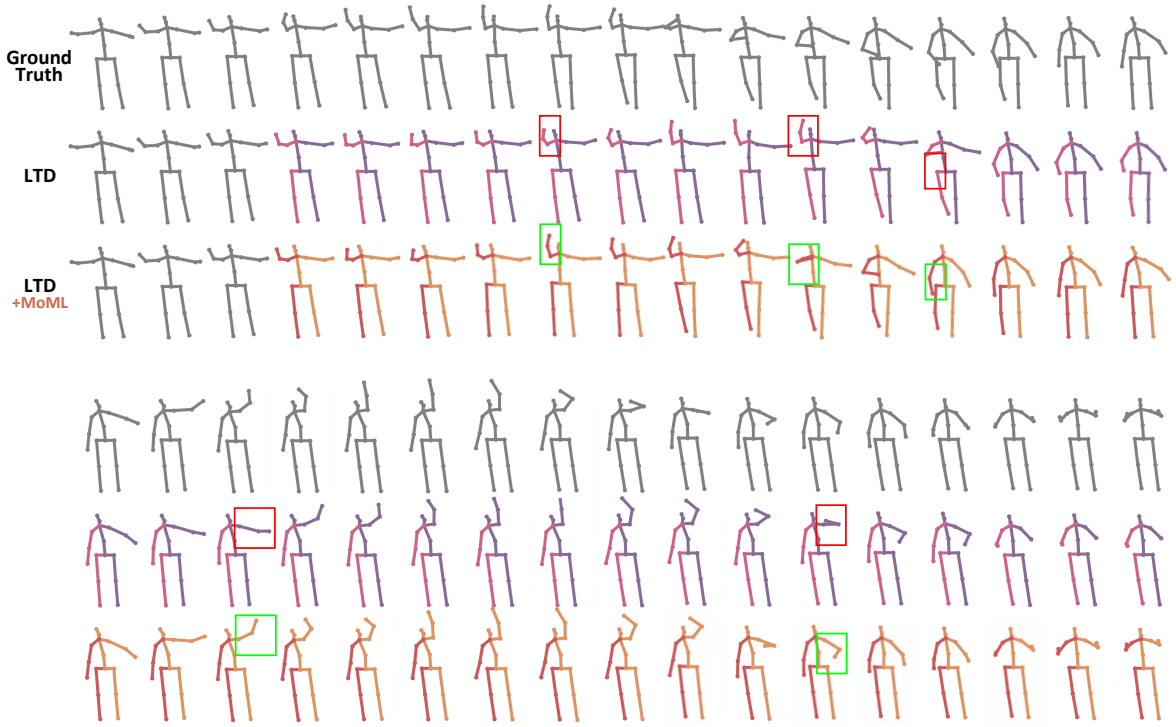

Figure 3. Another visualized case on *greeting*. In each case, we draw motion contents in eight seconds. The significant predictive errors marked in red boxes are alleviated by our adaptive setting of MoML and marked in green boxes.

References

[1] Arij Bouazizi, Adrian Holzbock, Ulrich Kressel, Klaus Dietmayer, and Vasileios Belagiannis. Motionmixer: Mlp-based 3d human body pose forecasting. In *IJCAI*, 2022. 1

[2] 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. 2

[3] 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

[4] 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

[5] 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] 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

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

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