Supplementary Material *for*
"Cyclic Test-Time Adaptation on Monocular Video for 3D Human Mesh Reconstruction"

In this supplementary material, we present more technical details and additional experimental results that could not be included in the main manuscript due to the lack of space. The contents are summarized below:

- S1. Visualization in video format
- S2. Results on other HMRNet architectures
- S3. Online adaptation scenario
- S4. Details of MDNet
- S5. Effect of pre-training HMRNet
- S6. MPJPE curves of diverse video sequences
- S7. Limitations
- S8. More qualitative results

S1. Visualization in video format

We provide supplementary video (**`CycleAdapt.mp4`**) that consists of three parts. The first part shows intermediate adaptation results during the cyclic adaptation process. Before adaptation, the HMRNet fails to produce plausible reconstruction results due to domain gap between training and test data. Our cyclic adaptation progressively adapts both the HMRNet and the MDNet as cycle repeats. The second part compares our proposed CycleAdapt with DynaBOA [4] and DAPA [13]. For the comparisons, we followed the released codes of the previous test-time adaptation methods. The last part provides results of CycleAdapt on Internet videos. We obtained human bounding boxes and 2D human keypoints for the test-time adaptation with AlphaPose [3].

S2. Results on other HMRNet architectures

Table S1 demonstrates that our CycleAdapt also significantly improves the accuracy of other HMRNet architectures [14, 10] in the test-time adaptation scenario. In the first and second rows of each block, we train HMRNet only using source dataset (*i.e.*, Human3.6M [7]) and evaluate it on each dataset. In the third row of each block, we apply our test-time adaptation framework by employing Human3.6M [7] as source dataset and 3DPW [12] as target dataset. Without the adaptation, all of HMRNet architectures suffer from domain gap problem and show poor performance on 3DPW, despite their superior performance on Human3.6M. Our CycleAdapt effectively adapts each of the networks with substantial improvements.

HMRNet architecture	Evaluation data	MPJPE	PA-MPJPE	MPVPE
SPIN [9]	Human3.6M	99.1	65.4	-
	3DPW before adapt.	230.3	123.4	253.4
	3DPW after adapt.	**87.7**	**53.8**	**105.7**
PyMAF [14]	Human3.6M	83.5	52.0	-
	3DPW before adapt.	309.1	152.8	336.7
	3DPW after adapt.	**98.5**	**57.2**	**122.7**
Pose2Pose [10]	Human3.6M	86.9	56.9	-
	3DPW before adapt.	331.8	157.5	364.2
	3DPW after adapt.	**108.1**	**55.8**	**121.9**

Table S1. Quantitative comparisons of CycleAdapt with different HMRNet architectures on 3DPW [12].

Meanwhile, we can observe that errors of PyMAF [14] and Pose2Pose [**?**]moon2022accurate after adaptation are slightly higher than those of SPIN [9]. We conjecture the reason is that PyMAF and Pose2Pose learn more domain-specific knowledge (*e.g.*, appearance) than SPIN and are more vulnerable to the domain gap problem. Accordingly, PyMAF and Pose2Pose show better performance on Human3.6M than SPIN (the first row of each block), but they show inferior performance on 3DPW (the second row of each block). Despite the various initial errors on 3DPW, our CycleAdapt uniformly reduces the MPJPE of SPIN, PyMAF, and Pose2Pose by 38%, 32%, and 33%.

S3. Online adaptation scenario

Table S2 shows that our CycleAdapt also achieves the best performance in online adaptation scenario, compared to BOA [5] and DynaBOA [4]. Since DAPA [13] does not support the online adaptation scenario, we only compare our CycleAdapt with BOA and DynaBOA. In the on-

Methods	MPJPE	PA-MPJPE	MPVPE
Base model (pre-trained on H36M)	230.3	123.4	253.4
BOA [5]	137.6	76.2	171.8
DynaBOA [4]	135.1	73.0	168.2
CycleAdapt (Ours)	**90.3**	**55.2**	**107.0**

(a) Source - Human3.6M / Target - 3DPW

Methods	MPJPE	PA-MPJPE	MPVPE
Base model (pre-trained on SURR)	193.2	92.0	216.5
BOA [5]	102.5	61.7	124.7
DynaBOA [4]	109.8	62.4	139.9
CycleAdapt (Ours)	**90.0**	**55.1**	**106.8**

(b) Source - SURREAL / Target - 3DPW

Table S2. Comparison between different test-time adaptation methods in **online adaptation scenario** on 3DPW [12]. OpenPose [2] is used to obtain 2D human keypoints from test images for the adaptation.

line adaptation scenario, test samples arrive in sequential order, and thus samples from future times cannot be utilized for adaptation. In this scenario, the accuracy of our CycleAdapt slightly drops as the MDNet cannot view human motion in the future. Nevertheless, CycleAdapt still outperforms BOA and DynaBOA.

S4. Details of MDNet

Architecture. Figure S1 shows the detailed architecture of the MDNet in our framework. Motivated by recent research [6] on human motion modeling for human motion prediction, we configure the MDNet with fully-connected layers and layer normalization [1]. For all layers, their input dimension is equal to their output dimension. The MDNet initially forms a matrix $\mathbf{\Theta} \in \mathbb{R}^{T \times H}$ by concatenating input SMPL pose parameters $\{\theta_0, \ldots, \theta_{T-1}\}$ that are randomly masked, where $T = 49$ and $H = 144$ denote the temporal length of the pose parameter sequence and the dimension of the pose parameter, respectively. The matrix is passed into a fully-connected layer followed by a transpose operation. The transposed matrix is forwarded into a series of M blocks ($M = 4$), which also consist of fully-connected layers and layer normalization. Finally, we perform the last transpose operation followed by a fully-connected layer to obtain denoised SMPL pose parameters $\mathbf{\Theta}' = \{\theta'_0, \ldots, \theta'_{T-1}\}$.

Pre-training scheme. To pre-train the MDNet, we utilize the MoCap dataset (*i.e.*, Human3.6M [7]), which contains accurate 3D labels. With the MoCap dataset, we add random gaussian noise into the SMPL pose parameters to mimic noisy human meshes reconstructed from HMRNet.

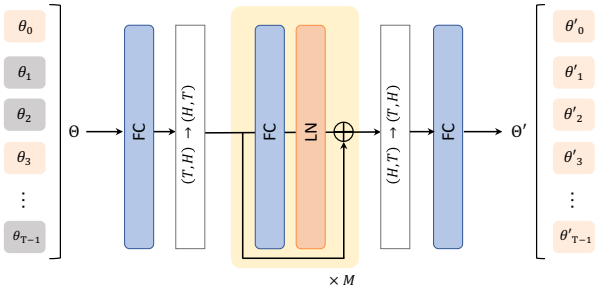

Figure S1: The pipeline of MDNet. FC and LN denote fully-connected layer and layer normalization [1], respectively.

The mean and standard deviation of the random gaussian noise are set to 0 and 0.01, respectively. We forward the parameters with synthesized noises into MDNet and construct a loss function as follows:

$$L_{\text{MD}} = \frac{1}{T} \sum_{t=0}^{T-1} \|\theta'_t - \theta^*_t\|_1, \tag{1}$$

where the asterisk denotes groundtruth from the MoCap dataset.

S5. Effect of pre-training HMRNet

Table S3 shows that pre-training HMRNet on the source dataset (*i.e.*, Human3.6M [7]) is necessary for the test-time adaptation scenario. Before adaptation, the HMRNet pre-trained on the source dataset (the third row) shows similar performance to HMRNet with random initialization (the first row). This is due to the domain gap between the source and target dataset, as described in Section 1. Although the effect of pre-training is not directly reflected on accuracy before adaptation, pre-training on source dataset (the fourth row) is considerably effective compared to random initialization (the second row), in the test-time adaptation scenario. This is because the pre-trained HMRNet on source dataset learned prior of human structure that is helpful in 3D human mesh reconstruction. Our test-time adaptation framework effectively takes advantage of the learned human prior during adaptation, which boosts the performance of test-time adaptation.

S6. MPJPE curves of diverse video sequences

Figure S2 shows that the MPJPE curve of MDNet is mostly below that of HMRNet for most cycles, similar to Figure 4. Such consistent tendency of the two curves demonstrates that the outputs of MDNet can serve as reliable guidance as supervision targets for HMRNet, during the adaptation.

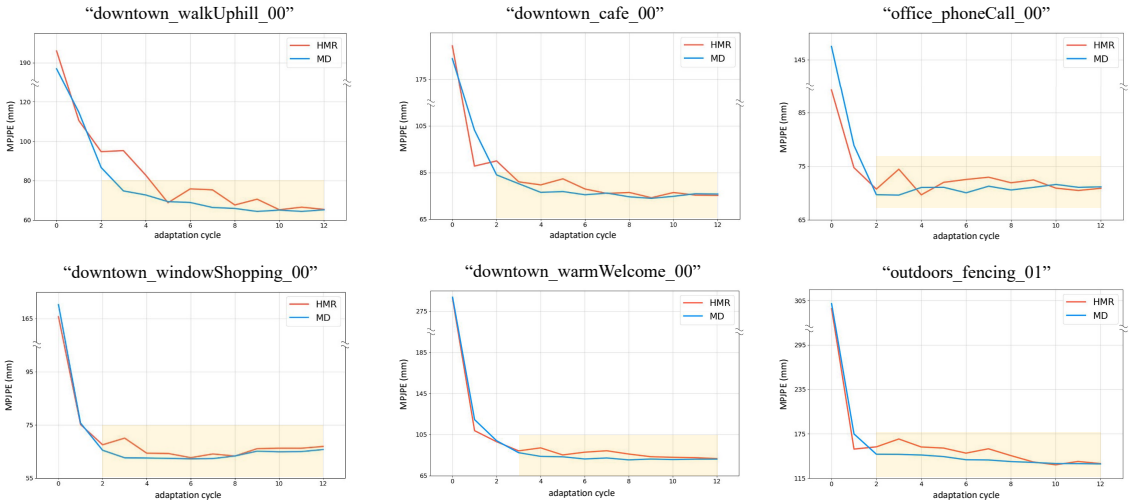

Figure S2: MPJPE curves during test-time adaptation for different video sequences in 3DPW [12].

Pre-training	Test-time adapt.	MPJPE	PA-MPJPE	MPVPE
Random init.	✗	272.0	111.7	324.0
	✓	140.6	89.6	163.3
Pre-training on H36M	✗	230.3	123.4	253.4
	✓	87.0	52.4	104.1

Table S3. Effect of pre-training HMRNet on test-time adaptation. 3DPW [12] is used for the adaptation.

S7. Limitations

Figure S3 shows that our framework often struggles to adapt on a test video when the video contains extremely fast human motion. Given fast human movements, the human meshes reconstructed from HMRNet dramatically change as the timestamp progresses. For MDNet, it is highly ambiguous to distinguish between dramatically changing human meshes and noisy human meshes. Thus, the MDNet often produces over-smoothed outputs when adaptation on such challenging test video. Due to the difficulty, test-time adaptation with fast human motion can be a future research direction.

S8. More qualitative results

We provide more qualitative result comparisons on the 3DPW [12] test set and the InstaVariety [8] test set. Figure S4 and S5 show that our CycleAdapt produces far more accurate results compared to previous test-time adaptation methods.

License of the Used Assets

- Human3.6M dataset [7]'s licenses are limited to academic use only.

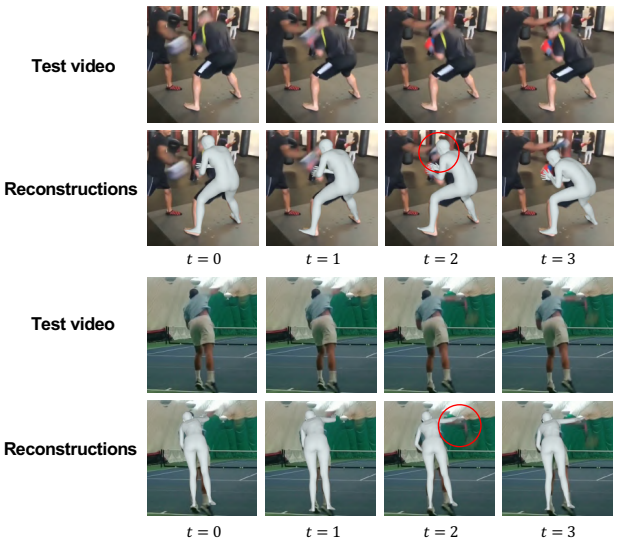

Figure S3: Failure cases of our framework.

- SURREAL dataset [11] is available for the sole purpose of performing non-commercial scientific research.
- 3DPW dataset [12] is released for academic research only and it is free to researchers from educational or research institutes for non-commercial purposes.
- InstaVariety dataset [8] is released for non-commercial academic use.
- BOA codes [5] are released for academic research only and it is free to researchers from educational or research institutes for non-commercial purposes.
- DynaBOA codes [4] are released for academic research only and it is free to researchers from educational or research institutes for non-commercial purposes.
- DAPA codes [13] are MIT licensed.

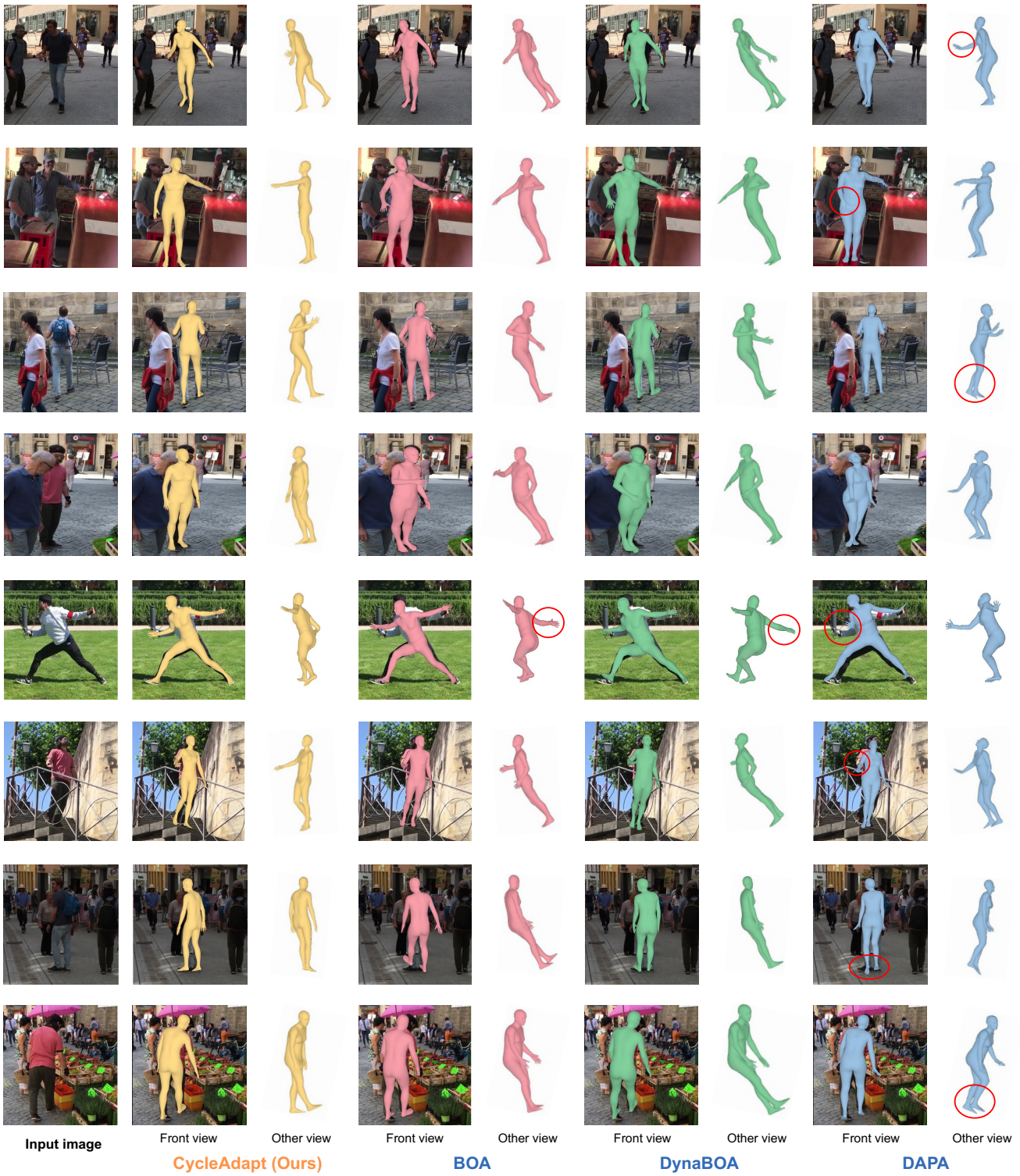

Figure S4: Comparison of HMRNet's accuracy between different test-time adaptation methods, when using Human3.6M [7] as source dataset and 3DPW [12] as target dataset. OpenPose [2] is used to obtain 2D human keypoints from test images for the adaptation.

Figure S5: Comparison of HMRNet's accuracy between different test-time adaptation methods, when using Human3.6M [7] as source dataset and InstaVariety [8] as target dataset. OpenPose [2] is used to obtain 2D human keypoints from test images for the adaptation.

References

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

[2] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. In *CVPR*, 2017.

[3] Hao-Shu Fang, Jiefeng Li, Hongyang Tang, Chao Xu, Haoyi Zhu, Yuliang Xiu, Yong-Lu Li, and Cewu Lu. AlphaPose: Whole-body regional multi-person pose estimation and tracking in real-time. *TPAMI*, 2022.

[4] Shanyan Guan, Jingwei Xu, Michelle Z He, Yunbo Wang, Bingbing Ni, and Xiaokang Yang. Out-of-domain human mesh reconstruction via dynamic bilevel online adaptation. *TPAMI*, 2022.

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

[6] Wen Guo, Yuming Du, Xi Shen, Vincent Lepetit, Xavier Alameda-Pineda, and Francesc Moreno-Noguer. Back to MLP: A simple baseline for human motion prediction. In *WACV*, 2023.

[7] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *TPAMI*, 2014.

[8] Angjoo Kanazawa, Jason Y Zhang, Panna Felsen, and Jitendra Malik. Learning 3D human dynamics from video. In *CVPR*, 2019.

[9] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3D human pose and shape via model-fitting in the loop. In *ICCV*, 2019.

[10] Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. Accurate 3D hand pose estimation for whole-body 3D human mesh estimation. In *CVPRW*, 2022.

[11] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*, 2017.

[12] 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*, 2018.

[13] Zhenzhen Weng, Kuan-Chieh Wang, Angjoo Kanazawa, and Serena Yeung. Domain adaptive 3D pose augmentation for in-the-wild human mesh recovery. In *3DV*, 2022.

[14] Hongwen Zhang, Yating Tian, Xinchi Zhou, Wanli Ouyang, Yebin Liu, Limin Wang, and Zhenan Sun. PyMAF: 3D human pose and shape regression with pyramidal mesh alignment feedback loop. In *ICCV*, 2021.