UPose3D: Uncertainty-Aware 3D Human Pose Estimation with Cross-View and Temporal Cues

Vandad Davoodnia[1,2], Saeed Ghorbani[2], Marc-André Carbonneau[2], Alexandre Messier[2], and Ali Etemad[1]

[1] Queen's University, Canada
[2] Ubisoft LaForge, Canada
{vandad.davoodnia,ali.etemad}@queensu.ca
{saeed.ghorbani,marc-andre.carbonneau2,alexandre.messier}@ubisoft.com

1 Additional Details

1.1 Details on Training and Multi-view Data Synthesis

Let $r \in \mathbb{R}^{T\times 3}$, $\Theta \in \mathbb{R}^{T\times 55\times 3\times 3}$, and $\beta \in \mathbb{R}^{16}$ be root position, body joint rotation matrices in a temporal window of length T, and shape parameters from human motion capture data. We begin our multi-view data generation by augmenting the shape parameters with Gaussian noise with a standard deviation equal to the standard deviation of all shape parameters within the datasets. Next, we apply mediolateral mirroring with a 50% chance and randomly rotate the motion sequence around its center. We pass the augmented $\{r, \Theta, \beta\}$ parameters to the forward-kinematic layer of the SMPL body model to obtain 3D vertices. Lastly, we use a dataset-specific joint regressor on the vertices to extract the 3D keypoints used in the next steps of our pipeline.

Next, we simulate a multi-camera recording setup by randomly positioning several cameras in cylindrical space. As depicted in Fig. 1-a, we randomly choose a recording volume size that encircles the space occupied by the human body. To ensure that the subject appears in most cameras, we select the tilt, pitch, and yaw so that they look at a random point in the center of the recording volume while maintaining the correct up direction. Additionally, we limit the camera height to mimic typical multi-view video recording setups.

After obtaining the camera intrinsic and extrinsic parameters, we project the 3D body keypoints onto each camera view (see Fig. 1-b). We use these 2D keypoints as ground truths μ_g to train our pose compiler (see Fig. 1-c). We then add 2D point corruptions to the 2D keypoints, including Gaussian noise with varying standard deviations, simulated occlusions with varying sizes and probabilities, mediolateral flipping, and occasional truncation effects (see Fig. 1-d). Next, we obtain the cross-view projected keypoints (see Fig. 1-e) via the algorithm described in Sec. 3.2 in the main paper. Finally, we train our pose compiler using the ground-truth keypoints and point clouds containing noisy 2D data, as depicted in Fig. 2.

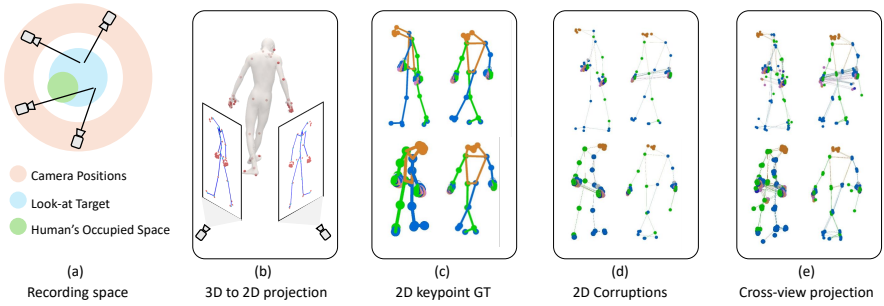

Fig. 1: We illustrate our multi-view data synthesis framework, starting with (a) camera placement in a space surrounding a motion-captured human body; (b) extraction and projection of keypoints onto the synthetic cameras; (c) 2D ground-truth keypoints; (d) data corruption; and (e) cross-view projection to prepare the point cloud training data for our pose compiler.

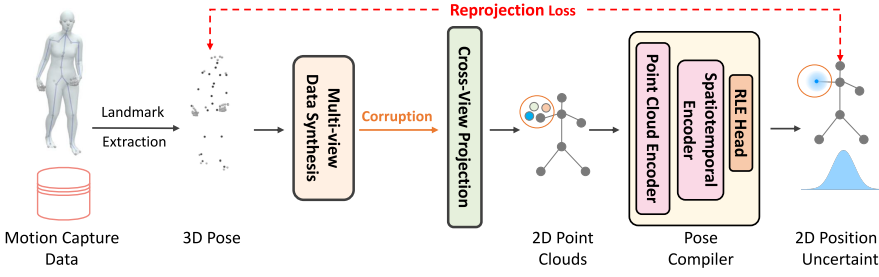

Fig. 2: We illustrate the training routine of our pose compiler using synthetic data generated based on real motion capture sequences.

1.2 Details on Criss-cross Attention

As discussed in Sec. 3.3 in the main paper, we use criss-cross attention blocks in our spatiotemporal encoder to process information more efficiently. Accordingly, the cross-view input features $\{f_i\}$ are first projected into queries, keys, and values ($Q, K, V \in \mathbb{R}^{T\times J\times 2H}$) via a linear layer. Next, we divide them into temporal $Q_T, K_T, V_T \in \mathbb{R}^{T\times J\times H}$ and spatial groups $Q_S, K_S, V_S \in \mathbb{R}^{T\times J\times H}$. The temporal and spatial (skeleton joints) attentions are then calculated in two separate self-attention modules and concatenated before the next feed-forward layer and normalization. As a result of this operation, the receptive field of each transformer layer is the information residing on the spatial and temporal axis, and stacking multiple layers can approximate the full spatiotemporal attention without large computational overhead. In the following sections, we study the effectiveness of our design choice and compare its computation cost and performance against full attention and concurrent attention designs.

2 Additional Experiments and Results

This section describes the 2D datasets used during training and fine-tuning of our 2D pose estimator. We then study the details of our pipeline to evaluate its performance under different inputs, network architectures, and initialization strategies for 3D keypoint estimation. Next, we provide additional comparisons on the Human3.6m [5] dataset with weak or semi-supervised methods. We will also provide more comparisons with monocular pose estimation approaches on the RICH [4] dataset.

2.1 2D Datasets

COCO WholeBody. The COCO WholeBody [6] dataset is a large-scale whole-body pose estimation dataset with over 250K samples. This dataset is an extension of Common Objects in COntext (COCO) [14] dataset with the same training and testing splits. The dataset provides 133 2D keypoints (17 for body, 6 for feet, 68 for face, and 42 for hands) on in-the-wild images. We use this dataset to train our 2D pose estimator during OoD experiments.
MPII. The MPII Human Pose dataset [1] dataset is a popular 2D pose estimation benchmark. It contains over 40,000 images of people performing over 400 actions in diverse scenarios. The dataset contains 16 body joint labels and is frequently used to pre-train [24] or improve cross-dataset generalization [16, 25]. We use this dataset for 2D pose estimator pre-training and fine-tuning.

2.2 Additional Ablation Study

Following the ablation study originally presented in Sec. 5.3 of the main paper, we investigate the impact of temporal length, our spatiotemporal encoder's architecture, different formulations of the point clouds, and our initialization strategy for 3D keypoint estimation in Tab. 1. Additionally, we report and provide the computational cost comparisons for a single input batch with T = 27 and 4 views. Our pose compiler is significantly smaller than a single 2D pose estimator, taking less than 1% of the total parameter count.
Random Initialization. We use the L-BFGS [15] optimization algorithm to solve the 3D keypoint MLE iteratively. To speed up this process, we stop the optimization when the changes of our optimization variables, namely U, are less than a specific tolerance (Tol. = 0.001 mm). We further speed up the optimization process by using a DLT algorithm to initialize the 3D points U. Table 1 first examines the effect of our initialization strategy when U is initialized to zero, and the tolerance remains unchanged, showing a significant rise in the 3D keypoint estimation error and inference time. Next, Tab. 1 shows that by lowering the tolerance, zero-initialization performs similarly to our proposed strategy, but at 3 times more inference time. Therefore, we conclude that unlike prior works [22], our method is not reliant on initialization, and the initialization only speeds up the estimation process. This may be due to the smooth nature of the uncertainty distributions learned by the normalizing flows [11].

Table 1: Additional ablation study on Human3.6m dataset. We only report the computation cost of our pose compiler (in FLOPs) and exclude the CPN [2] network with $5.16T$ FLOPs for 27 frames of 4 views. Additionally, $64.87M$ of parameters in all experiments belong to the CPN network.

Method	MPJPE↓	PA-MPJPE↓	Param. (M)↓	Time (s)↓	FLOPs (G)
UPose3D (T = 27, Tol. = 10^{-3} *mm*)	26.42	23.42	65.407	10.1	2.04
w/ zero init	28.51	32.85	65.407	12.5	2.04
w/ zero init (Tol. = 10^{-6} *mm*)	26.42	23.42	65.407	28.9	2.04
w/ T = 243	33.17	25.11	62.660	10.9	20.18
w/ concurent attention	26.57	23.61	65.391	10.3	2.01
w/ full attention	26.50	23.57	65.322	10.3	2.28
w/ full attention (T = 243)	34.97	28.60	65.336	10.3	51.56
w/ epipolar line	26.46	23.45	65.407	10.1	2.04
w/ relative camera pos. emb.	26.37	23.43	65.407	10.2	2.04

Longer Temporal Window. We study the computational cost and performance impact of very long temporal context size. Following [26], we report the performance of UPose3D when 243 frames, as opposed to 27 frames, to infer the 3D keypoints of the center frame in Tab. 1. This new model takes 10 times more FLOPs to compute and does not perform as well as our original model. This may be because our synthetic data augmentations and corruption strategies are tuned for smaller time windows, as longer context sizes were not in our considerations. Our observations of the training and validation losses also show signs of overfitting during training for longer time windows. As extremely long context sizes are not in the scope of this paper, we do not perform any additional tuning of these models and leave them for future research.

Pose Compiler Architecture. We compare the effect of our criss-cross attention modules with vanilla (full) and concurrent attention. Table 1 shows that criss-cross attention outperforms the other two designs while requiring less computation (FLOPs) than full attention. Additionally, we observe that on the extreme case of very long temporal context sizes (T = 243), criss-cross attention still outperforms full attention models by 1.8 *mm* while requiring 60% less computations.

Inputs of Pose Compiler. Finally, we investigate the effect of different point cloud formation strategies in our pipeline. Specifically, we study the impact of appending a relative camera position embedding, inspired by [19], to the cross-projected 2D keypoints while creating the point clouds. Accordingly, in our first experiment, we concatenate the epipolar line parameters of other views to the point cloud of the reference view. Similarly, in our second experiment, we concatenate the relative position of the other cameras to the input point cloud as well. However, as shown in Tab. 1, adding extra inputs does not greatly impact the performance.

2.3 Additional Baselines and Comparisons

Table 2 complements Tab. 1 of the main paper by providing more comparisons with prior works on the Human3.6m [5] dataset. Here, we include 3D keypoint estimation approaches regardless of their input modality or supervision type

Table 2: Additional comparisons with prior works on the full test set of the Human3.6m dataset in InD settings. (-) denotes that the error was not reported in the original work.

Method	Supervision	Multi-view	Frames	MPJPE↓	PA-MPJPE↓	N-MPJPE↓
Rhodin *et al.* [17]	3D	✗	1	66.8	51.6	63.3
Rhodin *et al.* [17]	Weakly 3D	✗	1	-	65.1	80.1
EpipolarPose [9]	Weakly 3D	✗	1	55.08	47.91	54.90
CanonPose [23]	Weakly 3D	✗	1	-	53.0	82.0
Gong *et al.* [3]	Synthetic 3D	✓	1	53.8	42.4	-
BKinD-3D [20]	3D Discovery	✓	20	125.0	105.0	-
UPose3D (Ours)	2D	✓	1	26.9	24.1	26.2
UPose3D (Ours)	2D	✓	27	26.4	23.4	25.6

Table 3: Comparison of our method in OoD setting on RICH dataset against prior works. * denotes our replication of prior works.

Method	MPJPE↓	PA-MPJPE↓	OoD	Multi-view	Output
SA-HMR [18]	93.9	-	✗	✗	SMPL
IPMAN-R [21]	79.0	47.6	✗	✗	SMPL
METRO [13]	98.8	-	✗	✗	SMPL
METRO [13]	129.6	-	✓	✗	SMPL
SPIN [10]	112.2	71.5	✓	✗	SMPL
PARE [8]	107.0	73.1	✓	✗	SMPL
CLIFF [12]	107.0	67.2	✓	✗	SMPL
AdaFuse* [24]	524.0	85.8	✓	✓	3D Keypoints
HRNet-W48+Grid Search*	64.4	54.9	✓	✓	3D Keypoints
HRNet-W48+DLT*	66.0	55.1	✓	✓	3D Keypoints
Ours ($T = 1$)	36.2	33.4	✓	✓	3D Keypoints
Ours ($T = 27$)	34.7	32.0	✓	✓	3D Keypoints

in InD settings. We observe that our method outperforms all of the other approaches despite only using 2D supervision. Additionally, in Tab. 3, we compare our work with prior research on the RICH [4] dataset. Since this dataset was recently published, only monocular 3D body modeling techniques have reported their performance on this dataset. Here, we observe that our method outperforms the majority of prior works. More importantly, when comparing Tab. 2 and Tab. 3 we notice that our method achieves consistent results between InD and OoD evaluations on the Human3.6m [5] and the RICH [4] datasets, showing generalizability across in-studio and outdoor environments.

3 Additional Visual Examples

We provide a supplementary video that describes our method with visual demonstrations. Additionally, we provide several video clips of input and output data from Human3.6m [5], RICH [4], and CMU-Panoptic [7] datasets and compare the visual fidelity of our approach with the state-of-the-art method on Human3.6m, AdaFuse [24].

4 Qualitataive Comparisons

In Fig. 3, Fig. 4, and Fig. 5, we demonstrate some examples of our UPose3D on Human3.6m [5] dataset to showcase its visual fidelity in comparison to ground-truth keypoints and AdaFuse [24] in InD evaluation scheme. Additionally, we provide more visual examples of UPose3D results in Fig. 6, Fig. 7, and Fig. 8 in comparison to our implementation of AdaFuse [24] in OoD settings on the RICH [4] dataset. To better visualize the sharp keypoint distribution output of our 2D pose estimators, we show the logarithm of heatmaps in all figures for 2D pose estimators. We refer the reader to Fig. 4 of the main paper for an illustration of the real heatmaps without any post-processing. Our method performs consistently in both settings, while AdaFuse fails to correctly predict the human keypoints in some OoD samples. In all cases, the 2D pose estimator generally results in more refined predictions and sharper uncertainty distributions, while our pose compiler outputs a coarser distribution. Moreover, our method typically depicts higher horizontal uncertainties, which may be due to more frequent horizontal movements.

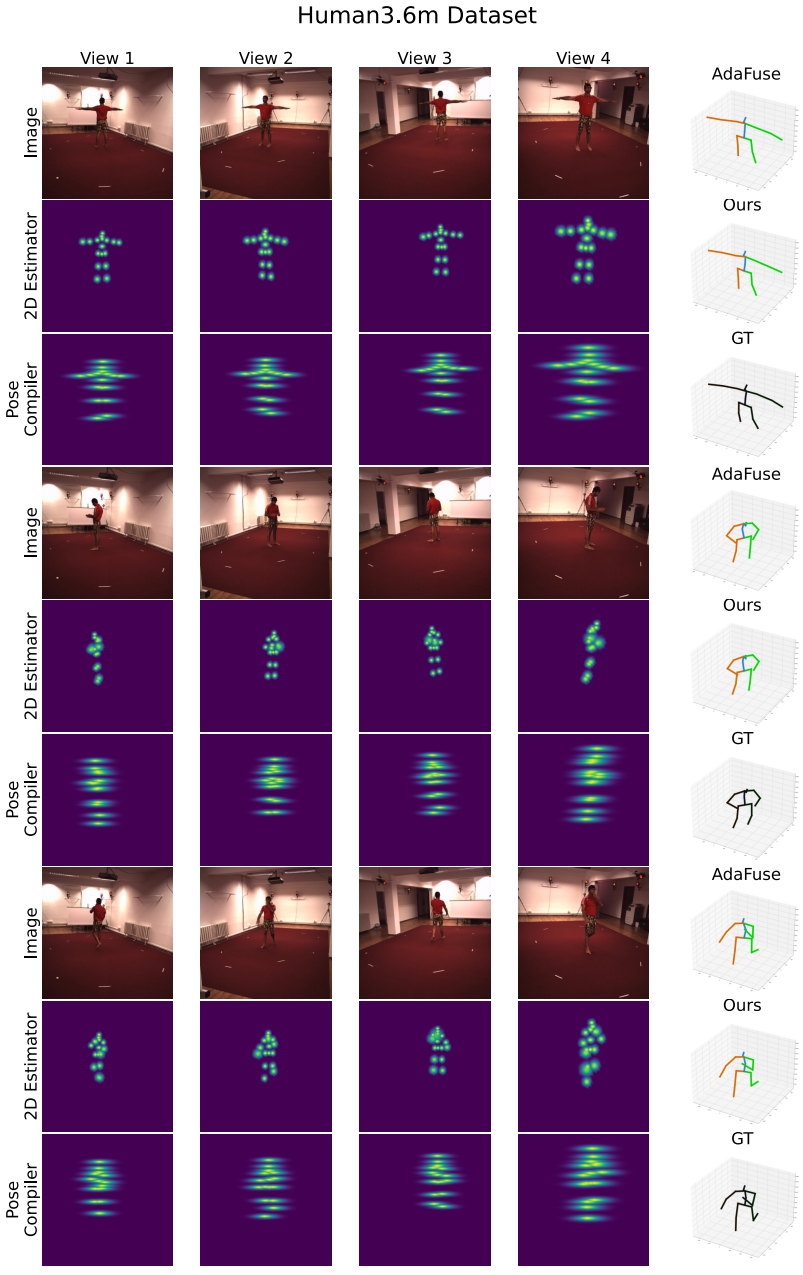

Fig. 3: Example output of our proposed UPose3D pipeline in comparison to AdaFuse [24] is presented in the InD evaluation scheme on the Human3.6m [5] dataset.

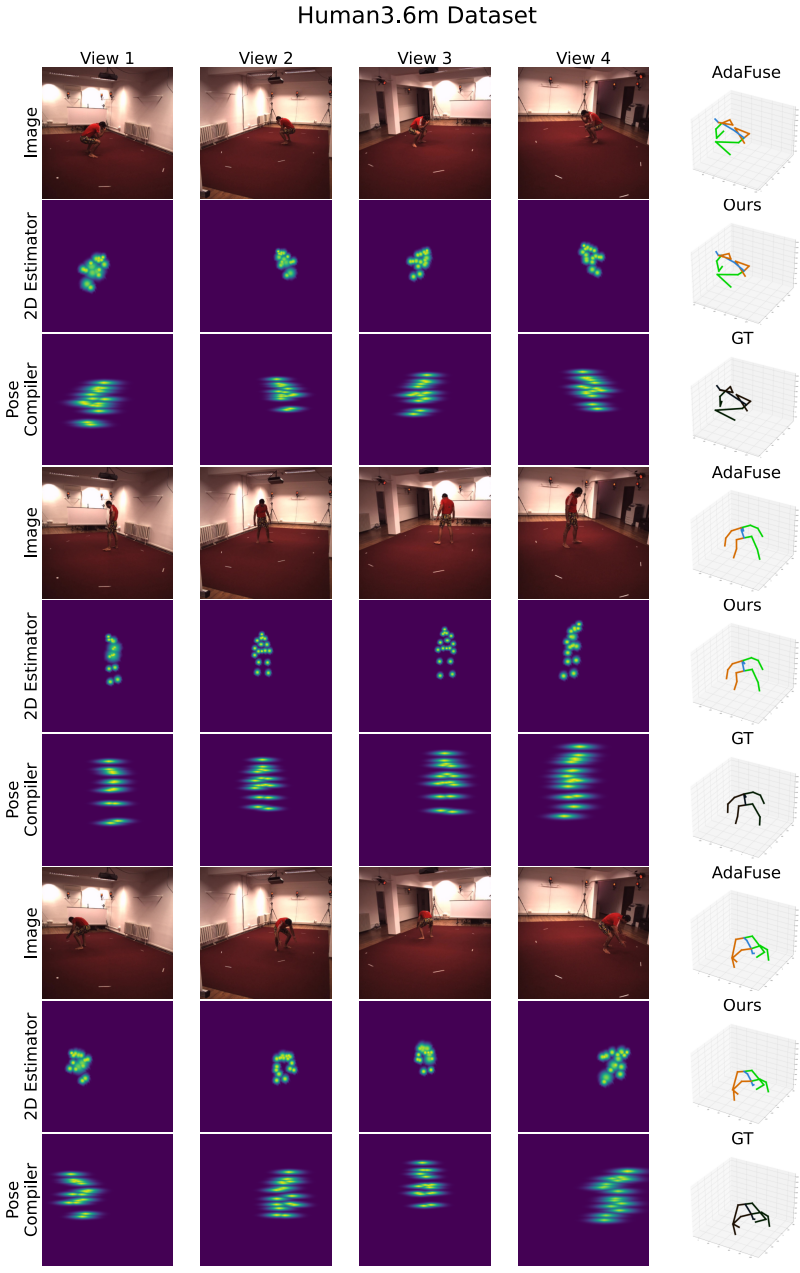

Fig. 4: Example output of our proposed UPose3D pipeline in comparison to AdaFuse [24] is presented in the InD evaluation scheme on the Human3.6m [5] dataset.

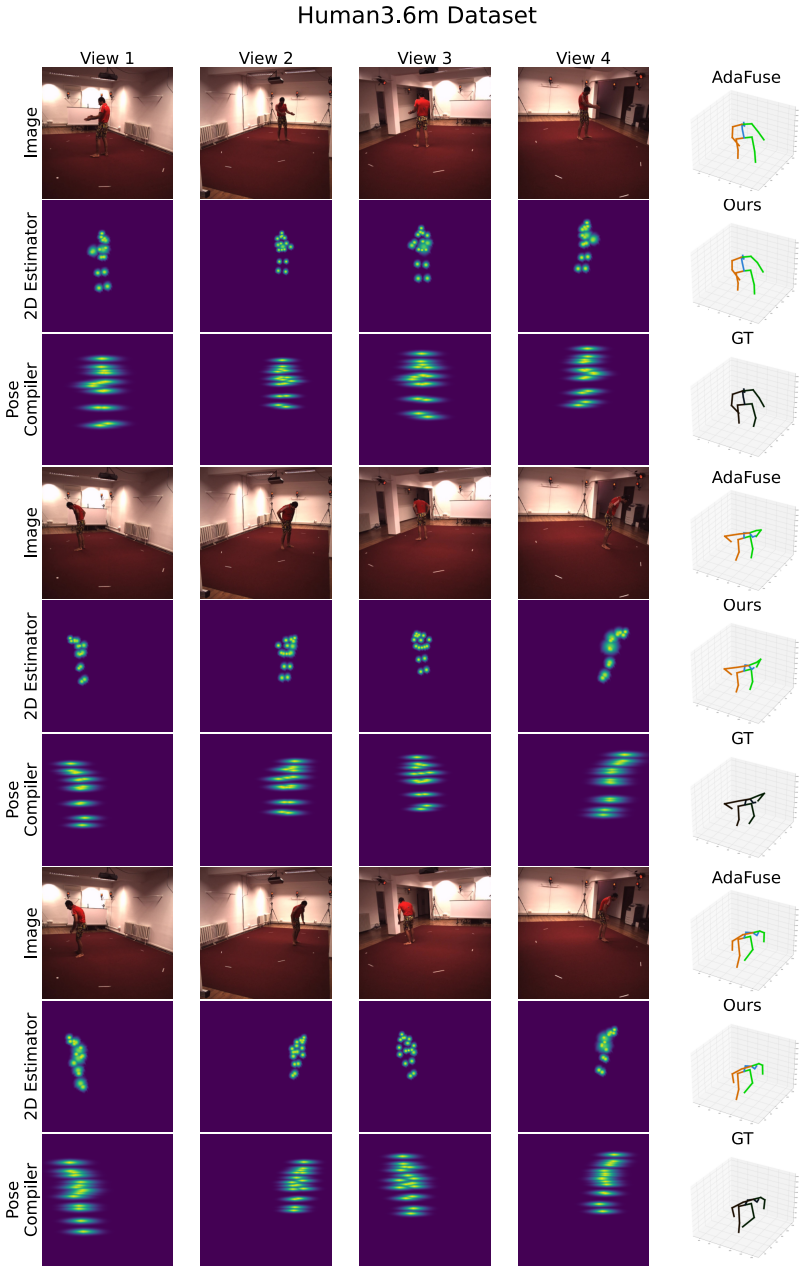

Fig. 5: Example output of our proposed UPose3D pipeline in comparison to AdaFuse [24] is presented in the InD evaluation scheme on the Human3.6m [5] dataset.

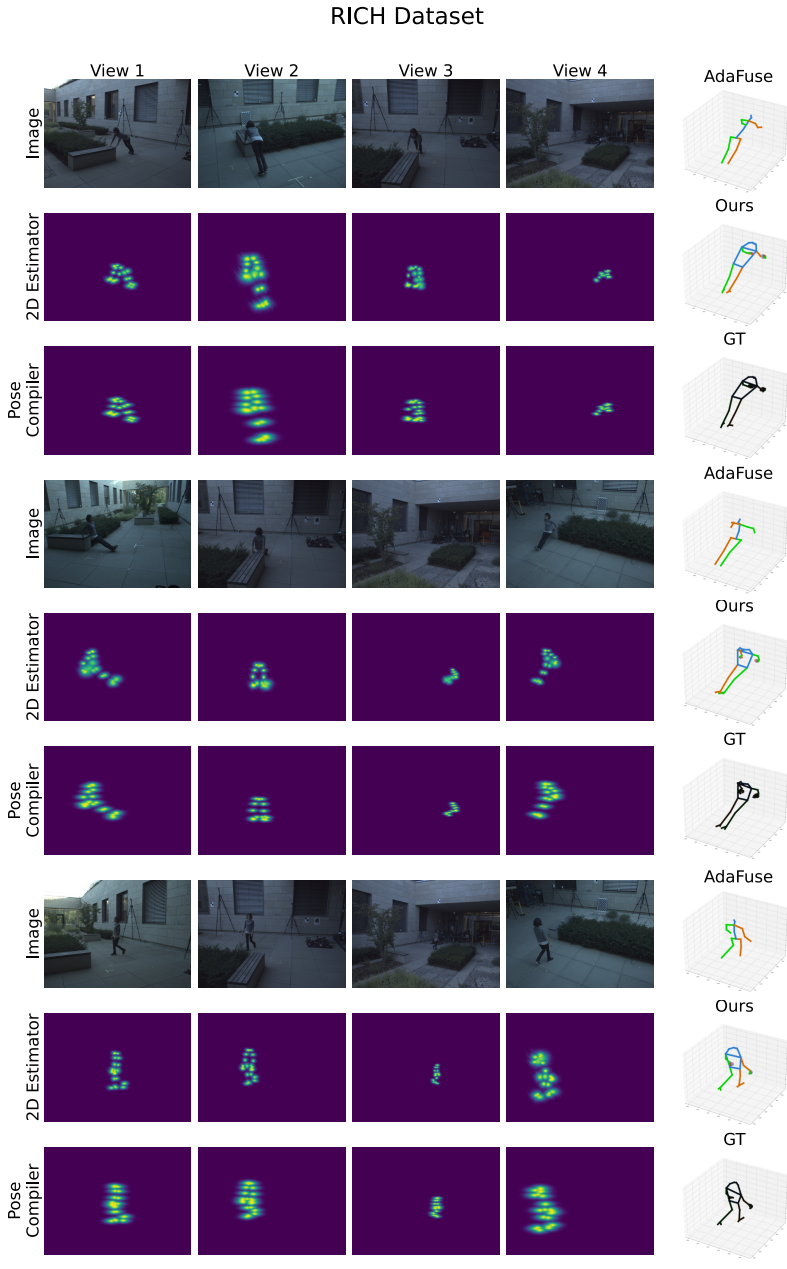

Fig. 6: Example output of our proposed UPose3D pipeline in comparison to AdaFuse [24] is presented in the OoD evaluation scheme on the RICH [4] dataset. The first and second samples show the effectiveness of our approach in solving occlusions for detecting hands and feet.

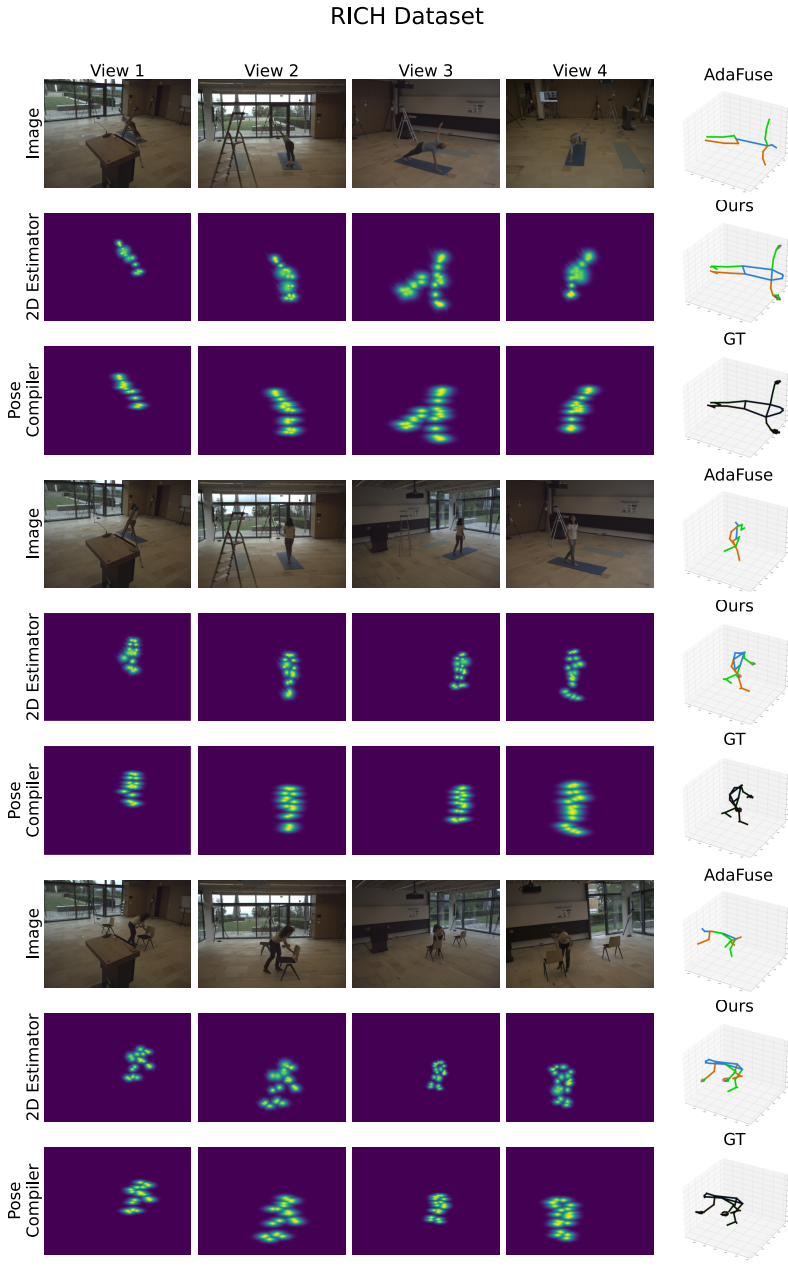

Fig. 7: Example output of our proposed UPose3D pipeline in comparison to AdaFuse [24] is presented in the OoD evaluation scheme on the RICH [4] dataset. The first sample illustrates a challenging input with a rare posture, where both AdaFuse and our method successfully predict the correct posture.

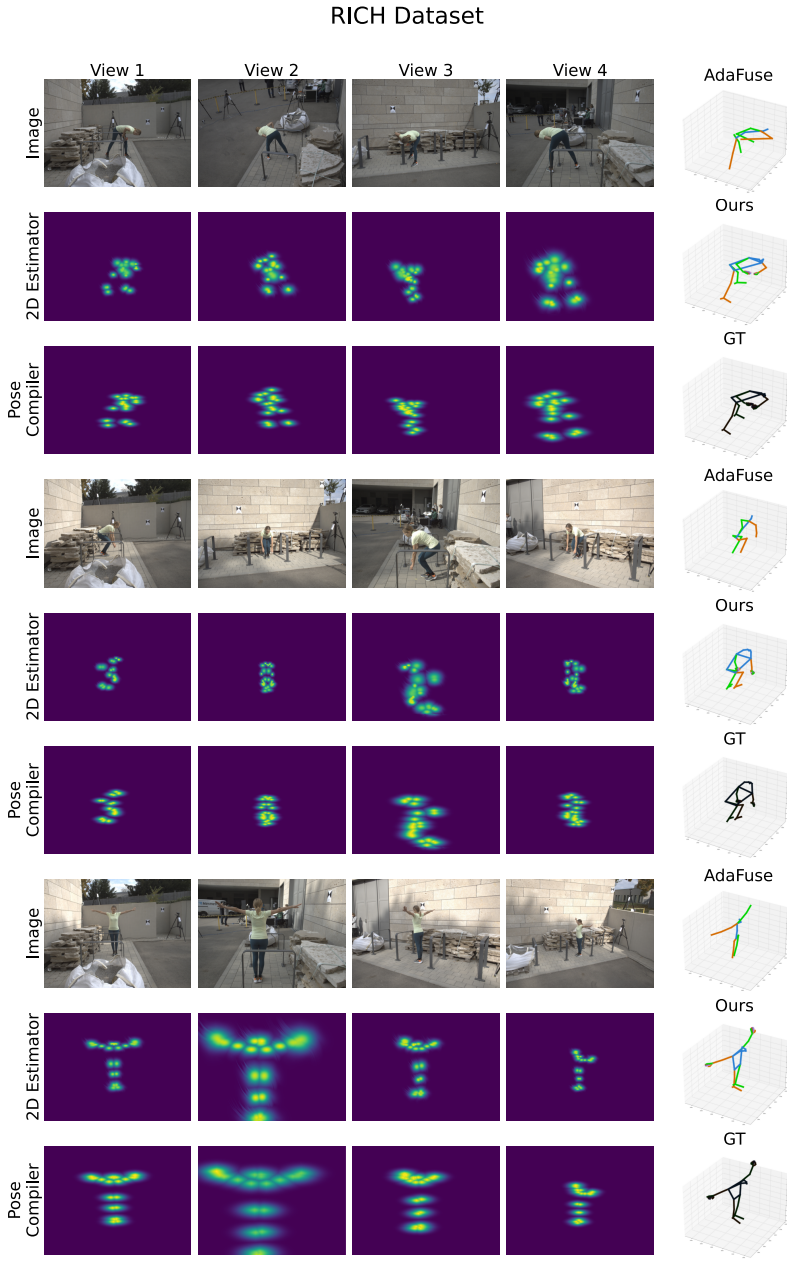

Fig. 8: Example output of our proposed UPose3D pipeline in comparison to AdaFuse [24] is presented in the OoD evaluation scheme on the RICH [4] dataset. We observe that our method outperforms AdaFuse in the first and third samples.

References

1. Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B.: 2D human pose estimation: New benchmark and state of the art analysis. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3686–3693 (2014). https://doi.org/10.1109/CVPR.2014.471
2. Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., Sun, J.: Cascaded pyramid network for multi-person pose estimation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7103–7112 (2018). https://doi.org/10.1109/CVPR.2018.00742
3. Gong, X., Song, L., Zheng, M., Planche, B., Chen, T., Yuan, J., Doermann, D., Wu, Z.: Progressive multi-view human mesh recovery with self-supervision. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 676–684 (2023). https://doi.org/10.1609/aaai.v37i1.25144
4. Huang, C.H.P., Yi, H., Höschle, M., Safroshkin, M., Alexiadis, T., Polikovsky, S., Scharstein, D., Black, M.J.: Capturing and inferring dense full-body human-scene contact. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13274–13285 (2022). https://doi.org/10.1109/CVPR52688.2022.01292
5. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3.6m: Large scale datasets and predictive methods for 3D human sensing in natural environments. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(7), 1325–1339 (2013). https://doi.org/10.1109/TPAMI.2013.248
6. Jin, S., Xu, L., Xu, J., Wang, C., Liu, W., Qian, C., Ouyang, W., Luo, P.: Whole-body human pose estimation in the wild. In: European Conference on Computer Vision (ECCV). pp. 196–214. Springer (2020). https://doi.org/10.1007/978-3-030-58545-7_12
7. Joo, H., Liu, H., Tan, L., Gui, L., Nabbe, B., Matthews, I., Kanade, T., Nobuhara, S., Sheikh, Y.: Panoptic studio: A massively multiview system for social motion capture. In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3334–3342 (2015). https://doi.org/10.1109/ICCV.2015.381
8. Kocabas, M., Huang, C.H.P., Hilliges, O., Black, M.J.: Pare: Part attention regressor for 3D human body estimation. In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 11127–11137 (2021). https://doi.org/10.1109/ICCV48922.2021.01094
9. Kocabas, M., Karagoz, S., Akbas, E.: Self-supervised learning of 3D human pose using multi-view geometry. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1077–1086 (2019). https://doi.org/10.1109/CVPR.2019.00117
10. Kolotouros, N., Pavlakos, G., Black, M.J., Daniilidis, K.: Learning to reconstruct 3D human pose and shape via model-fitting in the loop. In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 2252–2261 (2019). https://doi.org/10.1109/ICCV.2019.00234
11. Li, J., Bian, S., Zeng, A., Wang, C., Pang, B., Liu, W., Lu, C.: Human pose regression with residual log-likelihood estimation. In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 11025–11034 (2021). https://doi.org/10.1109/ICCV48922.2021.01084
12. Li, Z., Liu, J., Zhang, Z., Xu, S., Yan, Y.: Cliff: Carrying location information in full frames into human pose and shape estimation. In: European Conference on Computer Vision (ECCV). pp. 590–606. Springer (2022). https://doi.org/10.1007/978-3-031-20065-6_34

13. Lin, K., Wang, L., Liu, Z.: End-to-end human pose and mesh reconstruction with transformers. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1954–1963 (2021). https://doi.org/10.1109/CVPR46437.2021.00199
14. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision (ECCV). pp. 740–755. Springer (2014). https://doi.org/10.1007/978-3-319-10602-1_48
15. Liu, D.C., Nocedal, J.: On the limited memory bfgs method for large scale optimization. Mathematical Programming **45**(1-3), 503–528 (1989). https://doi.org/10.1007/BF01589116
16. Qiu, H., Wang, C., Wang, J., Wang, N., Zeng, W.: Cross view fusion for 3D human pose estimation. In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 4342–4351 (2019). https://doi.org/10.1109/ICCV.2019.00444
17. Rhodin, H., Spörri, J., Katircioglu, I., Constantin, V., Meyer, F., Müller, E., Salzmann, M., Fua, P.: Learning monocular 3D human pose estimation from multi-view images. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8437–8446 (2018). https://doi.org/10.1109/CVPR.2018.00880
18. Shen, Z., Cen, Z., Peng, S., Shuai, Q., Bao, H., Zhou, X.: Learning human mesh recovery in 3D scenes. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 17038–17047 (2023). https://doi.org/10.1109/CVPR52729.2023.01634
19. Shuai, H., Wu, L., Liu, Q.: Adaptive multi-view and temporal fusing transformer for 3D human pose estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(4), 4122–4135 (2022). https://doi.org/10.1109/TPAMI.2022.3188716
20. Sun, J.J., Karashchuk, L., Dravid, A., Ryou, S., Fereidooni, S., Tuthill, J., Katsaggelos, A., Brunton, B.W., Gkioxari, G., Kennedy, A., Yue, Y., Perona, P.: BKinD-3D: self-supervised 3D keypoint discovery from multi-view videos. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9001–9010 (2023). https://doi.org/10.1109/CVPR52729.2023.00869
21. Tripathi, S., Müller, L., Huang, C.H.P., Taheri, O., Black, M.J., Tzionas, D.: 3D human pose estimation via intuitive physics. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4713–4725 (2023). https://doi.org/10.1109/CVPR52729.2023.00457
22. Usman, B., Tagliasacchi, A., Saenko, K., Sud, A.: Metapose: Fast 3D pose from multiple views without 3D supervision. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6759–6770 (2022). https://doi.org/10.1109/CVPR52688.2022.00664
23. Wandt, B., Rudolph, M., Zell, P., Rhodin, H., Rosenhahn, B.: Canonpose: Self-supervised monocular 3D human pose estimation in the wild. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13294–13304 (2021). https://doi.org/10.1109/CVPR46437.2021.01309
24. Zhang, Z., Wang, C., Qiu, W., Qin, W., Zeng, W.: Adafuse: Adaptive multiview fusion for accurate human pose estimation in the wild. International Journal of Computer Vision **129**, 703–718 (2021). https://doi.org/10.1007/s11263-020-01398-9
25. Zhou, X., Huang, Q., Sun, X., Xue, X., Wei, Y.: Towards 3D human pose estimation in the wild: a weakly-supervised approach. In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 398–407 (2017). https://doi.org/10.1109/ICCV.2017.51

26. Zhu, W., Ma, X., Liu, Z., Liu, L., Wu, W., Wang, Y.: Motionbert: A unified perspective on learning human motion representations. In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 15085–15099. IEEE (2023). https://doi.org/10.1109/ICCV51070.2023.01385