SynSP: Synergy of Smoothness and Precision in Pose Sequences Refinement

1. Motivation

1.1. Causes of Jitters at Joints

There are multiple reasons for joint jitters in a video, but they are all caused by errors generated by the human pose estimators. According to our knowledge, the errors leading to jitters can be categorized as follows:
Miss error: Failure in joint localization.
Swap error: Confusion between the joints of different individuals, resulting in incorrect joint identification.
Inversion error: Confusion between joints within a person due to their similarity, such as left and right hands.
Jitter error: Small localization error of a joint, which is the main cause of joint jitters.

Image-based pose estimators are more likely to have these errors due to a lack of temporal information compared to video-based pose estimators. However, SynSP can improve the performance of image-based estimators and make them comparable to the video-based pose estimators, as shown in Tab.2 of the manuscript.

1.2. Tension between Smoothness and Precision

Both the Smoothness and Precision objectives aim to minimize the discrepancy between predicted pose sequences (from pose estimators) and ground-truth sequences (from human annotation), with a focus on aligning the two in terms of position and acceleration dimensions, respectively. At first glance, these objectives appear to optimize the predicted sequences in a harmonious manner without conflict. However, due to their lack of knowledge about the exact position of the ground-truth data, both objectives attempt to adjust joint positions according to their own criteria, leading to a tension relationship. The greater the tension, the more uncertain the network is about the prediction.

This tension relationship prevents previous works from achieving better performance, but we translate this disagreement between smoothness and precision into the quality cue of predicted sequences with the Pose Quality Encoding module in SynSP, outperform previous works with much shorter length of input sequences and lower time delay.

1.3. Pearson Correlation Coefficient

The Pearson correlation coefficient [2], commonly known as Pearson's r, is a statistical measure widely used in scientific research to quantify the strength and direction of the linear relationship between two continuous variables. It possesses properties such as independence of scale, symmetry, and sensitivity to outliers, which is suitable for Diff1 and Diff2 (Please refer to Fig.1 of the manuscript for the definition of Diff1 and Diff2).

We support our motivation by calculating Pearson's r between Diff1 and Diff2 by the following procedure. First, we take the difference between predicted sequences and ground-truth sequences as Diff1, which is inversely proportional to the quality of predicted sequences. Then we train the base stage of SynSP with use SynSP while setting the ratio of λ_a to λ_p is 5 and 0.5 (please refer to Equ.4 of the manuscript for λ_a and λ_p), and use the difference of their outputs as Diff1. Finally,the Pearson's correlation coefficient between Diff1 and Diff2 is calculated to be 0.48. Moreover, the correlation is observed to be lower in easily recognizable areas such as the eye and nose when compared to more challenging joints like the shoulder and hip.

2. Experiments

2.1. Datasets Details

Human3.6M dataset [3] is a widely used benchmark for multi-view-based 3D human posture research, recording the performance of 5 female and 6 male subjects, under 4 different viewpoints, for training realistic human sensing systems and for evaluating the next generation of human pose estimation models and algorithms. This dataset has 1,559,752 frames for the training and 543,344 frames for the testing.
3DPW dataset [9] is an important outdoor dataset to evaluate the effectiveness of video-based methods with **occlusion** and **crowded** scenes, having 22,463 frames for the training and 35,219 frames for the testing with accurate 3D pose in challenging sequences, including walking in the city, going up-stairs, *etc*.
AIST++ dataset [5] is a new multi-modal dataset of **rapid** 3D dance motion and music, covering 10 dance genres with multi-view videos. This dataset has 5,858,138 frames for the training and 2,851,927 frames for the testing.
CMU-Mocap dataset [8] contains the vast majority of daily behaviors like walking, swimming and climbing, with the exception of flying and rock climbing mentioned on the official website. We split this dataset into 2,241,016 frames

for the training and 755,857 frames for the testing.

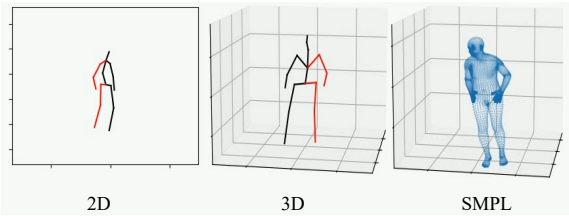

Figure 1. Illustration of 2D, 3D, and SMPL representations.

Pose Representations. As mentioned in Sec.2 of the manuscript, there are 2D, 3D, and SMPL representations for the human pose, and heir visualization is shown in Fig. 1.

2.2. Comparison with more methods.

There are also some methods[4, 10] with a super-long window length, and the comparison with them is shown in Tab. 1. Both DeciWatch [10] and HANet [4] employ a larger window size setting, which may potentially result in overfitting on normal-scale datasets such as H36m and 3DPW, while offering an advantage on large-scale datasets like AIST++.

Table 1. Comparison with DeciWatch and HANet.

Dataset	Method	WS	MPJPE	Accel
Human3.6M	FCN	1	54.6	19.2
	FCN+DeciWatch	1+**101**	52.8	1.5
	FCN+HANet	1+**101**	51.8	2.0
	FCN+SynSP	1+8	51.4	1.0
	FCN+SynSP*	1+8	**41.8**	**1.0**
3DPW	PARE	1	78.9	25.7
	PARE+DeciWatch	1+**101**	77.2	6.9
	PARE+HANet	1+**101**	77.1	6.8
	PARE+SynSP	1+8	**76.2**	**6.2**
AIST++	SPIN	1	107.7	33.8
	SPIN+DeciWatch	1+**101**	71.3	5.7
	SPIN+HANet	1+**101**	**69.2**	**5.4**
	SPIN+SynSP	1+8	84.6	6.1

2.3. SynSP's Other Outputs

Due to page limitations, the outputs produced by the Acceleration Branch and Position Branch will be shown and discussed here.

As shown in Tab. 2, we display the three outputs with Human3.6M + FCN [7], 3DPW + TCMR [1], and Human3.6M + PPT [6]. In the later two experiments, results from the branches are less accurate than the input, probably because it is more difficult to resolve the tension relationship between smoothness and precision at the base stage, and the network prefers to solve the acceleration errors first.

Table 2. Outputs from the two stages with different pose estimators. AB output represents the output from Acceleration Branch, and PB output represents the output from Position Branch.

Method	Type	MPJPE	Accel
FCN	input	54.6	19.2
	AB output	44.5	1.04
	PB output	43.0	1.09
	final output	**41.8**	**0.98**
TCMR	input	86.5	6.75
	AB output	88.5	**3.92**
	PB output	88.4	5.99
	final output	**86.1**	5.90
PPT	input	25.1	11.6
	AB output	29.9	1.14
	PB output	29.6	1.15
	final output	**21.5**	**1.10**

2.4. Analysis of Window Length.

The window size of 8 was adopted for pose sequences in the manuscript, as this setting is time-saving and sufficient to demonstrate the capability of SynSP. Here, we also explored the effect of window length on the performance of SynSP on the Human3.6M dataset with pose estimator FCN and the AIST++ dataset with pose estimator VIBE as shown in Tab. 3. However, increasing the window length did not result in the expected improvement for SynSP's performance. Both FCN and VIBE methods exhibited a similar trend for the evaluation, where MPJPE achieved its minimum value with a window length of 16, and the Accel error decreased with an increase in window length.

Considering Equ.4 of the loss function presented in the manuscript, we believe that the window length does not affect SynSP's attention towards precision and smoothness. Therefore, we can hypothesize two reasons for this trend: 1) an increase in window length indeed improve the performance of SynSP in Precision (MPJPE), since more temporal information can be referenced to correct the predicted poses; 2) self-attention module is friendly for NLP tasks with robust semantic values for input. However, for precise numerical values, as the number of tokens increases, the calculated attention in the self-attention module may be more easily distracted and result in errors. Since it is difficult to replace transformer-based SynSP with other backbone, *e.g.*, CNN and RNN, verifying these two hypotheses becomes challenging. We will conduct more experiments in the future to test them.

2.5. Exploration of SynSP's Structure

After the base stage, a straightforward approach would involve averaging these outputs and replicating the base stage

Table 3. Analysis of window size. WS denotes window size.

Dataset/Method	WS	MPJPE	PA-MPJPE	Accel
Human3.6M /FCN	1	54.55	42.20	19.17
	1+8	41.78	40.13	1.02
	1+16	40.11	31.27	1.13
	1+32	42.79	33.75	3.51
AIST++ /VIBE	16	106.90	72.84	31.60
	16+8	77.00	61.47	4.32
	16+16	68.87	53.94	7.39
	16+32	70.39	56.02	9.20

pipeline once again. However, there is no improvement with this operation as we test in Human3.6M + FCN [7] and 3DPW + TCMR [1]. We suspect that the output of base stage loses the original predicted position information, which affects the precision and makes the refinement stage useless compared with the baseline.

2.6. Acceleration for Sliding Window Average Algorithm

The Sliding Window Average Algorithm (SWAA) is essential for motion refinement methods to average poses of the same frame from multiple predictions, thereby optimizing one pose from multiple time points. SWAA can reduce the MPJPE error by 27.1% and the Accel error by 45.4% for SmoothNet and SynSP in the Human3.6M dataset with FCN estimators. As shown in Tab.4 of the manuscript, the delay mainly comes from the waiting time for real-time input video. For videos that have already been processed, the calculation speed of SWAA is particularly important. As shown in Fig. 2, we accelerate the SWAA through convolution operation and reduce the time of SWAA calculation to 0.374% of the original, further enhancing the efficiency of motion refinement methods.

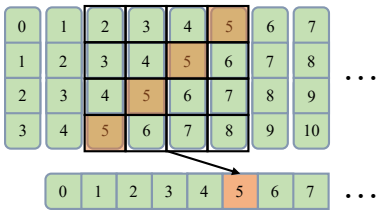

Figure 2. Illustration of Sliding Window Average Algorithm (SWAA) for parallel computing. Take the output window length of 4 as an example. The numbers in the figure represent the frame number of the output, and each column represents the output sequence of the network. It can be observed that the calculation of SWAA can be implemented in convolution operation (the black wire frame represents the convolution kernel).

2.7. Exploration of Other Input Combinations.

Limited by space, we discuss about other combinations of input pose sequences in detail here. Interestingly, we find that even if the input poses are not in the same frame and from different people, *i.e.*, a random combination, it can bring a small performance improvement as shown in Tab. 7 of the manuscript. During inference, input people with similar poses as much as possible at the same time, and the results will be further improved. We think that the network may use good pose sequences to repair those poses with jitters.

3. Limitations and Future Work.

Compared with SmoothNet which can be trained across multiple representations by its temporal structure, we cannot pretrain the SynSP among different representations. Although our method achieves better results, we still plan to modify the structure of the transformer to enable cross-representation training and inference. Moreover, we tend to extend our multi-view pose refinement to multi-person pose refinement, which has been verified by Tab. 7 in the manuscript.

References

[1] Hongsuk Choi, Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Beyond static features for temporally consistent 3d human pose and shape from a video. In *CVPR*, pages 1964–1973, 2021. 2, 3

[2] Israel Cohen, Yiteng Huang, Jingdong Chen, Jacob Benesty, Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. *Noise reduction in speech processing*, pages 1–4, 2009. 1

[3] 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. 1

[4] Kyung-Min Jin, Byoung-Sung Lim, Gun-Hee Lee, Tae-Kyung Kang, and Seong-Whan Lee. Kinematic-aware hierarchical attention network for human pose estimation in videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5725–5734, 2023. 2

[5] Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In *ICCV*, pages 13401–13412, 2021. 1

[6] Haoyu Ma, Zhe Wang, Yifei Chen, Deying Kong, Liangjian Chen, Xingwei Liu, Xiangyi Yan, Hao Tang, and Xiaohui Xie. Ppt: Token-pruned pose transformer formonocular andmulti-view human pose estimation. In *ECCV*, 2022. 2

[7] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *ICCV*, pages 2640–2649, 2017. 2, 3

[8] Carnegie Mellon University. Cmu graphics lab motion capture database: http://mocap.cs.cmu.edu/. 1

[9] 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. 1

[10] Ailing Zeng, Xuan Ju, Lei Yang, Ruiyuan Gao, Xizhou Zhu, Bo Dai, and Qiang Xu. Deciwatch: A simple baseline for 10× efficient 2d and 3d pose estimation. In *ECCV*, pages 607–624. Springer, 2022. 2