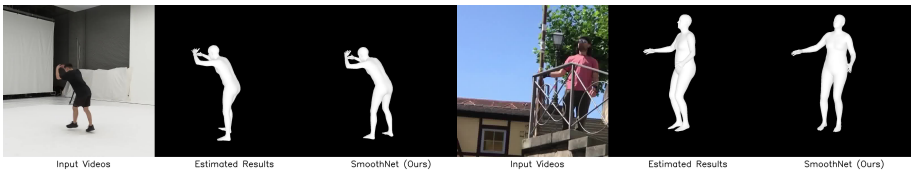
SmoothNet: A Plug-and-Play Network for Refining Human Poses in Videos

Ailing Zeng[1], Lei Yang[2], Xuan Ju[1], Jiefeng Li[3], Jianyi Wang[4], and Qiang Xu[1]

[1]The Chinese University of Hong Kong, [2] Sensetime Group Ltd.,
[3]Shanghai Jiao Tong University, [4]Nanyang Technological University
{alzeng, qxu}@cse.cuhk.edu.hk



(a) Rare Pose Refinement (b) Occluded Pose Refinement

Fig. 1: State-of-the-art human pose/shape estimators (e.g., VIBE [21]) suffer from severe jitters on videos containing rarely seen or occluded poses, resulting in untrustworthy perceptions. We propose a novel plug-and-play temporal refinement network, SmoothNet, significantly alleviating this problem. *Note that this is a video figure, best viewed with Acrobat Reader.*

Abstract. When analyzing human motion videos, the output jitters from existing pose estimators are highly-unbalanced with varied estimation errors across frames. Most frames in a video are relatively easy to estimate and only suffer from slight jitters. In contrast, for rarely seen or occluded actions, the estimated positions of multiple joints largely deviate from the ground truth values for a consecutive sequence of frames, rendering significant jitters on them.

To tackle this problem, we propose to attach a dedicated *temporal-only* refinement network to existing pose estimators for jitter mitigation, named SmoothNet. Unlike existing learning-based solutions that employ spatio-temporal models to co-optimize per-frame precision and temporal smoothness at all the joints, SmoothNet models the natural smoothness characteristics in body movements by learning the long-range temporal relations of every joint without considering the noisy correlations among joints. With a simple yet effective motion-aware fully-connected network, SmoothNet improves the temporal smoothness of existing pose estimators significantly and enhances the estimation accuracy of those challenging frames as a side-effect. Moreover, as a temporal-only model, a unique advantage of SmoothNet is its strong transferability across various types of estimators, modalities, and datasets. Comprehensive experiments on *five datasets* with *eleven popular backbone networks* across *2D and 3D pose estimation and body recovery tasks* demonstrate the efficacy of the proposed solution. Code is available at https://github.com/cure-lab/SmoothNet.

Keywords: Human Pose Estimation, Jitter, Temporal Models

1 Introduction

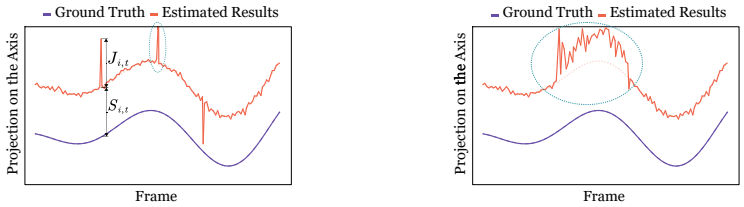
Human pose estimation has broad applications such as motion analysis and human-computer interaction. While significant advancements have been achieved with novel deep learning techniques (e.g., [5, 18, 22, 34, 38, 47]), the estimation errors for rarely seen or occluded poses are still relatively high.

When applying existing image-based pose estimators for video analysis, significant jitters occur on those challenging frames with large estimation errors as L1/L2 loss optimization is directionless. Moreover, they often last for a consecutive sequence of frames, causing untrustworthy perceptions (see *Estimated Results* in Figure 1). Various video-based pose estimators are proposed in the literature to mitigate this problem. Some use an end-to-end network that takes jitter errors into consideration [6, 19, 21, 35, 45], while the rest smooth the estimation results with spatial-temporal refinement models [20, 27, 43] or low-pass filters [4, 8, 11, 13, 17, 36, 41, 46]. These solutions, however, do not consider the highly-unbalanced nature of the jitters in the estimated poses, resulting in unsatisfactory performance.

On the one hand, existing learning-based solutions (including end-to-end and refinement networks) employ Spatio-temporal models to co-optimize per-frame precision and temporal smoothness at all the joints. This is a highly challenging task as jittering frames typically persist for a while, and they are associated with untrustworthy local temporal features and noisy correlation among the estimated joints. On the other hand, applying low-pass filters on each estimated joint with a long filtering window could reduce jitters to an arbitrarily small value. Nevertheless, such fixed temporal filters usually lead to considerable precision loss (e.g., over-smoothing) without prior knowledge about the distribution of human motions.

Motivated by the above, this work proposes to attach a dedicated *temporal-only* refinement network to existing 2D/3D pose estimators for jitter mitigation, named SmoothNet. Without considering the noisy correlations (especially on jittering frames) among estimated joint positions, SmoothNet models the natural smoothness characteristics in body movements in a data-driven manner. The main contributions of this paper include:

- We investigate the highly-unbalanced nature of the jitter problem with existing pose estimators and empirically show that significant jitters usually occur on a consecutive sequence of challenging frames with poor image quality, occlusion, or rarely seen poses.
- To the best of our knowledge, this is the first data-driven temporal-only refinement solution for human motion jitter mitigation. Specifically, we design simple yet effective fully-connected networks with a long receptive field to learn the temporal relations of every joint for smoothing, and we show it outperforms other temporal models such as temporal convolutional networks (TCNs) and vanilla Transformers.



(a) Sudden Jitters. (b) Long-term Jitters.

Fig. 2: Two kinds of jitters are caused by pose estimation errors. The horizontal coordinate represents frames, and the vertical coordinate shows joint position values. Output errors are composed of jitter errors J and biased errors S.

– As a temporal-only model, SmoothNet is a plug-and-play network with strong transferability across various types of estimators and datasets.

SmoothNet is conceptually simple yet empirically powerful. We conduct extensive experiments to validate its effectiveness and generalization capability on *five datasets*, *eleven backbone networks*, and *three modalities* (2D/3D position and 6D rotation matrix [54]). Our results show that SmoothNet improves the temporal smoothness of existing pose estimators significantly and enhances the estimation accuracy of the challenging frames as a side-effect, especially for those video clips with severe pose estimation errors and long-term jitters.

2 Preliminaries and Problem Definition

2.1 Human Pose Estimation

For video-based human pose estimation, L frames of a video $\mathbf{X}$ are inputs to the pose estimator f, and it outputs the estimated poses $\hat{\mathbf{Y}} \in \mathbb{R}^{L\times C}$, where $C = N \times D$. N is the number of keypoints associated with datasets, and D denotes the dimensions of each keypoint (2D [5,24,34,38] or 3D [29,35,47,48,50]). The above process can be simply formulated as $\hat{\mathbf{Y}} = f(\mathbf{X})$. The estimator is trained in a supervised manner with the labeled ground truth $\mathbf{Y} \in \mathbb{R}^{L\times C}$.
Key Evaluation Metrics. To evaluate the per-frame precision, the metric is the mean per joint position error (*MPJPE*). To measure the smoothness or jitter errors, the metric is the mean per joint acceleration error (*Accel*).

2.2 The Jitter Problem from Pose Estimators

An ideal pose estimator that outputs accurate joint positions would not suffer from jitters. In other words, the jitter problem is caused by pose estimation errors, which can be divided into two parts: the *jitter error* $\mathbf{J}$ between adjacent frames and the *biased error* $\mathbf{S}$ between the ground truth and smoothed poses.

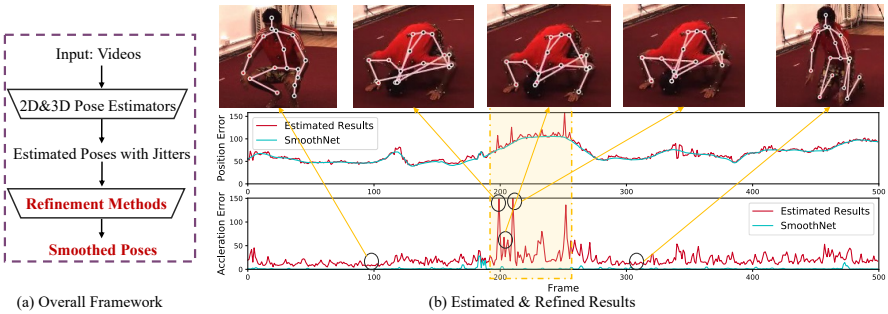


Fig. 3: SmoothNet is a plug-and-play temporal-only refinement network for jitter mitigation. (a) shows the refinement flow. (b) demonstrates the estimated results of a state-of-the-art estimator RLE [24] and how SmoothNet can improve the precision (upper curve) and smoothness (lower curve).

In Figure 2, we differentiate sudden jitters and long-term jitters based on the duration of jitters. Moreover, according to the degree of jitters, existing jitters can be split into small jitters caused by inevitably inaccurate and inconsistent annotations in the training dataset (e.g., [1, 26]) and large jitters caused by poor image quality, rare poses, or heavy occlusion.

State-of-the-art end-to-end estimators such as [21, 24] can output relatively accurate estimation results and small jitters for most frames (see Figure 3(b)). However, they tend to output large position errors when the video segments with rare/complex actions and these clips also suffer from significant jitters (e.g., from 200 to 250 frames in Figure 3(b)).

The jitter problem in human pose estimation is hence highly unbalanced. Generally speaking, sudden jitters are easy to remove with low-pass filters [3, 36, 46]. However, handling long-term jitters **J** is quite challenging because they usually entangle with ambiguous biased errors **S**.

3 Related Work and Motivation

3.1 Spatio-Temporal Models for Smoothing

Existing learning-based jitter mitigation solutions can be categorized into two types: end-to-end solutions and refinement networks after pose estimators. For the former category, various types of temporal models (e.g., gated recurrent units (GRUs) [6, 21, 27, 53], temporal convolutional networks (TCNs) [35, 47], and Transformers [44, 51]) are used for temporal feature extraction. Other end-to-end solutions employ regularizers or loss functions to constrain the temporal consistency across successive frames [19, 31, 33, 39, 43, 49]. Recent pose refinement works [15, 20, 43] take smoothness into consideration with spatial-temporal modeling. Specifically, Jiang *et al.* [15] designed a transformer-based network to

smooth 3D poses in sign language recognition. Kim *et al.* [20] propose a non-local attention mechanism with convolutions represented by *quaternions*. Considering occlusions on multi-person scenes, Vege *et al.* [43] conduct energy optimization with visibility scores to adaptively filter the keypoint trajectories.

Without considering the highly-unbalanced nature of the jitter problem in pose estimation, the above solutions still cannot yield a smooth sequence of poses. There are mainly two reasons for such unsatisfactory performance. On the one hand, multiple joint positions largely deviate from the ground-truth for consecutive frames with long-term jitters, and the extracted spatial/temporal features themselves are untrustworthy, rendering less effective smoothing results. On the other hand, co-optimizing the jitter error **J** and the biased error **S** is challenging, and we name it the *spatio-temporal optimization bottleneck.*

We conduct comprehensive experiments on the popular 3D skeleton-based methods [29, 35] and SMPL-based approaches [21, 22] under single-frame, multi-frame, and smoothness loss settings. Due to space limitations, we put the results in the supplementary materials and summarize our key findings here: (i). compared to single-frame models, spatial-temporal models have better performance. However, the reduction in jitter errors **J** are still unsatisfactory (e.g., *Accels* are reduced from 33mm to 27mm [21]); (ii). further adding an acceleration loss between consecutive frames or enhancing temporal modeling in the decoder design can benefit *Accels* but harm MPJPEs (increase biased errors **S**), due to the optimization bottleneck between per-frame precision and smoothness.

With the above, a temporal-only pose smoothing solution is more promising for jitter mitigation. Moreover, without using vastly different spatial information, such solutions have the potential to generalize across different datasets and motion modalities.

3.2 Low-Pass Filters for Smoothing

Low-pass filters are general smoothing solutions, and they are used for pose refinement in the literature. For example, moving averages [12] that calculate the mean values over a specified period of time can be used to smooth sudden jitters. Savitzky-Golay filter [36] uses a local polynomial least-squares function to fit the sequence within a given window size. Gaussian filter [46] modifies the input signal by convolution with a Gaussian function to obtain the minimum possible group delay. Recently, a One-Euro filter was proposed in [4] for real-time jitter mitigation with an adaptive cutoff frequency.

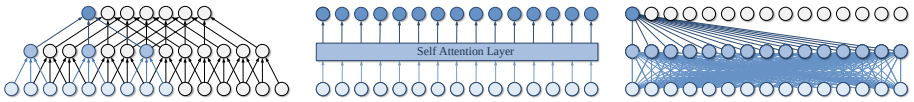
As a general temporal-only solution, low-pass filters can be applied to various pose refinement tasks without training. However, it inevitably faces the trade-off between jitters and lags, resulting in significant errors under long-term jitters.

Motivated by the limitations of existing works, we propose a novel data-driven temporal-only refinement solution for 2D/3D human pose estimation tasks, as detailed in the following section.

4 Method

Instead of fusing spatial and temporal features for pose refinement, we explore long-range temporal receptive fields to capture robust temporal relations for mitigating large and long-term jitters. Specifically, the proposed SmoothNet g learns from the noisy estimated poses $\hat{\mathbf{Y}} \in \mathbb{R}^{L \times C}$ generated by any pose estimators f.

The refinement function can be simply formulated as $\hat{\mathbf{G}} = g(\hat{\mathbf{Y}})$, where $\hat{\mathbf{G}} \in \mathbb{R}^{L \times C}$ is the smoothed poses.



(a) TCN block (b) Transformer block (c) SmoothNet block

Fig. 4: Temporal relation extraction with (a) TCN, (b) Transformer, and (c) FCN. The input circles mean T frames information in one spatial dimension.

4.1 Basic SmoothNet

Consider a fixed-length long sequence of the estimated joint positions, our objective is to capture temporal relations for smoothing. There are three popular temporal architectures that support long receptive fields without error accumulation, as illustrated in Fig. 4. Temporal convolutional networks (TCNs) [2] conduct local convolutions (e.g., kernel size is 3) at each layer and employ dilation operations with multiple layers to enlarge the receptive field. In contrast, Transformers [42] or fully-connected networks (FCNs) have global receptive fields at every layer, which can better tolerate long-term jitters than local convolutions in TCNs. While Transformers have become the *de facto* sequence-to-sequence models in many application scenarios [9,37,51,52], we argue it is less applicable to our problem when compared to FCNs. In a Transformer model, the critical issue is to extract the semantic correlations between any two elements in a long sequence (e.g., words in texts or 2D patches in images) with self-attention mechanisms. However, for pose refinement, we are more interested in modeling the continuity of motions on each joint (instead of point-wise correlations), which spans a continuous sequence of poses.

Consequently, in this work, we propose to use FCNs as the backbone of our SmoothNet design, which is position-aware and easy to train with abundant pose data from human motion videos. Additionally, according to the superposition of movements [10], a movement can be decomposed as several movements performed independently. Based on this principle, each axis i in channel C can be processed independently.

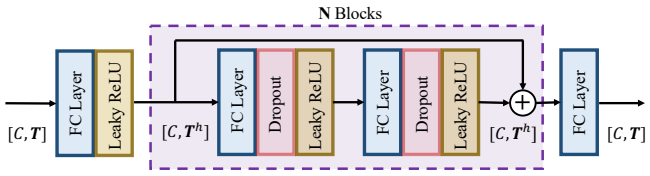


Fig. 5: A simple yet effective SmoothNet design.

The proposed network is shown in Figure 5, where we construct multiple FC layers with N residual connected blocks along the temporal axis. The computation of each layer can be formulated as follows.

$$\hat{Y}_{i,t}^{l+1} = \sigma(\sum_{t=1}^{T} w_t^l * \hat{Y}_{i,t}^l + b^l), \tag{1}$$

where w_t^l and b^l are learnable weights and bias at the t_{th} frame and they are shared among different i_{th} axis, respectively. σ is the non-linear activation function (LeakyReLU is chosen by default). To process $\hat{\mathbf{Y}}$ with SmoothNet, we adopt a sliding-window scheme similar to filters [23, 36, 46], where we first extract a chunk with size T, yield refined results thereon, and then move to the next chunk with a step size s ($s \leq T$), preventing a loss of the last few frames.

4.2 Motion-aware SmoothNet

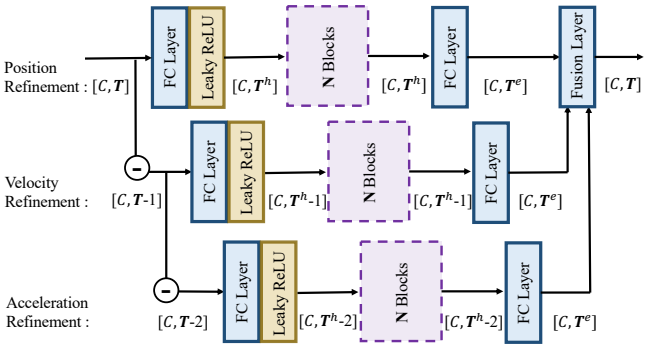


Fig. 6: The motion-aware SmoothNet design. It explicitly models the velocity and acceleration with adjacent frames to achieve better pose refinement.

Our goal is to capture jitter patterns and reduce jitter errors **J**, which mainly present as acceleration errors. It is straightforward to model acceleration explicitly in addition to position. Accordingly, we further inject the movement function

into our network, *i.e.*, velocity and acceleration. Given the prior with physical meaning, it is beneficial to leverage first-order and second-order motion information, making the learning process converge better and faster than the Basic SmoothNet. Specifically, given the input $\hat{\mathbf{Y}}$, we first compute the velocity and acceleration (subtract by two consecutive frames) for each axis i, according to the Equation 2.

$$\hat{V}_{i,t} = \hat{Y}_{i,t} - \hat{Y}_{i,t-1}, \quad \hat{A}_{i,t} = \hat{V}_{i,t} - \hat{V}_{i,t-1}. \tag{2}$$

As shown in Figure 6, the top branch is the baseline stream to refine noisy positions $\hat{\mathbf{Y}}$. The other two branches input the corresponding noisy velocity $\hat{\mathbf{V}}$ and acceleration $\hat{\mathbf{A}}$. To capture the long-term temporal cue, we also employ Equation 1 to refine the velocity and acceleration. All branches consist of the same FC layers and blocks. Then, we concatenate the top embedding of three branches to aggregate information from different order of motions and perform a linear fusion layer to obtain the final refined poses $\hat{\mathbf{G}}$. Similar to the basic scheme in Section 4.1, this motion-aware scheme also works in a sliding-window manner to process the whole input sequence.

4.3 Loss Function

SmoothNet aims to minimize both position errors and acceleration errors during training, and these objective functions are defined as follows.

$$L_{pose} = \frac{1}{T \times C} \sum_{t=0}^{T} \sum_{i=0}^{C} |\hat{G}_{i,t} - Y_{i,t}|, \tag{3}$$

$$L_{acc} = \frac{1}{(T-2) \times C} \sum_{t=0}^{T} \sum_{i=0}^{C} |\hat{G}''_{i,t} - A_{i,t}|, \tag{4}$$

where $\hat{G}''_{i,t}$ is the computed acceleration from predicted pose $\hat{G}_{i,t}$ and $A_{i,t}$ is the ground-truth acceleration. We simply add L_{pose} and L_{acc} as our final target.

5 Experiment

We validate the effectiveness of the proposed SmoothNet and show quantitative results in the following sections. Due to space limitations, we leave more analysis, discussions, and demos to the *supplementary material*. For more experimental details, please refer to the code.

5.1 Experimental Settings

Backbones. We validate the generalization ability on both smoothness and precision of the proposed SmoothNet covering three related tasks and several corresponding backbone models. For 2D pose estimation, we use Hourglass [34], CPN [5], HRNet [38] and RLE [24]; for 3D pose estimation, we implement

FCN [29], RLE [24], TPoseNet [43] and VPose [35]; in terms of body recovery, we test on SPIN [22], EFT [16], VIBE [21] and TCMR [6].

Training sets. To prepare training data, we first save the outputs of existing methods, including estimated 2D positions, 3D positions, or SMPL parameters. Then, we take these outputs as the inputs of SmoothNet and use the corresponding ground-truth data as the supervision to train our model. In particular, we use the outputs of FCN on Human3.6M, SPIN on 3DPW [28], and VIBE on AIST++ [25] to train SmoothNet.

Testing sets. We validate SmoothNet on five dataset: Human3.6M [14], 3DPW [28], MPI-INF-3DHP [30], AIST++ [25, 40] and MuPoTS-3D [32] datasets.

Evaluation Metrics. To measure the jitter errors, we follow the related works [6, 19, 21] to adopt *Accel*. This is measured in *mm/ frame*^{2} for 3D poses and *pixel/ frame*^{2} for 2D poses. To evaluate the precision for each frame, besides *MPJPE*, the *Procrustes Analysis MPJPE (PA-MPJPE)* is another commonly used metric, where it removes effects on the inherent scale, rotation, and translation issues. For the 3D pose, the unit is *mm*. For the 2D pose, we simply use *pixel* in an image to validate the accurate localization precision.

Implementation Details The basic SmoothNet is an eight-layer model including the first layer, three cascaded blocks (N=3), and the last layer as a decoder. The motion-aware SmoothNet contains three parallel branches with the first layer, one cascaded block, and the last layer for each branch. The input window size T is 32 and the moving step size s is 1. In addition, we use the sliding window average algorithm [23] based on smoothed results to avoid frame drop and reduce spikes. The parameters of SmoothNet is 0.33M, and the average inference time is less than 1.3k fps on a CPU and 46.8k on an A100-SXM4-40GB GPU.

5.2 Comparison with Existing Solutions

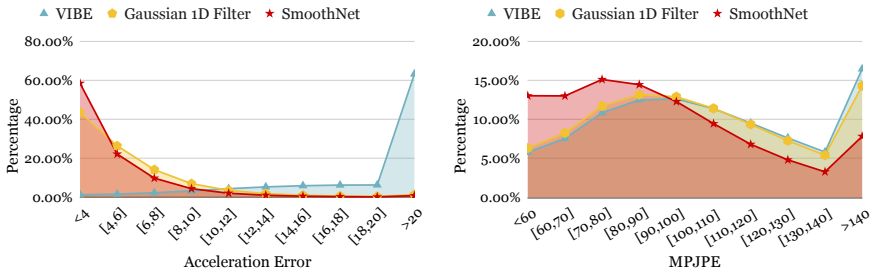
Comparison with Filters We compare SmoothNet against three commonly used filters on the AIST++ dataset with pose estimator VIBE [21]. Experimental results are shown in Table 1. As can be observed, SmoothNet achieves the best performance, and it reduces *Accel* by 86.88% and *MPJPE* by 8.82% compared to the original pose estimation results. Since we can easily trade off smoothness and lag in filter designs, there could be a large set of solutions with different *Accel* and *MPJPE* values. In this table, we present two possible solutions with the greedy search: one with comparable *Accel* with SmoothNet and the other with the minimum *MPJPE*.

As a data-driven approach, SmoothNet effectively learns the motion distribution of the complex movements in the dataset, resulting in much better *MPJPEs* values, especially when *Accel* is comparable. Among the three filters, the one-Euro filter shows inferior performance, and we attribute it to the real-time frame-by-frame jitter mitigation strategy used in it. Additionally, as SmoothNet can benefit from GPU acceleration, it yields a much faster inference speed than filters (marked by *).

Table 1: **Comparison SmoothNet with widely-used filters on AIST++ [25].** The upper table with filters shows their lowest *MPJPEs*, and the lower table is when their *Accels* are comparable to ours. * means the inference speed is tested on a GPU.

	Method	Accel	MPJPE	PA-MPJPE	Test FPS
Human Mesh Recovery	VIBE [21]	31.64	106.90	72.84	-
	w/ One-Euro [4]	10.82	108.55	74.67	2.31k
	w/ Savitzky-Golay [36]	5.84	105.80	72.15	31.22k
	w/ Gaussian1d [46]	4.95	103.42	71.11	37.45k
	w/ One-Euro [4]	4.67	135.71	103.22	2.43k
	w/ Savitzky-Golay [36]	4.36	118.25	85.39	30.19k
	w/ Gaussian1d [46]	4.47	105.71	71.49	38.21k
	w/ Ours	**4.15**	**97.47**	**69.67**	1.30k/**46.82k***

We further plot the *MPJPE* and *Accel* distribution of the original pose output from VIBE, VIBE with a Gaussian filter, and VIBE with SmoothNet in Figure 7. As can be observed, 98.7% of VIBE's original *Accel* output falls above 4 $mm/frame^2$. With Gaussian filter and SmoothNet, this percentage decreases to 56.5% and 41.6%, respectively. As for MPJPEs, 5.78% of VIBE's outputs are smaller than 60 *mm* and 16.43% estimated poses are larger than 140 *mm*. Gaussian filter increase the former proportion to 6.31% and decrease the latter proportion to 14.27%, improving precision slightly by removing some *small* jitters and *sudden* jitters. In contrast, SmoothNet can increase the former percentage to 13.01% and decrease the latter percentage to 7.82% (a relative 45.2% reduction). We attribute the much higher performance of our solution to the fact that SmoothNet can relieve large and long-term jitters effectively, thanks to its data-driven modeling of the smoothness characteristics in body movements.



(a) *Accel* Distribution (b) *MPJPE* Distribution

Fig. 7: Comparison of smoothness and precision distributions on AIST++.

5.3 Refinement Results for Existing Methods

As a plug-and-play network, SmoothNet can be combined with any existing pose estimators. Here, we show the results on both skeleton-based methods and SMPL-based methods.

Table 2: **Results of SmoothNet attached to 2D and 3D pose estimators on Human3.6M dataset.** * is spatio-temporal backbones.

	Method	*Accel*	*MPJPE*	*PA-MPJPE*	*MPJPE-1%*	*Accel-1%*
2D Pose Estimation	Hourglass [34]	1.54	9.42	7.64	55.81	2.71
	Hourglass w/ours	**0.15**	**9.25**	**7.57**	**55.50**	**0.23**
	CPN [5]	2.91	6.67	5.18	51.86	4.17
	CPN w/ours	**0.14**	**6.45**	**4.96**	**51.65**	**0.22**
	HRNet [38]	1.01	4.59	4.19	18.16	3.55
	HRNet w/ours	**0.13**	**4.54**	**4.13**	**16.98**	**0.26**
	RLE [24]	0.90	5.14	4.82	16.67	2.28
	RLE w/ours	**0.13**	**5.21**	**4.78**	**16.16**	**0.19**
3D Pose Estimation	FCN [29]	19.17	54.55	42.20	161.00	40.03
	FCN w/ours	**1.03**	**52.72**	**40.92**	**151.08**$\downarrow_{6.2\%}$	**1.52**$\downarrow_{96.2\%}$
	RLE [24]	7.75	48.87	38.63	139.04	16.54
	RLE w/ours	**0.90**	**48.27**	**38.13**	**136.70**$\downarrow_{1.7\%}$	**1.01**$\downarrow_{93.9\%}$
	VPose [35] (T=27)*	3.53	50.13	39.13	153.87	7.95
	VPose (T=27)* w/ours	**0.88**	**50.04**	**39.04**	**153.29**$\downarrow_{3.8\%}$	**0.94**$\downarrow_{88.2\%}$
	VPose (T=81)*	3.06	48.97	38.27	149.97	6.52
	VPose (T=81)* w/ours	**0.87**	**48.89**	**38.21**	**149.57**$\downarrow_{0.3\%}$	**0.85**$\downarrow_{87.0\%}$
	VPose (T=243)*	2.82	48.11	37.71	150.25	6.01
	VPose (T=243)* w/ours	**0.87**	**48.05**	**37.66**	**149.88**$\downarrow_{0.2\%}$	**0.83**$\downarrow_{86.2\%}$

All estimation results are re-implemented or tested by us for fair comparisons.

2D and 3D Pose Estimation In Table 2, we compare the results of skeleton-based methods on the Human3.6M dataset. The *Accel* of all the backbones followed by our pose refinement method is significantly reduced, and *MPJPE* is also reduced to some extent. Specifically, *Accel* and *MPJPE* are reduced to a greater extent for the single-frame networks. Also, we observe that the refined *Accel* is similar with different backbones, indicating that SmoothNet can effectively remove different kinds of jitters in pose estimation. Since SmoothNet is only trained with FCN-Human3.6M, the improvements on FCN [29] is larger than other backbones with 94.6%, 3.4% and 3.0% reduction in *Accel*, *MPJPE* and *PA-MPJPE*, respectively. To explore the impact on significant biased errors and long-term jitters, we further calculate the largest 1% of *MPJPE* (*MPJPE-1%*) and their corresponding *Accel* (*Accel-1%*) as the worst 1% estimated poses for each backbone. On average, the estimated *Accel* on these poses are decreased by

about 90%. In particular, we could achieve an 6.2% improvement on the trained backbone FCN with *MPJPE* reduced from 161.00*mm* to 151.08*mm*. This is because significant position errors are usually accompanied by long-term and large jitters, and SmoothNetcan reduce them as a side-effect during smoothing. Moreover, the above results across backbones also validate the generalization capability of SmoothNet.

Human Mesh Recovery In Table 3, we give results of SMPL-based methods for body recovery on 3DPW [28], MPI-INF-3DHP [30], and Human3.6M dataset [14]. SmoothNet is trained with the pose outputs from SPIN [22]. We test its performance across multiple backbone networks.

Table 3: **Results of SmoothNet attached to human mesh recovery models** on 3DPW [28], MPI-INF-3DHP [30], and Human3.6M [14] dataset. * is spatio-temporal backbones.

Method	3DPW			MPI-INF-3DHP			Human3.6M		
	Accel	*MPJPE*	*PA-MPJPE*	*Accel*	*MPJPE*	*PA-MPJPE*	*Accel*	*MPJPE*	*PA-MPJPE*
SPIN [22]	30.8	87.6	53.3	28.5	100.2	61.4	18.6	68.5	46.5
SPIN w/ours	**5.5**	**86.7**	**52.7**	**6.5**	**92.9**	**60.2**	**2.8**	**67.5**	**46.3**
VIBE* [21]	23.2	83.0	52.0	22.3	91.9	58.9	15.8	78.1	53.7
VIBE* w/ours	**6.0**	**81.5**	**51.7**	**6.5**	**87.6**	**58.8**	**2.9**	**77.2**	**53.4**
TCMR* [6]	6.8	**86.5**	**52.7**	8.0	92.6	**58.2**	3.8	**73.6**	**52.0**
TCMR w/MEVA* [27]	6.2	88.7	55.0	-	-	-	3.1	77.2	55.4
TCMR* w/ours	**6.0**	**86.5**	53.0	**6.5**	**88.9**	58.9	**2.8**	73.9	52.1

All estimation results are re-implemented or tested by us for fair comparisons.

Overall, our method has a consistent improvement in smoothness and precision. Specifically, SmoothNet can reduce *Accel* on SPIN and VIBE by a large margin. Compared to the original estimated poses from SPIN, our method improves by about 82.1% and 1.0% on *Accel* and *MPJPE*, respectively. For the TCMR backbone, since it has used some smoothing strategies in its models, their original *Accel* is relatively small. However, the first and last few frames could not be smoothed out with their method. Our model can relieve such jitters and further enhance its performance. Moreover, we add the post-processing slerp filter to minimize Euclidean distance on quaternion from MEVA [27] on top of TCMR backbone. The filter can improve *Accel*, but causes over-smoothness, leading to higher position errors.

5.4 Ablation Study

Comparisons on Temporal Models. To further validate the capability of the proposed FCN-based temporal model SmoothNet, we compare it with (i). traditional Gaussian1d filter; (ii). temporal convolutional networks [2] with a small kernel size (here is 3) in each layer, with 6, 8, 10 layers to obtain 27, 81,

Table 4: **Comparison results with different temporal models on VIBE-AIST++.** × is to use overlapped sliding-window scheme, which is used in SmoothNet by default. Ours⋆ is the same model with a non-overlapping sliding window.

Method	Gaussian1d	TCN(27)	TCN(81)	TCN(81)×	TCN(243)	Trans.×	Ours⋆	Ours×
Accel	4.95	14.46	11.84	8.71	10.07	6.15	5.45	**4.15**
MPJPE	103.42	103.53	101.17	99.54	99.76	99.30	98.34	**97.47**
PA-MPJPE	71.11	72.99	72.30	71.80	71.92	71.89	71.02	**69.67**

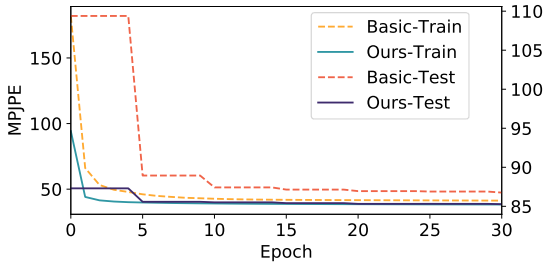


Fig. 8: Impact of model designs from the training and testing precision curves.

and 243 final receptive fields, respectively; (iii). self-attention-based Transformer (Trans.); (iv). TCN (81)× with overlapped sliding window scheme to enhance the output quality. Same as the Section 5.2, we use inputs from VIBE-AIST++.

Results are shown in Table 4, which indicates (i). the performance of TCN improves with increased receptive fields; (ii). the *Accel* of TCNs are worse than that of the filter [46], implying local aggregation of noisy poses with the shared kernels cannot handle large and long-term jitters well; (iii) the *MPJPE* of TCNs and Transformers are lower than that of the filter, indicating learning-based methods can further reduce biased errors S with learning the noisy pose prior; (iv) Transformer achieves a good balance between *Accel* and *MPJPE* with the global receptive field at each layer, but not as good as SmoothNet. We attribute it to the unnecessary self-attention operations for the pose refinement task, which is no guarantee to model the smoothness pattern well. Lastly, our method show superiority in all metrics even without overlapped sliding window scheme. Note that the sliding window scheme can relieve the spikes at the junction of two sliding windows, especially when MPJPE is huge.

Comparison between the Two Proposed Models. To capture the long-range temporal relations from noisy estimated pose sequences, we first propose a simple model with the residual fully connected network on temporal dimension, named *basic* SmoothNet. To further improve performance, we design a motion-aware temporal network as the SmoothNet in Sec. 4.2. Figure 8 illustrates the training and testing precision curves of these two models on 3DPW. We can

observe that (i) *basic* model tends to somewhat overfit; (ii) SmoothNet fits better and obtain slightly lower position errors. In comprehensive studies, we summarize the motion-aware SmoothNet can fit better than the basic one, while the basic one can obtain impressive results with its simple design.

Impact of Window Size. The window size W will largely impact of smoothness from previous sliding-window-based methods [6, 20, 21, 36]. We demonstrate the effects on different window sizes from 2 to 256 frames in Table 5. As the window size becomes longer, the *Accel* decreases consistently, but the *MPJPE* and *PA-MPJPE* initially decrease, then begin to increase slightly, indicating that when the size exceeds 64 frames, the results of the three metrics tend to be saturated. Therefore, 64 frames can be suitable to balance the smoothness and precision.

Table 5: **Impact of window size W on VIBE-AIST++ [25].**

W	VIBE	2	8	16	32	64	128	256
Accel	31.63	17.89	5.76	4.54	4.15	4.07	4.04	**4.03**
MPJPE	106.90	102.57	99.98	98.62	97.47	97.06	**93.20**	94.89
PA-MPJPE	72.84	71.48	70.51	69.85	**69.67**	69.89	70.57	71.52

6 Conclusion

In this work, we propose SmoothNet, a simple yet effective pose refinement network to improve the temporal smoothness and per-frame precision of existing pose/body estimators. Compared to existing solutions, SmoothNet can deal with long-term significant jitters, which often occur with rare/complex poses, as verified with comprehensive experiments on a large number of backbone networks, commonly modalities and datasets.

Broader Impact: SmoothNet is a temporal-only model targeting at removing various jitters, which takes advantage of the continuity of human motion, and generalizes well across backbones, modalities, and even datasets. Accordingly, this idea could be applied to other related tasks, such as whole-body estimation, pose tracking, and multi-object tracking, to further improve their smoothness and precision. Moreover, SmoothNet could potentially provide a smoothness prior over human motion, which is complementary to pose prior VPoser [7] and motion prior MPoser [21].

Limitation and Future Work: SmoothNet is a sliding-window-based model, which limits its use in real-time systems since we can not aggregate future poses to refine the historical poses. A real-time and accurate refinement model will be beneficial for online applications. We leave them for future work.

Acknowledgement. This work is supported in part by Shenzhen-Hong Kong-Macau Science and Technology Program (Category C) of Shenzhen Science Technology and Innovation Commission under Grant No. SGDX2020110309500101.

References

1. Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B.: 2d human pose estimation: New benchmark and state of the art analysis. In: Proceedings of the IEEE Conference on computer Vision and Pattern Recognition. pp. 3686–3693 (2014) 4
2. Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. ArXiv **abs/1803.01271** (2018) 6, 12
3. Brownrigg, D.R.: The weighted median filter. Communications of the ACM **27**(8), 807–818 (1984) 4
4. Casiez, G., Roussel, N., Vogel, D.: 1€ filter: a simple speed-based low-pass filter for noisy input in interactive systems. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 2527–2530 (2012) 2, 5, 10
5. Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., Sun, J.: Cascaded pyramid network for multi-person pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7103–7112 (2018) 2, 3, 8, 11
6. Choi, H., Moon, G., Chang, J.Y., Lee, K.M.: Beyond static features for temporally consistent 3d human pose and shape from a video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1964–1973 (2021) 2, 4, 9, 12, 14
7. Choutas, V., Pavlakos, G., Bolkart, T., Tzionas, D., Black, M.J.: Monocular expressive body regression through body-driven attention. In: European Conference on Computer Vision. pp. 20–40. Springer (2020) 14
8. Coskun, H., Achilles, F., DiPietro, R.S., Navab, N., Tombari, F.: Long short-term memory kalman filters: Recurrent neural estimators for pose regularization. 2017 IEEE International Conference on Computer Vision (ICCV) pp. 5525–5533 (2017) 2
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020) 6
10. Fischman, M.G.: Programming time as a function of number of movement parts and changes in movement direction. Journal of Motor Behavior **16**(4), 405–423 (1984) 6
11. Gauss, J.F., Brandin, C., Heberle, A., Löwe, W.: Smoothing skeleton avatar visualizations using signal processing technology. SN Computer Science **2**(6), 1–17 (2021) 2
12. Hunter, J.S.: The exponentially weighted moving average. Journal of quality technology **18**(4), 203–210 (1986) 5
13. Hyndman, R.J.: Moving averages. (2011) 2
14. 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) 9, 12
15. Jiang, T., Camgoz, N.C., Bowden, R.: Skeletor: Skeletal transformers for robust body-pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3394–3402 (2021) 4
16. Joo, H., Neverova, N., Vedaldi, A.: Exemplar fine-tuning for 3d human model fitting towards in-the-wild 3d human pose estimation. In: 2021 International Conference on 3D Vision (3DV). pp. 42–52. IEEE (2021) 9

17. Kalman, R.E.: A new approach to linear filtering and prediction problems (1960) 2
18. Kanazawa, A., Black, M.J., Jacobs, D.W., Malik, J.: End-to-end recovery of human shape and pose. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7122–7131 (2018) 2
19. Kanazawa, A., Zhang, J.Y., Felsen, P., Malik, J.: Learning 3d human dynamics from video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5614–5623 (2019) 2, 4, 9
20. Kim, D.Y., Chang, J.Y.: Attention-based 3d human pose sequence refinement network. Sensors **21**(13), 4572 (2021) 2, 4, 5, 14
21. Kocabas, M., Athanasiou, N., Black, M.J.: Vibe: Video inference for human body pose and shape estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5253–5263 (2020) 1, 2, 4, 5, 9, 10, 12, 14
22. Kolotouros, N., Pavlakos, G., Black, M.J., Daniilidis, K.: Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2252–2261 (2019) 2, 5, 9, 12
23. Lee, C.H., Lin, C.R., Chen, M.S.: Sliding-window filtering: an efficient algorithm for incremental mining. In: Proceedings of the tenth international conference on Information and knowledge management. pp. 263–270 (2001) 7, 9
24. Li, J., Bian, S., Zeng, A., Wang, C., Pang, B., Liu, W., Lu, C.: Human pose regression with residual log-likelihood estimation. In: ICCV (2021) 3, 4, 8, 9, 11
25. Li, R., Yang, S., Ross, D.A., Kanazawa, A.: Ai choreographer: Music conditioned 3d dance generation with aist++. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 13401–13412 (October 2021) 9, 10, 14
26. 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. pp. 740–755. Springer (2014) 4
27. Luo, Z., Golestaneh, S.A., Kitani, K.M.: 3d human motion estimation via motion compression and refinement. In: Proceedings of the Asian Conference on Computer Vision (2020) 2, 4, 12
28. von Marcard, T., Henschel, R., Black, M.J., Rosenhahn, B., Pons-Moll, G.: Recovering accurate 3d human pose in the wild using imus and a moving camera. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 601–617 (2018) 9, 12
29. Martinez, J., Hossain, R., Romero, J., Little, J.J.: A simple yet effective baseline for 3d human pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2640–2649 (2017) 3, 5, 9, 11
30. Mehta, D., Rhodin, H., Casas, D., Fua, P., Sotnychenko, O., Xu, W., Theobalt, C.: Monocular 3d human pose estimation in the wild using improved cnn supervision. In: 2017 international conference on 3D vision (3DV). pp. 506–516. IEEE (2017) 9, 12
31. Mehta, D., Sotnychenko, O., Mueller, F., Xu, W., Elgharib, M., Fua, P., Seidel, H.P., Rhodin, H., Pons-Moll, G., Theobalt, C.: Xnect: Real-time multi-person 3d motion capture with a single rgb camera. ACM Transactions on Graphics (TOG) **39**(4), 82–1 (2020) 4
32. Mehta, D., Sotnychenko, O., Mueller, F., Xu, W., Sridhar, S., Pons-Moll, G., Theobalt, C.: Single-shot multi-person 3d pose estimation from monocular rgb. 2018 International Conference on 3D Vision (3DV) pp. 120–130 (2018) 9

33. Mehta, D., Sridhar, S., Sotnychenko, O., Rhodin, H., Shafiei, M., Seidel, H.P., Xu, W., Casas, D., Theobalt, C.: Vnect: Real-time 3d human pose estimation with a single rgb camera. ACM Transactions on Graphics (TOG) **36**(4), 1–14 (2017) 4
34. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: European conference on computer vision. pp. 483–499. Springer (2016) 2, 3, 8, 11
35. Pavllo, D., Feichtenhofer, C., Grangier, D., Auli, M.: 3d human pose estimation in video with temporal convolutions and semi-supervised training. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7753–7762 (2019) 2, 3, 4, 5, 9, 11
36. Press, W.H., Teukolsky, S.A.: Savitzky-golay smoothing filters. Computers in Physics **4**(6), 669–672 (1990) 2, 4, 5, 7, 10, 14
37. So, D., Le, Q., Liang, C.: The evolved transformer. In: International Conference on Machine Learning. pp. 5877–5886. PMLR (2019) 6
38. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5693–5703 (2019) 2, 3, 8, 11
39. Tripathi, S., Ranade, S., Tyagi, A., Agrawal, A.: Posenet3d: Learning temporally consistent 3d human pose via knowledge distillation. In: 2020 International Conference on 3D Vision (3DV). pp. 311–321. IEEE (2020) 4
40. Tsuchida, S., Fukayama, S., Hamasaki, M., Goto, M.: Aist dance video database: Multi-genre, multi-dancer, and multi-camera database for dance information processing. In: ISMIR. pp. 501–510 (2019) 9
41. Van Loan, C.: Computational frameworks for the fast Fourier transform. SIAM (1992) 2
42. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017) 6
43. Véges, M., Lőrincz, A.: Temporal smoothing for 3d human pose estimation and localization for occluded people. In: International Conference on Neural Information Processing. pp. 557–568. Springer (2020) 2, 4, 5, 9
44. Wan, Z., Li, Z., Tian, M., Liu, J., Yi, S., Li, H.: Encoder-decoder with multi-level attention for 3d human shape and pose estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13033–13042 (2021) 4
45. Wang, J., Yan, S., Xiong, Y., Lin, D.: Motion guided 3d pose estimation from videos. ArXiv **abs/2004.13985** (2020) 2
46. Young, I.T., Van Vliet, L.J.: Recursive implementation of the gaussian filter. Signal processing **44**(2), 139–151 (1995) 2, 4, 5, 7, 10, 13
47. Zeng, A., Sun, X., Huang, F., Liu, M., Xu, Q., Lin, S.C.F.: Srnet: Improving generalization in 3d human pose estimation with a split-and-recombine approach. In: ECCV (2020) 2, 3, 4
48. Zeng, A., Sun, X., Yang, L., Zhao, N., Liu, M., Xu, Q.: Learning skeletal graph neural networks for hard 3d pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision (2021) 3
49. Zhang, S., Zhang, Y., Bogo, F., Pollefeys, M., Tang, S.: Learning motion priors for 4d human body capture in 3d scenes. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11343–11353 (2021) 4
50. Zhao, L., Peng, X., Tian, Y., Kapadia, M., Metaxas, D.N.: Semantic graph convolutional networks for 3d human pose regression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3425–3435 (2019) 3

51. Zheng, C., Zhu, S., Mendieta, M., Yang, T., Chen, C., Ding, Z.: 3d human pose estimation with spatial and temporal transformers. arXiv preprint arXiv:2103.10455 (2021) 4, 6
52. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W.: Informer: Beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of AAAI (2021) 6
53. Zhou, K., Bhatnagar, B.L., Lenssen, J.E., Pons-Moll, G.: Toch: Spatio-temporal object correspondence to hand for motion refinement. In: arXiv (May 2022) 4
54. Zhou, Y., Barnes, C., Lu, J., Yang, J., Li, H.: On the continuity of rotation representations in neural networks. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 5738–5746 (2019) 3