Supplementary Material
Improving the Robustness of 3D Human Pose Estimation: A Benchmark Dataset and Learning from Noisy Input

Trung-Hieu Hoang[1] Mona Zehni [1] Huy Phan[2] Duc Minh Vo[3] Minh N. Do[1]
[1]University of Illinois at Urbana-Champaign, USA
[2]VinUniversity, Ha Noi, Vietnam [3]The University of Tokyo, Japan
{hthieu,mzehni2,minhdo}@illinois.edu, 20huy.pn@vinuni.edu.vn, vmduc@nlab.ci.i.u-tokyo.ac.jp

1. Temporal Additive Gaussian Noise

Algorithm 1 elaborates upon our proposed Temporal Additive Gaussian Noise (TAGN) strategy.

Algorithm 1 Temporal Additive Gaussian Noise (TAGN)

Input: Keypoints sequence $\mathbf{x}(t) \in \mathbb{R}^{|\mathcal{J}|\times 2}$, temporal distortion ratio $k\%$, joint distortion ratio $p\%$, noise variance σ^2, set of frame indices $\mathcal{T}$, set of joints indices $\mathcal{J}$

```
// Uniformly select k% of frames
```
$\tilde{\mathcal{T}} \leftarrow$ Choice$(\mathcal{T}, k)$ `// Set of distorted frames`
$\tilde{\mathbf{x}} \leftarrow \mathbf{x}$ `// Initialize`
for $t \in \tilde{\mathcal{T}}$ **do**
  `// Uniformly select p% of joints`
  $\tilde{\mathcal{J}}_t \leftarrow$ Choice$(\mathcal{J}, p)$ `// Set of distorted joints at frame t`
  for $j \in \tilde{\mathcal{J}}_t$ **do**
    `// Add noise`
    $\tilde{\mathbf{x}}_j(t) \leftarrow \mathbf{x}_j(t) + \varepsilon,\ \varepsilon \sim \mathcal{N}(0, \sigma^2)$
  end
end
`// Return distorted keypoints`
Output: $\tilde{\mathbf{x}}$

2. Implementation Details

2.1. Video Corruption Operators

Guided-patch Erasing (GPE): Given a video, the masking patch positions are selected such that they have the most overlap with a selected keypoints' trajectory. We perform K-means clustering (K is randomly selected from $\{2, 3, 4\}$) over the set of all aggregated keypoint locations throughout the input video to find the centroids of the K-clusters. These centroids would then mark the center of the masking patches. Finally, GPE deletes K square patches with the fixed size of $\lfloor \frac{\min(W,H)}{10} \rfloor$ (W and H denoting the width and height of a frame, respectively).

Cropping: We perform horizontal cropping over the original video. The cut-off location is determined according to the average of all keypoints' horizontal positions (y_{avg}). However, to ensure the cut-off not making the aspect ratio too small, we limit its value by $\min(y_{\text{avg}}, 2H/3)$. Finally, the cropped images are resized to the original resolution of the dataset.

Gaussian & Impulse Noise: Following [3] and adhering to their settings, we perturb the video frames using Gaussian and impulse noise. We randomly inject zero-mean Gaussian noise with $\sigma = 0.38$ over all pixels at each frame in the original video. For the impulse noise, we randomly add impulse noise to 27% of the pixels at each frame.

Motion Blur: We utilize the motion blur kernel implemented by [3]. This distortion emulates the blurring effect caused by a fast moving subject or camera, by applying a shifted Gaussian kernel specified by the angle of the shifting operation (θ) and its standard deviation along horizontal and vertical axes (δ). In this paper, we chose $\theta \sim \text{Uniform}(-\pi/4, \pi/4)$ and $\delta = [20, 15]^T$.

In Figure 1, we provide frame samples of the H36M-C and HumanEva-I-C datasets, to better visualize the effect of each video corruption operator.

2.2. Baseline Models

All baselines are optimized using Adam optimizer [6] with a learning rate 0.001, step decay 0.95, and batch-size equal to 1024, unless otherwise stated. We train Pose3D-RIE for 240 and the rest of the baselines for 80 epochs. All baselines are implemented in PyTorch and trained on a single GPU (except for PoseFormer [16] and Pose3D-RIE [12]).

VideoPose3D (VP3D) [10]: We trained VP3D models with various receptive fields of 1, 3, 9 and 27. For the single frame model (i.e. receptive field size of 1), we use $B = 3$ convolution blocks, each with kernel size $K = 1$. We keep

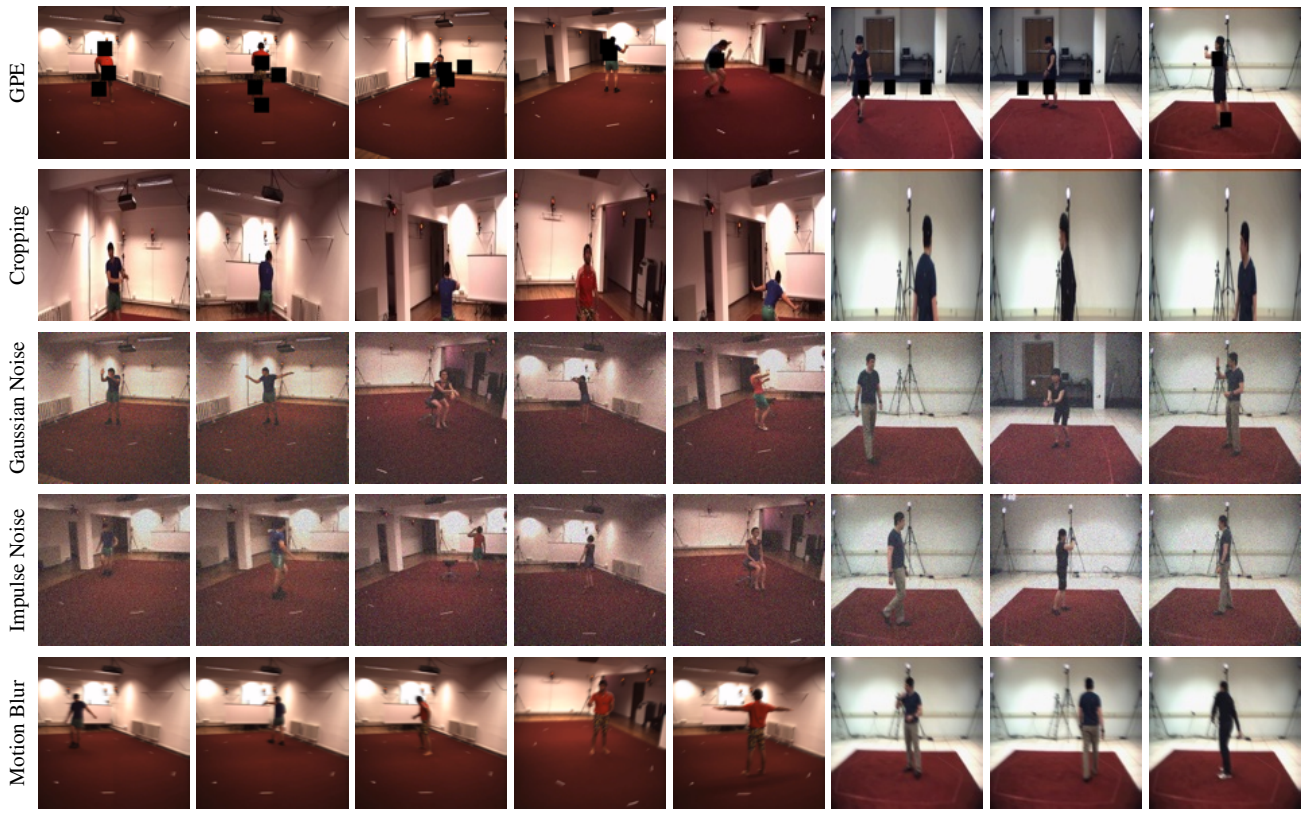

Figure 1. Examples of H36M-C (first five columns from the left) and HumanEva-I-C (last three columns from the left) dataset. Each row corresponds to samples obtained under different video corruption operators.

the number of intermediate channels as $C = 1024$. For VP3D models with receptive field ≥ 1, the kernel size of all convolution blocks is $K = 3$. For models with a receptive field of size 3^m, $m \geq 1$, the number of convolution blocks is m. We also set the dropout rate as 25%.

SRNet [15]: We used the proposed split-and-recombine model. For better comparability, we select the same architecture as VP3D, with $B = 3$ convolution blocks and a receptive field of 27 frames.

PoseFormer [16]: We adopted a PoseFormer model with 27 frames receptive field. The spatial and temporal transformers both consist of 4 blocks and the stochastic depth rate [4] is set to 10% during training. Additionally, the token embedding size is set to 32 while the number of heads is 8. We used $4\times$A100 GPUs for training in parallel with a batch size of 256 on each single GPU.

Attention3DHP [9]: We employed an Attention3DHP model with 243-frame receptive field and $C = 1024$ intermediate channels. For this baseline, we chose a batch size of 2048.

Pose3D-RIE [12]: Following [12], we use a 3-stage optimization pipeline, run for a total of 240 epochs to fully exploit the positional and temporal information in human keypoint groups. The encoder and the feature fusion module (FFM) is trained in the first stage. In the second stage, only the FFM and decoder are optimized. Finally, the parameters of the entire framework is fine-tuned with a small learning rate in the third stage. For a fair comparison with other single stage baselines, we used a receptive field size of 27 instead of 243 frames, as originally used by [12]. Meanwhile, we keep the latent feature dimension of 256 and use a smaller learning rate of 0.0005 similar to [12]. We used $4\times$A100 GPUs for training in parallel, with an effective batch size of 2048.

2.3. 2D Keypoint Detectors

We use HRNet [13] and LiteHRnet [14] for the 2D keypoint detection. Both 2D pose estimators are top-down solutions requiring a human detector. In our experiments with H36M-C, we found that a simple Faster-RCNN [11] with a ResNet-50 [2] and feature pyramid network backbone trained [7] on COCO [8] performs well on the human detection task. We used HRNet-W48 and LiteHRNet-18 both trained on COCO with input image size of 256×192. The specific configurations of the 2D keypoint detectors can be found in the MMPose framework[1].

Table 1. Comparison between various baselines trained with TAGN. The mean and standard deviation across 5 random runs are reported.

Model	Gaussian Noise	Impulse Noise	Temporal-patch Erasing	Guided-patch Erasing	Cropping	Motion Blur	Average
VP3D[10]/H36M+*TAGN* ($\sigma = 0.05; p = k = 20\%$)	91.73 ± 0.30	94.12 ± 0.32	96.08 ± 0.38	112.72 ± 0.56	117.23 ± 1.09	76.54 ± 0.24	98.07 ± 0.48
VP3D[10]/H36M+*TAGN* ($\sigma = 0.1; p = k = 30\%$)	88.44 ± 0.35	90.69 ± 0.35	93.56 ± 0.13	108.95 ± 0.22	114.30 ± 0.68	75.38 ± 0.09	95.22 ± 0.30
VP3D[10]/H36M+*TAGN* ($\sigma = 0.3; p = k = 50\%$)	86.30 ± 0.27	88.46 ± 0.27	92.57 ± 0.33	106.74 ± 0.65	107.71 ± 0.54	75.89 ± 0.16	92.94 ± 0.37
PoseFormer[16]/H36M+*TAGN* ($\sigma = 0.3; p = k = 50\%$)	102.24 ± 4.03	104.78 ± 4.13	113.80 ± 3.67	129.33 ± 3.67	148.39 ± 7.11	88.70 ± 4.31	114.54 ± 4.49
SRNet[15]/H36M+*TAGN* ($\sigma = 0.3; p = k = 50\%$)	91.68 ± 1.32	94.04 ± 1.49	96.16 ± 0.82	111.94 ± 0.90	117.28 ± 3.18	78.70 ± 0.59	98.30 ± 1.38
Attention3DHP[9]/H36M+*TAGN* ($\sigma = 0.3; p = k = 50\%$)	92.05 ± 0.77	93.94 ± 0.90	100.48 ± 1.12	115.08 ± 1.13	112.80 ± 1.52	84.35 ± 0.81	99.78 ± 1.04
Pose3D-RIE[12]/H36M+*TAGN* ($\sigma = 0.3; p = k = 50\%$)	92.14 ± 1.15	101.34 ± 0.23	105.45 ± 1.13	115.31 ± 0.34	104.42 ± 0.12	92.40 ± 1.62	101.84 ± 0.68

Table 2. Effect of TAGN on $\text{MPJPE}_{\leq 0.1}$ of VP3D [10] models trained/tested on 2D keypoints detected by HRNet[13] and Lite-HRNet [14]. The mean and standard deviations across 5 random runs are reported.

Training Keypoints	Testing Keypoints	Model	Gaussian Noise	Impulse Noise	Temporal-patch Erasing	Guided-patch Erasing	Cropping	Motion Blur	Average
HRNet	HRNet	VP3D[10]	94.27	96.64	99.32	116.54	118.08	78.14	100.50
HRNet	HRNet	VP3D[10]+TAGN	86.30 ± 0.27	88.46 ± 0.27	92.57 ± 0.33	106.74 ± 0.65	107.71 ± 0.54	75.89 ± 0.16	92.94 ± 0.37
HRNet	Lite-HRNet	VP3D[10]	123.69	127.27	118.76	132.68	116.56	91.76	118.45
HRNet	Lite-HRNet	VP3D[10]+TAGN	121.78 ± 0.25	125.04 ± 0.28	110.21 ± 0.36	121.85 ± 0.73	107.11 ± 0.76	87.59 ± 0.16	112.26 ± 0.42
Lite-HRNet	HRNet	VP3D[10]	94.32	96.76	101.47	118.77	121.07	77.56	101.66
Lite-HRNet	HRNet	VP3D[10]+TAGN	91.73 ± 0.30	94.12 ± 0.32	96.08 ± 0.38	112.72 ± 0.56	117.23 ± 1.09	76.54 ± 0.24	98.07 ± 0.48
Lite-HRNet	Lite-HRNet	VP3D[10]	120.30	123.76	116.66	130.20	116.80	87.94	115.94
Lite-HRNet	Lite-HRNet	VP3D[10]+TAGN	85.82 ± 0.33	88.05 ± 0.33	95.73 ± 0.24	110.03 ± 0.30	109.13 ± 1.05	76.19 ± 0.17	94.16 ± 0.40

Table 3. Effect of CA-Conv block on $\text{MPJPE}_{\leq 0.1}$ of VP3D[10] models trained/tested on 2D keypoints detected by HRNet [13] and Lite-HRNet [14].

Training Keypoints	Testing Keypoints	Model	Gaussian Noise	Impulse Noise	Temporal-patch Erasing	Guided-patch Erasing	Cropping	Motion Blur	Average
HRNet	HRNet	VP3D[10]	73.11	74.03	81.65	90.56	79.76	68.40	77.92
HRNet	HRNet	VP3D[10]+CA-Conv	72.31	73.28	79.45	87.72	76.84	67.16	76.13
HRNet	Lite-HRNet	VP3D[10]	105.71	109.93	101.01	107.46	83.36	80.71	98.03
HRNet	Lite-HRNet	VP3D[10]+CA-Conv	107.72	111.27	101.56	107.20	83.92	84.17	99.31
Lite-HRNet	HRNet	VP3D[10]	78.21	80.34	87.56	97.35	89.27	71.77	84.08
Lite-HRNet	HRNet	VP3D[10]+CA-Conv	75.24	77.49	84.49	93.92	88.86	69.52	81.59
Lite-HRNet	Lite-HRNet	VP3D[10]	91.60	91.89	97.96	103.43	86.79	79.12	91.80
Lite-HRNet	Lite-HRNet	VP3D[10]+CA-Conv	88.24	88.31	95.35	100.66	84.97	76.51	89.01

Table 4. Effect of receptive field on $\text{MPJPE}_{\leq 0.1}$ of VP3D models trained with and without TAGN. The mean and standard deviations across 5 random runs are reported.

Receptive Field	Model	Gaussian Noise	Impulse Noise	Temporal-patch Erasing	Guided-patch Erasing	Cropping	Motion Blur	Average
1	VP3D[10]	97.80	100.06	102.80	119.24	119.00	80.39	103.22
1	VP3D [10] + TAGN	94.98 ± 0.09	96.87 ± 0.10	94.95 ± 0.08	107.69 ± 0.14	102.16 ± 0.62	80.26 ± 0.14	96.15 ± 0.19
9	VP3D[10]	96.61	99.11	101.95	119.03	118.85	80.92	102.74
9	VP3D [10] + TAGN	93.08 ± 0.40	94.95 ± 0.45	96.69 ± 0.21	110.26 ± 0.37	107.86 ± 0.77	81.25 ± 0.23	97.35 ± 0.40
27	VP3D[10]	94.27	96.64	99.32	116.54	118.08	78.14	100.50
27	VP3D [10] + TAGN	86.30 ± 0.27	88.46 ± 0.27	92.57 ± 0.33	106.74 ± 0.65	107.71 ± 0.54	75.89 ± 0.16	92.94 ± 0.37
81	VP3D[10]	93.14	95.73	97.34	114.60	120.78	76.54	99.69
81	VP3D [10] + TAGN	84.61 ± 0.37	86.67 ± 0.41	91.36 ± 0.20	105.75 ± 0.38	106.92 ± 0.51	75.16 ± 0.16	91.74 ± 0.34

3. Experimental Results

In Figure 2, we evaluate the effect of the threshold τ on the number of joints considered when computing the $\text{MPJPE}_{\leq \tau}$ metric. As expected, with a larger threshold, more joints are included in the computation of $\text{MPJPE}_{\leq \tau}$. In our experiments, we selected a default value of $\tau = 0.1$ which captures 87% of the total joints.

To showcase the consistency of the performance of

Table 5. Effect of VP3D [10]'s receptive field on MPJPE$_{\leq 0.1}$ of VP3D models trained with CA-Conv

Receptive Field	Model	Gaussian Noise	Impulse Noise	Temporal-patch Erasing	Guided-patch Erasing	Cropping	Motion Blur	Average
1	VP3D[10]	86.96	87.96	87.75	96.52	81.05	76.61	86.14
1	VP3D [10] + CA-Conv	85.21	86.16	84.16	92.16	77.14	74.85	83.28
9	VP3D[10]	78.11	79.27	85.68	94.57	83.53	71.92	82.18
9	VP3D [10] + CA-Conv	76.68	77.58	82.96	91.61	79.53	69.92	79.71
27	VP3D[10]	73.11	74.03	81.65	90.56	79.76	68.40	77.92
27	VP3D [10] + CA-Conv	72.31	73.28	79.45	87.72	76.84	67.16	76.13
81	VP3D[10]	70.85	71.82	79.55	87.83	77.94	66.78	75.80
81	VP3D [10] + CA-Conv	69.92	70.91	77.02	85.23	76.24	65.42	74.12

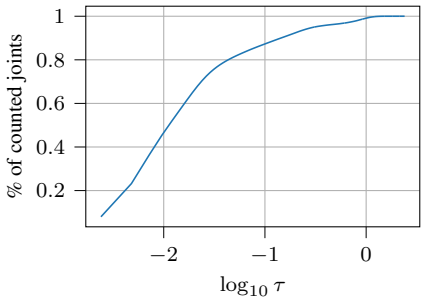

Figure 2. Percentage of joints (per frame) involved in deriving MPJPE$_{\leq\tau}$ as a function of the threshold τ. The results are reported on H36M-C test set.

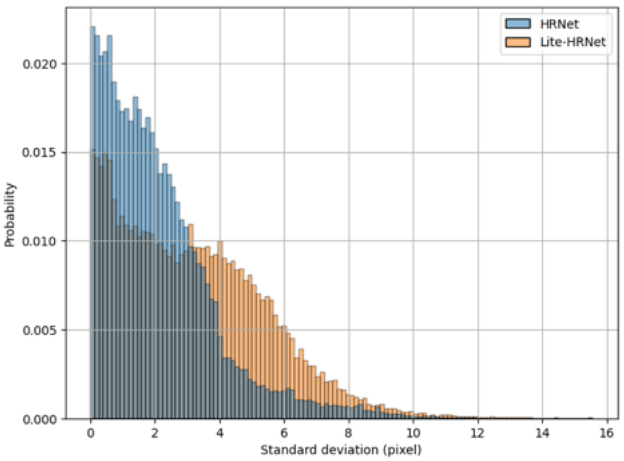

Figure 3. Histogram of the standard deviation of the heatmaps around 2D keypoint predictions of one subject (H36M dataset) from HRNet [13] and Lite-HRNet [14]. The confidence score of each detected 2D keypoint corresponds to the maximum of the associated heatmap.

TAGN, in Table 1, 2, and 4, we report the mean and standard deviation across 5 random runs. Note that, in each run, we have different noise realizations added by TAGN to the 2D input pose. The small standard deviations affirm the stable boost in performance, offered by TAGN.

In Figure 3, we provide the histogram of the standard deviation of output heatmaps around all 2D keypoint predictions of one subject from HRNet [13] and Lite-HRNet [14]. We notice the difference between the two distributions produced by HRNet and Lite-HRNet. In general, HRNet predicts not only more accurate but also more confident keypoints than Lite-HRNet.

Furthermore, in Table 2 and 3, we study VP3D [10] trained with TAGN or CA-Conv, with 2D pose output by HRNet and LiteHRNet. Similarly, Table 4 and 5 summarize the effect of receptive field size on VP3D's [10] performance. In all scenarios, our proposed TAGN and CA-Conv solutions lead to improvements in MPJPE$_{\leq\tau}$ compared to the upper and lower-bound benchmarks.

Lastly, in Figure 4, we visualize additional qualitative comparisons between VP3D[10] and VP3D [10] + TAGN models, both trained on the original Human3.6M [5] dataset.

References

[1] MMPose Contributors. Openmmlab pose estimation toolbox and benchmark. https://github.com/open-mmlab/mmpose, 2020. 2

[2] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 2

[3] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. 1

[4] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q. Weinberger. Deep networks with stochastic depth. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 646–661, Cham, 2016. Springer International Publishing. 2

[5] 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 Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014. 4

[6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Represen-*

Figure 4. Qualitative comparison of VP3D+TAGN ($\sigma = 0.3, p = k = 50\%$) versus VP3D, on H36M-C dataset (with guided-patch erasing corruption). The bone color leading to a joint turns red when its MPJPE increases.

tations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015. 1

[7] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017. 2

[8] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In

David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. 2

[9] Ruixu Liu, Ju Shen, He Wang, Chen Chen, Sen-ching Cheung, and Vijayan Asari. Attention mechanism exploits temporal contexts: Real-time 3D human pose reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5064–5073, 2020. 2, 3

[10] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3D human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7753–7762, 2019. 1, 3, 4

[11] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015. 2

[12] Wenkang Shan, Haopeng Lu, Shanshe Wang, Xinfeng Zhang, and Wen Gao. Improving robustness and accuracy via relative information encoding in 3D human pose estimation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3446–3454, 2021. 1, 2, 3

[13] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019. 2, 3, 4

[14] Changqian Yu, Bin Xiao, Changxin Gao, Lu Yuan, Lei Zhang, Nong Sang, and Jingdong Wang. Lite-HRnet: A lightweight high-resolution network. In *CVPR*, 2021. 2, 3, 4

[15] Ailing Zeng, Xiao Sun, Fuyang Huang, Minhao Liu, Qiang Xu, and Stephen Ching-Feng Lin. SRNet: Improving generalization in 3D human pose estimation with a split-and-recombine approach. In *ECCV*, 2020. 2, 3

[16] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3D human pose estimation with spatial and temporal transformers. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 1, 2, 3