Supplemental Document of Paper “NIKI: Neural Inverse Kinematics with Invertible Neural Networks for 3D Human Pose and Shape Estimation”

A. Architecture of INN

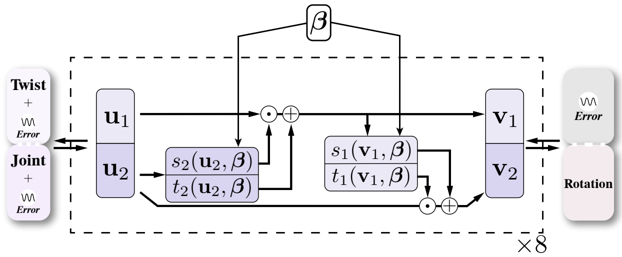

β
Twist
+
Error
Joint
+
Error
$\mathbf{u}_1$
$\mathbf{u}_2$
$s_2(\mathbf{u}_2, \boldsymbol{\beta})$
$t_2(\mathbf{u}_2, \boldsymbol{\beta})$
$s_1(\mathbf{v}_1, \boldsymbol{\beta})$
$t_1(\mathbf{v}_1, \boldsymbol{\beta})$
$\mathbf{v}_1$
$\mathbf{v}_2$
Error
Rotation
×8

Figure 1. **Detailed architecture of the one-stage mapping model.**

A.1. One-Stage Mapping

The detailed architecture of the one-stage mapping model is illustrated in Fig. 1. We follow the architecture of RealNVP [2]. The model consists of multiple basic blocks to increase capacity. The input vector $\mathbf{u}$ of the block is split into two parts, $\mathbf{u}_1$ and $\mathbf{u}_2$, which are subsequently transformed with coefficients $\exp(s_i)$ and t_i ($i \in \{1, 2\}$) by the two affine coupling layers:

$$\mathbf{v}_1 = \mathbf{u}_1 \odot \exp(s_2(\mathbf{u}_2, \boldsymbol{\beta})) + t_2(\mathbf{u}_2, \boldsymbol{\beta}), \tag{1}$$

$$\mathbf{v}_2 = \mathbf{u}_2 \odot \exp(s_1(\mathbf{v}_1, \boldsymbol{\beta})) + t_1(\mathbf{v}_1, \boldsymbol{\beta}), \tag{2}$$

where $\mathbf{v} = [\mathbf{v}_1, \mathbf{v}_2]$ is the output vector of the block and $\odot$ denotes element-wise multiplication. The coefficients of the affine transformation can be learned by arbitrarily complex functions, which do not need to be invertible. The invertibility is guaranteed by the affine transformation in Eq. 1 and 2. The scale network s_i is a 3-layer MLP with the hidden dimension of 512, and the translation network t_i has the same architecture followed by a $\tanh$ activation function.

A.2. Twist-and-Swing Mapping

The detailed architecture of the twist-and-swing mapping model is illustrated in Fig. 2. The two-step mapping is implemented by two separate invertible networks. The first network has the same architecture as the one-stage mapping model, while its input is only the joint positions, and the output is the swing rotations. The second network removes the shape condition and directly transforms the twist and swing rotations to complete rotations.

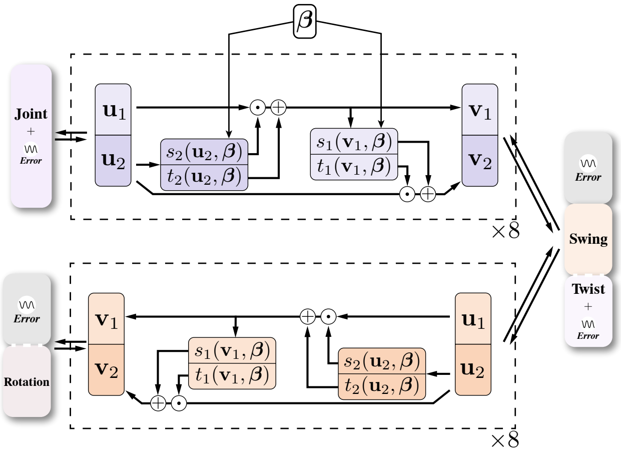

β
Joint
+
Error
$\mathbf{u}_1$
$\mathbf{u}_2$
$s_2(\mathbf{u}_2, \boldsymbol{\beta})$
$t_2(\mathbf{u}_2, \boldsymbol{\beta})$
$s_1(\mathbf{v}_1, \boldsymbol{\beta})$
$t_1(\mathbf{v}_1, \boldsymbol{\beta})$
$\mathbf{v}_1$
$\mathbf{v}_2$
×8
Error
Swing
Twist
+
Error
Error
Rotation
$\mathbf{v}_1$
$\mathbf{v}_2$
$s_1(\mathbf{v}_1, \boldsymbol{\beta})$
$t_1(\mathbf{v}_1, \boldsymbol{\beta})$
$s_2(\mathbf{u}_2, \boldsymbol{\beta})$
$t_2(\mathbf{u}_2, \boldsymbol{\beta})$
$\mathbf{u}_1$
$\mathbf{u}_2$
×8

Figure 2. **Detailed architecture of the twist-and-swing mapping model.**

B. Implementation Details

In our experiments, we use the weights pretrained on `COCO` [9] 2D pose estimation task for the initialization of the CNN backbone to accelerate convergence. The scalar coefficients in the loss function are $\lambda_{\mathit{inv}} = 1$, $\lambda_{\mathit{fwd}} = 1$, $\lambda_{\mathit{ind}} = 1$, $\lambda^{i}_{\mathit{bnd}} = 0.1$, $\lambda^{f}_{\mathit{bnd}} = 1$. We first train the CNN backbone following HybrIK [8] to obtain initial joint positions and twist rotations. Then we solely train NIKI and freeze the parameters of the CNN backbone. During training, we follow EFT [3], SPIN [6], and PARE [5], which use fixed data sampling ratios for each batch. We incorporate 50% `Human3.6M` and 50% `3DPW` when conducting experiments on the `3DPW` and `3DPW-XOCC` datasets. For experiments on the `3DPW-OCC` and `3DOH` datasets, we incorporate 35% `COCO`, 35% `Human3.6M`, and 30% `3DOH`.

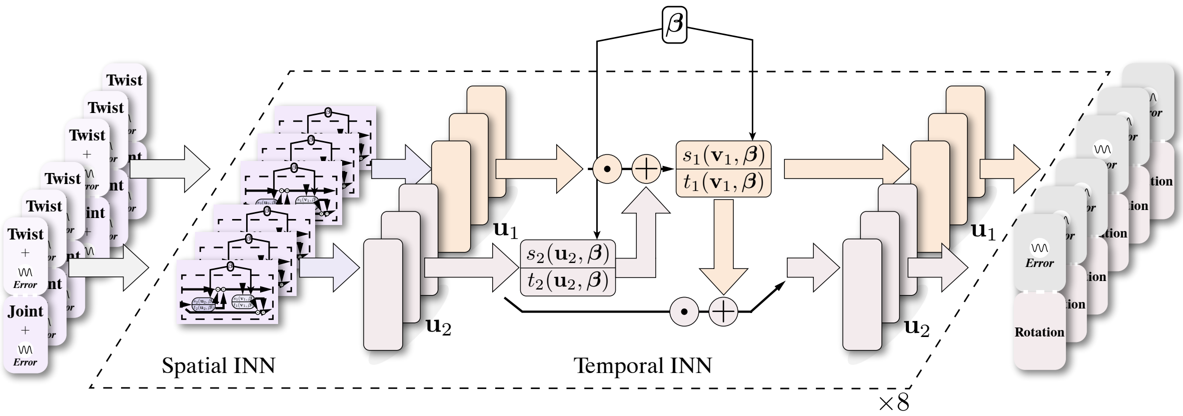

β
Twist
Twist
Twist
Twist
Twist
Twist
+
Error
Joint
+
Error
$\mathbf{u}_1$
$\mathbf{u}_2$
$s_1(\mathbf{v}_1, \boldsymbol{\beta})$
$t_1(\mathbf{v}_1, \boldsymbol{\beta})$
$s_2(\mathbf{u}_2, \boldsymbol{\beta})$
$t_2(\mathbf{u}_2, \boldsymbol{\beta})$
$\mathbf{u}_1$
$\mathbf{u}_2$
Error
Rotation
Spatial INN
Temporal INN
×8

Figure 3. **Detailed architecture of the temporal INN.**

Method	3DPW MPJPE ↓	PA-MPJPE ↓	PVE ↓	ACCEL ↓
VIBE [4]	82.9	51.9	99.1	23.4
MEVA [10]	86.9	54.7	-	11.6
TCMR [1]	86.5	52.7	102.9	7.1
MAED [15]	79.1	45.7	92.6	17.6
D&D [7]	73.7	42.7	88.6	**7.0**
NIKI (Frame-based)	71.3	40.6	86.6	15.1
NIKI (Temporal)	**71.2**	**40.5**	**86.3**	12.3

Table 1. **Quantitative comparisons with state-of-the-art temporal methods on the `3DPW` dataset.** Symbol "-" means results are not available.

Method	3DPW-XOCC MPJPE ↓	PA-MPJPE ↓	PVE ↓	ACCEL ↓
HybrIK [8]	148.3	98.7	164.5	108.6
PARE* [5]	114.2	67.7	133.0	90.7
PARE* [5] + VIBE [4]	97.3	60.2	114.9	18.3
NIKI (Frame-based)	110.7	60.5	128.6	74.4
NIKI (Temporal)	**88.9**	**52.1**	**98.0**	**17.3**

Table 2. **Quantitative comparisons with state-of-the-art temporal methods on the `3DPW-XOCC` dataset.** Symbol $*$ means finetuning on the `3DPW-XOCC` train set.

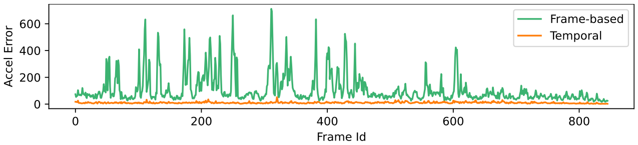


Figure 4. **Acceleration error curve.**

C. Temporal Extension of NIKI

C.1. Architecture

We extend the invertible network for temporal input. We design a spatial-temporal INN model to incorporate temporal information to solve the IK problem. For simplicity, we use the basic block in the one-stage mapping and twist-and-swing mapping models as the spatial INN. Self-attention modules are introduced to serve as the temporal INN and conduct temporal affine transformations. The temporal input vectors $\{\mathbf{u}^t\}_1^T$ are split into two subsets, $\{\mathbf{u}^t\}_1^{\lfloor T/2 \rfloor}$ and $\{\mathbf{u}^t\}_{\lfloor T/2 \rfloor+1}^T$, which are subsequently transformed with coefficients $\exp(s_i)$ and t_i ($i \in \{1, 2\}$) by the two affine coupling layers like Eq. 1 and 2. We adopt self-attention layers [13] as the temporal scale and translation layers. The detailed network architecture of the temporal INN is illustrated in Fig. 3.

C.2. Experiments of the Temporal Extension

We evaluate the temporal extension on both standard and occlusion-specific benchmarks. Tab. 1 compares temporal NIKI with previous state-of-the-art temporal HPS methods on the standard `3DPW` [14] dataset. Notice that we do not design complex network architecture or use dynamics information. Our temporal extension simply applies the affine coupling layers to the time domain. It shows that our simple extension obtains better accuracy than state-of-the-art dynamics-based approaches.

Tab. 2 presents the performance on the occlusion-specific benchmark. We compare the temporal extension with a strong baseline. The baseline combines PARE [5] with the state-of-the-art temporal approach, VIBE [4]. We first use the backbone of PARE [5] to extract attention-guided features. Then we apply VIBE [4] to incorporate temporal information to predict smooth and robust human motions. Temporal NIKI outperforms the baseline in challenging occlusions and truncations.

Fig. 4 present the acceleration error curves of the single-frame and temporal models in the `3DPW-XOCC` dataset. We

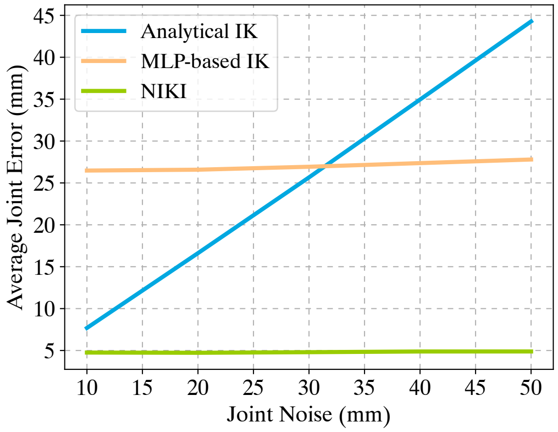


Figure 5. **Noise sensitivity analysis** of analytical IK, MLP-based IK and NIKI.

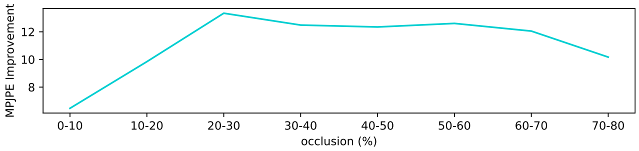


Figure 6. **Improvement over different occlusion levels.**

can observe that the temporal model can improve motion smoothness.

D. Noise Analysis

We assess the robustness of three different IK algorithms: analytical IK, MLP-based IK, and NIKI. We evaluate their performance on the AMASS dataset [11] with noisy joint positions. As shown in Fig. 5, MLP-based IK is more robust than the analytical IK when the noise is larger than 30 mm. However, MLP-based IK fails to obtain pixel-aligned performance when the noise is small. NIKI shows superior performance at all noise levels.

E. Collision Analysis

To quantitatively show that the output poses from NIKI are more plausible, we compare the collision ratio of mesh triangles [12] between HybrIK and NIKI on the 3DPW-XOCC dataset. NIKI reduces the collision ratio from 2.6% to 1.0% (57.7% relative improvement).

F. Occlusion Analysis

We follow the framework of [5, 17] and replace the classification score with an error measure for body poses. We choose MPJPE as the error measurement. This analysis is not limited to a particular network architecture. We apply it to the state-of-the-art pixel-aligned approach, HybrIK [8], and the direct regression approach, PARE [5]. The visualizations of the error maps are shown in Fig. 7 and 8. Warmer colors denote a higher MPJPE. It shows that NIKI is more robust to body part occlusions.

	3DPW		3DPW-XOCC	
	MPJPE ↓	PA-MPJPE ↓	MPJPE ↓	PA-MPJPE ↓
NIKI	71.3	40.6	**110.7**	**60.5**
+ Heatmap Cond. [16]	**71.1**	**40.4**	110.8	60.6

Table 3. **Integrate heatmap condition.**

Additionally, we follow the official AGORA analyses to compare the performance in different occlusion levels. As shown in Fig. 6, in the low occlusion level (0-10%), NIKI brings 6.5 mm MPJPE improvement. The improvement reaches a peak (13.3 mm) in the medium occlusion level (20-30%). For the high occlusion level (70-80%), the improvement falls back to 10.2 mm. We can observe that NIKI is good at handling medium occlusions. There is still a lot of room for improvement in highly occluded scenarios.

G. Heatmap Condition

We follow Wehrbein *et al.* [16] and add heatmap condition in the INN. As shown in Tab. 3, it brings 0.2 mm improvement on the 3DPW dataset. However, it is 0.1 mm worse on the 3DPW-XOCC dataset. We assume this is because heatmap is not reliable under server occlusions.

H. Inference Time and Model Size

We benchmark the inference time of the analytical IK algorithm, HybrIK [8] and NIKI with an RTX 3090 GPU with a batch size of 1. The latency of HybrIK is 26 ms and NIKI is 8 ms, respectively. HybrIK is much slower since it needs to solve the rotations iteratively along the kinematic tree. For the model size, the total parameters of NIKI is 29.01M.

I. Details of 3DPW-XOCC

3DPW-XOCC is a new benchmark for human pose and shape estimation with extremely challenging occlusions and truncations. The dataset is augmented from the original 3DPW dataset by adding temporally-smooth synthetic occlusions and truncations. To ensure temporal smoothness, we choose keyframes at an interval of 8 frames, and the rest frames are generated by linearly interpolating the clipping and occlusion of the keyframes. In the keyframe, the image is randomly clipped to ensure that at least one body part is outside the clipped image with a possibility of over $2/3$. A square area that takes up to 30% of the clipped image is replaced by gaussian noise to serve as occlusion. The evaluation protocol and the split of the dataset are unchanged.

J. Limitations and Future Work

Our work has several limitations. First, NIKI does not include body shape refinement. Human body shape estimation is also challenging in occlusion scenarios. The incorrect body shape would cause incorrect distal joints reconstruction. For example, even the knee and ankle rotations are correct, the wrong leg length will cause a wrong ankle position. Exploiting the bone length information in joint positions can help refine β for better pose and shape estimation. Second, NIKI does not use the scene information to separate the pose error. The initial joint positions could be physiologically plausible but do not match the input scene. Using scene constraints can reduce implausible human-scene interactions and further improve robustness. Third, the training of NIKI relies on the diversity of datasets. To accurately built the bijective mapping, the training data need to be diverse enough. We believe these limitations are exciting avenues for future work to explore.

K. Qualitative Results

Additional qualitative results are shown in Fig. 9 and 10. More results on the Internet videos are provided in `demo.mp4` in the supplementary files.

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*, 2021. 2

[2] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. In *ICLR*, 2017. 1

[3] Hanbyul Joo, Natalia Neverova, and Andrea Vedaldi. Exemplar fine-tuning for 3d human model fitting towards in-the-wild 3d human pose estimation. In *3DV*, 2021. 1

[4] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *CVPR*, 2020. 2

[5] Muhammed Kocabas, Chun-Hao P Huang, Otmar Hilliges, and Michael J Black. Pare: Part attention regressor for 3d human body estimation. In *ICCV*, 2021. 1, 2, 3, 5, 7

[6] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *ICCV*, 2019. 1

[7] Jiefeng Li, Siyuan Bian, Chao Xu, Gang Liu, Gang Yu, and Cewu Lu. D&d: Learning human dynamics from dynamic camera. In *ECCV*, 2022. 2

[8] Jiefeng Li, Chao Xu, Zhicun Chen, Siyuan Bian, Lixin Yang, and Cewu Lu. Hybrik: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation. In *CVPR*, 2021. 1, 2, 3, 6, 8

[9] 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 *ECCV*, 2014. 1

[10] Zhengyi Luo, S Alireza Golestaneh, and Kris M Kitani. 3d human motion estimation via motion compression and refinement. In *ACCV*, 2020. 2

[11] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *ICCV*, 2019. 3

[12] Marko Mihajlovic, Shunsuke Saito, Aayush Bansal, Michael Zollhoefer, and Siyu Tang. Coap: Compositional articulated occupancy of people. In *CVPR*, 2022. 3

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017. 2

[14] Timo von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *ECCV*, 2018. 2

[15] Ziniu Wan, Zhengjia Li, Maoqing Tian, Jianbo Liu, Shuai Yi, and Hongsheng Li. Encoder-decoder with multi-level attention for 3d human shape and pose estimation. In *ICCV*, 2021. 2

[16] Tom Wehrbein, Marco Rudolph, Bodo Rosenhahn, and Bastian Wandt. Probabilistic monocular 3d human pose estimation with normalizing flows. In *ICCV*, 2021. 3

[17] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 3

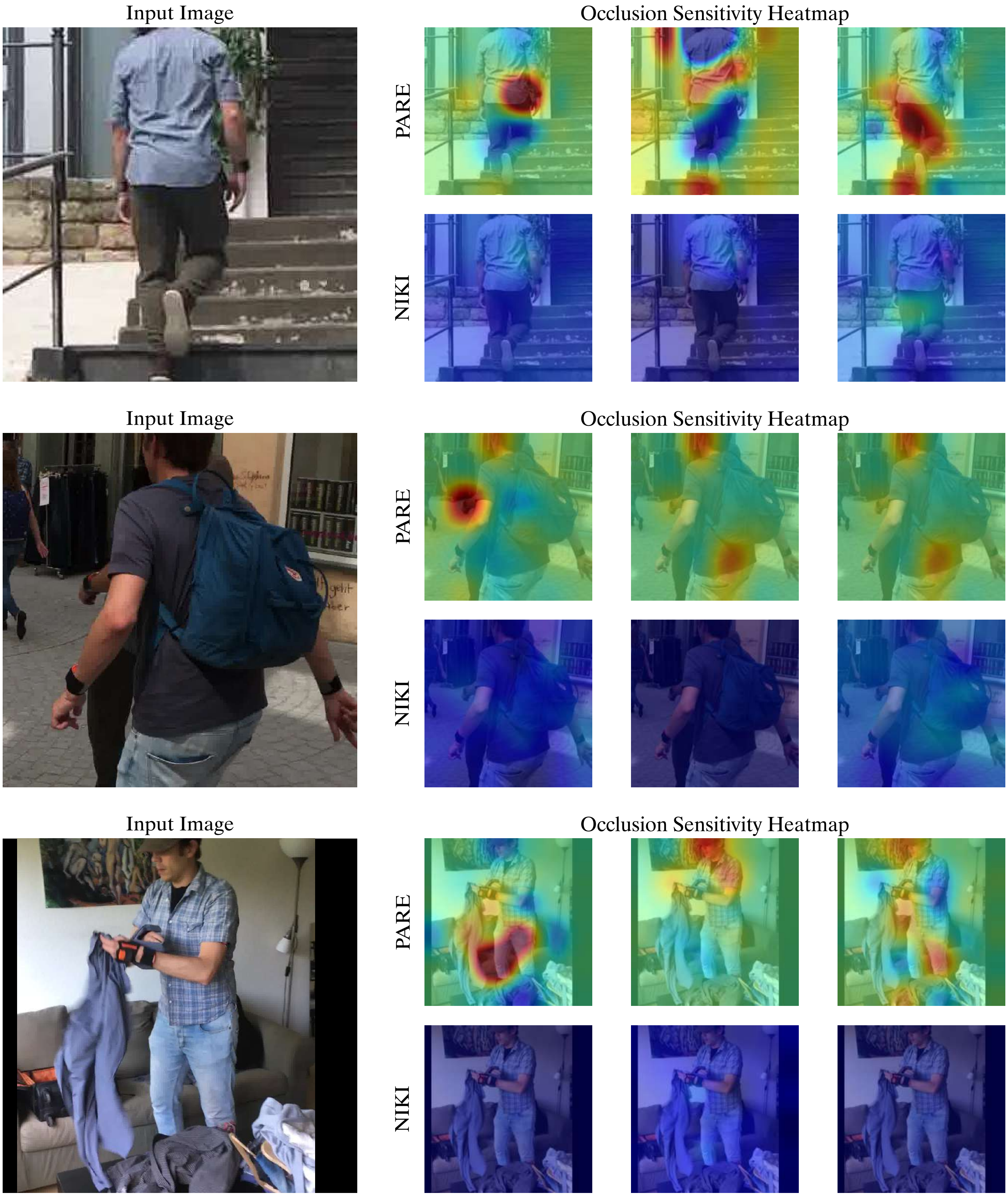

Input Image
Occlusion Sensitivity Heatmap
PARE
NIKI
Input Image
Occlusion Sensitivity Heatmap
PARE
NIKI
Input Image
Occlusion Sensitivity Heatmap
PARE
NIKI

Figure 7. **Occlusion Sensitivity Maps of PARE [5] and NIKI.**

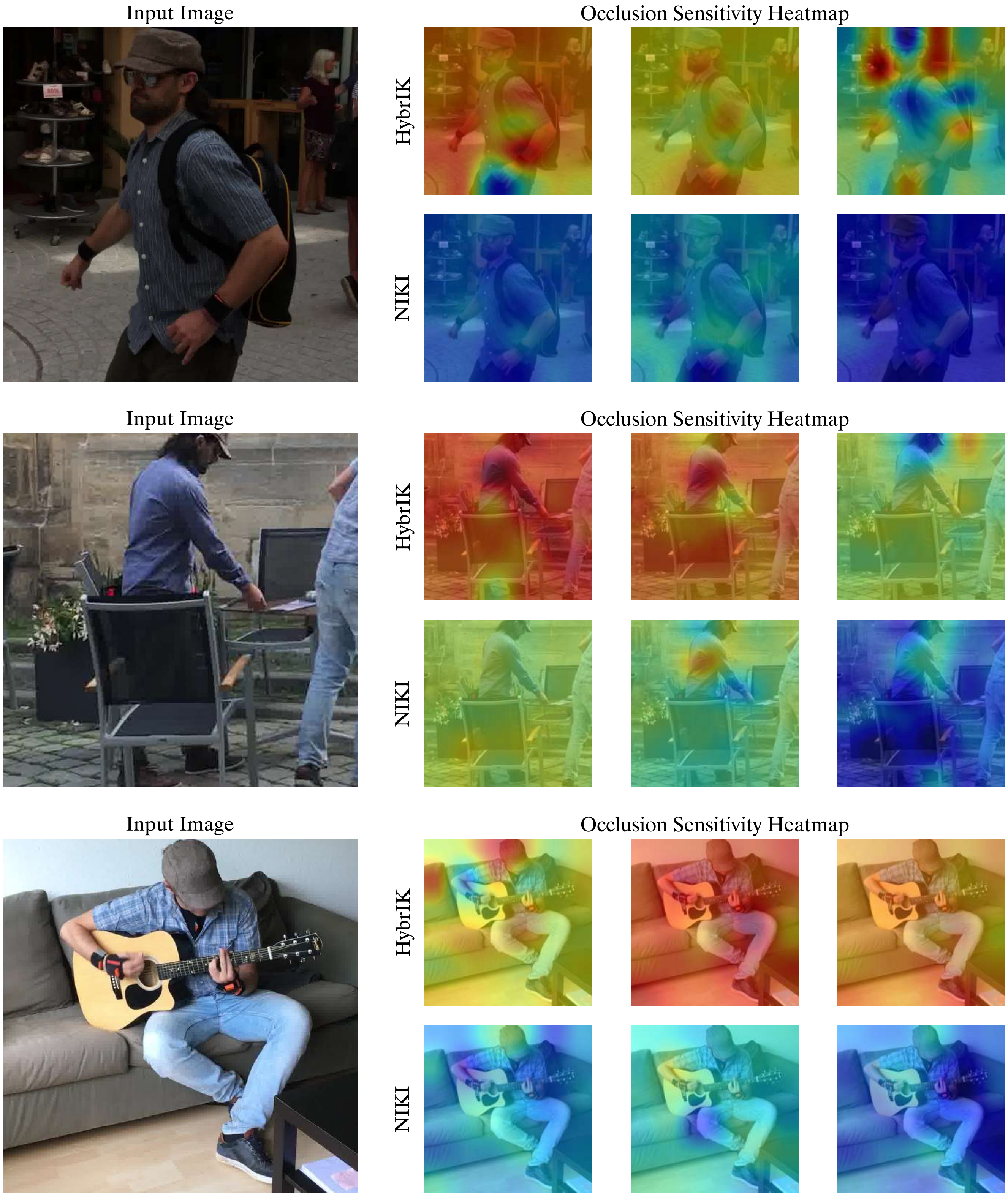
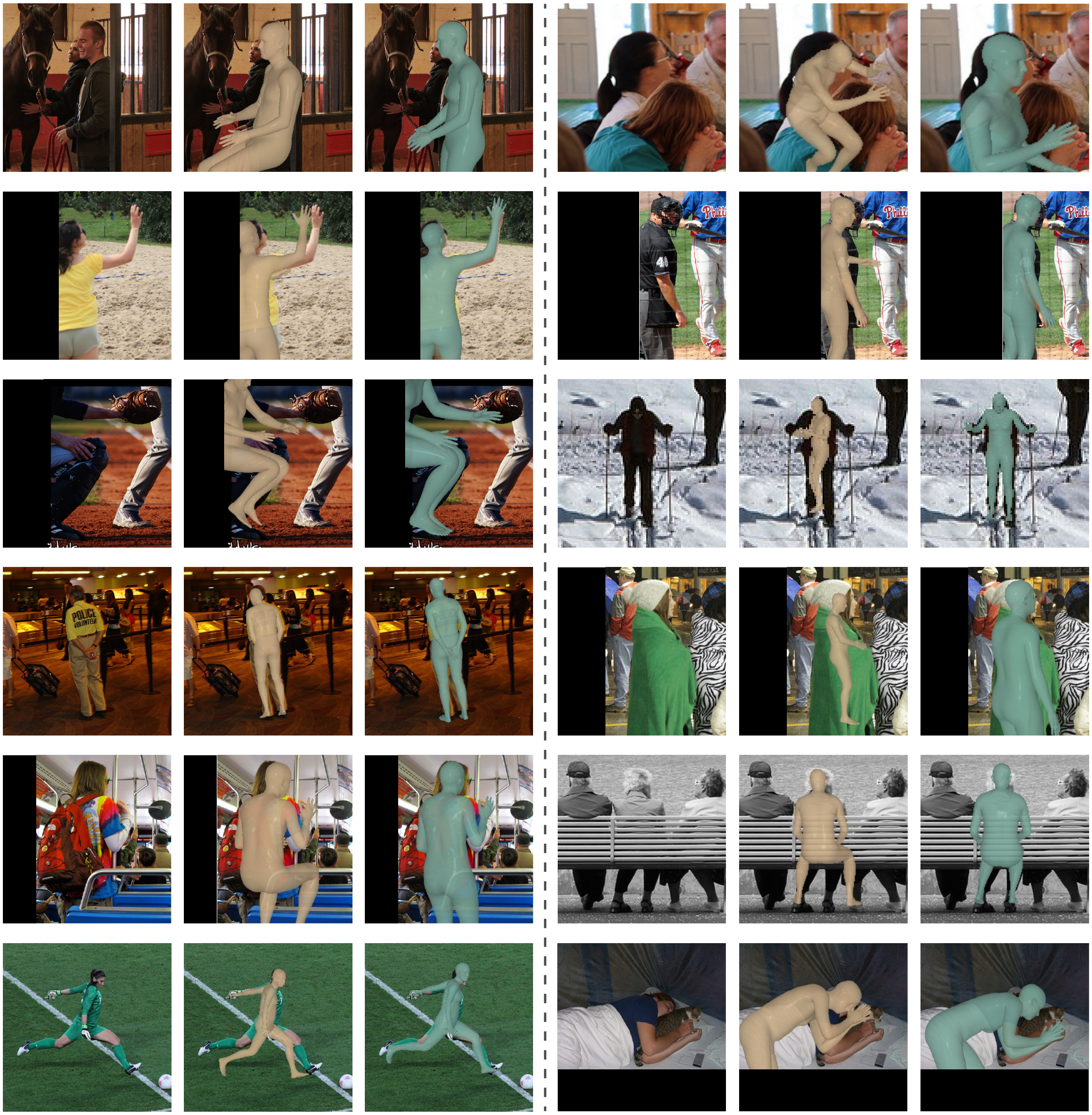
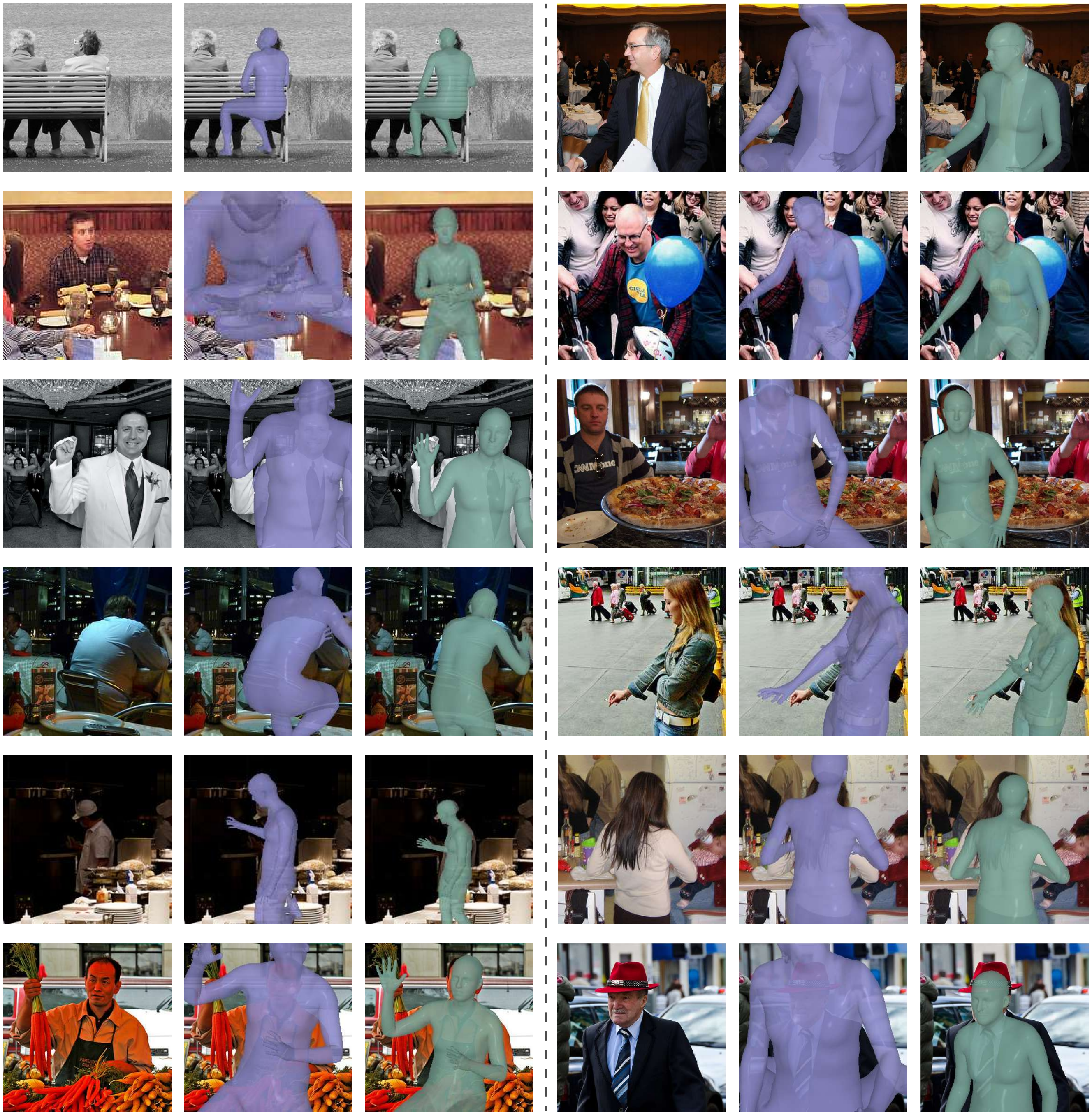

Input Image
Occlusion Sensitivity Heatmap
HybrIK
NIKI
Input Image
Occlusion Sensitivity Heatmap
HybrIK
NIKI
Input Image
Occlusion Sensitivity Heatmap
HybrIK
NIKI

Figure 8. **Occlusion Sensitivity Maps of HybrIK [8] and NIKI.**



(a) PARE (b) NIKI (a) PARE (b) NIKI

Figure 9. **Qualitative comparison with PARE [5].**



(a) HybrIK (b) NIKI (a) HybrIK (b) NIKI

Figure 10. **Qualitative comparison with HybrIK [8].**