End-to-End Detection and Pose Estimation of Two Interacting Hands – Supplemental Document

Dong Uk Kim Kwang In Kim Seungryul Baek
UNIST

In the main paper, we used the joint visibility only at interacting hands since constructing the ground-truth visibility can be challenging for single-hands. Section 1 shows that when available, the joint visibility can lead to a performance gain for single-hands. In all experiments in the main paper, we fixed the bounding box intersection over union (IOU) threshold τ at 0.3 (for training) and 0.5 (for testing). Here, we present the effect of varying τ values (Sec. 2).

In the main paper, we demonstrated that enforcing structural consistency of the estimated interacting hand poses via the joint pose discriminator meaningfully improves the performance. Here, we further support this finding via a statistical significance test (Sec. 3).

The rest of this document presents the details of generating ground-truths for the visibility estimator (Sec. 4) and the network architectures (Sec. 5), and additional hand pose estimation examples (Sec. 6).

1. Additional experiments on *Ego3D*

In the main paper, we used the joint visibility maps only for interacting hands as building the visibility ground-truth is challenging for single-hands: See Sec. 4 for details of ground-truth construction. As *Ego3D* is synthetic, such joint visibility is available for both single-hand and interacting-hand cases. Exploiting this information improves the error rates from 11.63mm to 11.44mm for single hand cases: Table 1 shows the complete results.

2. Effect of varying τ values at training

Our system classifies each input instance into two categories 1) *interacting* and 2) *non-interacting* hands, based on the IOU of the corresponding hand bounding boxes. For interacting hands, the pose estimators are jointly trained and tested, while for non-interacting hands, single-hand pose estimators are individually applied similarly to existing hand pose estimation approaches. Determining the IOU threshold τ value is crucial: In training, τ controls the size of the training set for the *joint* pose estimator: Large τ values lead to small training sets focusing on challenging *closely interacting* cases while small τ values offer large training sets, but they might include *loosely interacting* (easy) cases. Figure 1 shows the effect of varying τ values (on the 'interacting hands' cases of *InterHand2.6M* dataset). The best trade-off was achieved at 0.3 which is the same as the τ value that we used in the main paper (optimized via cross-validation on training sets).

For testing, for high τ values, our joint pose estimator focuses on limited cases of very closely interacting hands, leaving most other cases to the single-hand pose estimator. The overall performance degraded, as in this case mildly interacting cases did not benefit from joint estimation. On the other hand, when τ is too small, our joint estimator is applied to cases which differ significantly from what it has

Table 1: Performances (MPJPE in mm) of alternative design choices of our algorithm on Ego3D. 'Ours ($-L^{\mathrm{D}}$)' removes the contribution of the GAN discriminator L^{D}, 'Ours ($-L^{\mathrm{V}}, L^{\mathrm{D}}$)' further removes the joint visibility estimation and visibility-guided heatmap enhancement networks, 'Ours ($-$Interaction, L^{D})' completely removes the joint training of instances that belong to *interacting hands*, and in 'Ours ($-$end-to-end detection and estimation)' the hand detector is trained and is frozen before the subsequent pose estimation step.

Method	MPJPE
Entire dataset	
Ours ($-$Interaction, L^{D})	11.45 (9.75)
Ours ($-L^{\mathrm{V}}, L^{\mathrm{D}}$)	11.45 (9.75)
Ours ($-L^{\mathrm{D}}$)	11.44 (9.75)
Ours (Separate detection)	12.30 (10.48)
Ours	11.44 (9.74)
Only 'interacting hands' cases	
Ours ($-$Interacting class, L^{D})	18.05 (15.86)
Ours ($-L^{\mathrm{V}}, L^{\mathrm{D}}$)	17.85 (15.45)
Ours ($-L^{\mathrm{D}}$)	17.46 (15.03)
Ours (Separate detection)	18.98 (16.46)
Ours	17.28 (14.82)

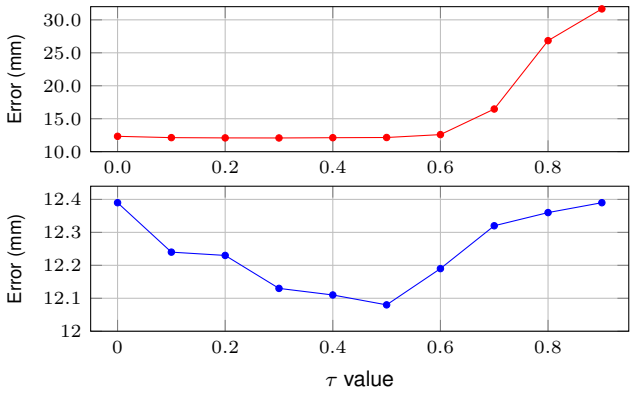


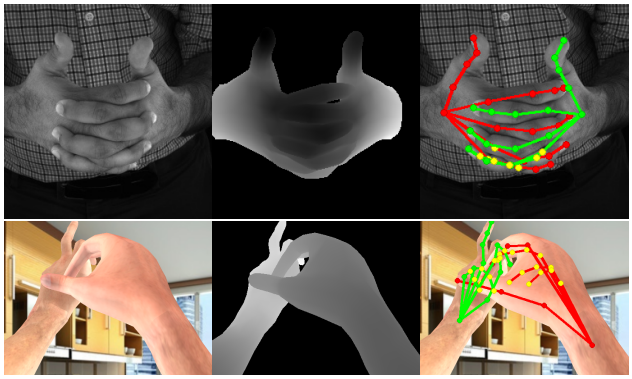
Figure 1: Error rates (MPJPE; in mm) of our system on *InterHand2.6M* under varying τ values at training (red) and testing (blue).

been trained for, again degrading the performance. The best tradeoff was achieved at $\tau = 0.5$ which we use for other datasets as well.

Figure 4 shows example hand classifications with respect to varying τ values.

3. Contribution of the hand pose discriminator

While training our hand pose estimator jointly on interacting hands helps exploit their underlying statistical dependence, it sometimes generates physically implausible configurations (Fig. 3 in the main paper). We account for this by enhancing structural consistency of the estimated *joint hand pose* using a GAN-type pose discriminator. This network sees the estimated skeletons of interacting hands and provides feedback to our 2D heatmap estimator and depth value estimator. Section 4 in the main paper demonstrates that this additional supervision helps improve the performance of our system quantitatively and qualitatively (Table 2 and Fig. 3, respectively, main paper). Here, we further support these findings via statistical significance tests: The *InterHand2.6M* dataset provides an experimental split of 528,000 training and 122,000 testing frames. Among these frames, 284,728 training and 66,734 test frames are provided with the ground-truth skeleton annotations that we used in the experiments of the main paper. We performed additional experiments with 10 different training and testing splits of the same sizes (284,728 and 66,734 frames, respectively). Table 2 shows the results: Ours ($-L^{\mathrm{D}}$) represents a variation of our system constructed by removing the discriminator loss L^{D} in the final training loss of our system (Eq. 5 of the main paper):



(a) original image (b) depth image (c) visibility

Figure 2: Visibility (pseudo) ground-truth examples: (a) input images; (b) depth maps rendered from the hand meshes fitting to the images in (a); (c) generated ground-truths: left and right hands are marked in green and red, respectively while all occluded joints are highlighted in yellow.

$$
\begin{aligned}
L = {} & L^{\mathrm{HPN}}(f^{\mathrm{HPN}}, f^{\mathrm{Feat}}) + L^{\mathrm{Hand}}(f^{\mathrm{CB}}, f^{\mathrm{Feat}}) \\
& + L^{\mathrm{2D}}(f^{\mathrm{H2D}}, f^{\mathrm{TVHE}}) + \lambda_1 L^{\mathrm{3D}}(f^{\mathrm{Z3D}}) \\
& + \lambda_2 L^{\mathrm{D}}(f^{\mathrm{TH2D}}, f^{\mathrm{TZ3D}}, f^{\mathrm{TVHE}}) + \lambda_3 L^{\mathrm{V}}(f^{\mathrm{TJVE}}), \quad (1)
\end{aligned}
$$

For each training and testing split, we measured the average error rates on (a) a subset of the test set consisting of only interacting hands and (b) the entire test set: As our discriminator takes effect only on interacting hands, only those test instances in set (a) benefit from this additional supervision. Our final system measured the average error rate reduction of 4.26% and 1.14% on (a) and (b), respectively. We performed a t-test with 5% significance level of null hypothesis (the difference is insignificant): For both (a) and (b), our pose discriminator contributes to *significantly* improve the performance.

4. Generating (pseudo) visibility ground-truth

To train the joint visibility estimation network f^{TJVE}, we generated (pseudo) visibility ground-truths: For each input image, hand meshes were first synthesized by fitting MANO models where the parameters are estimated using NeuralAnnot [1]. Then, a depth image was generated by rendering these hand meshes. Next, the visibility of each joint was determined based on the difference between the ground-truth depth value of that joint and the corresponding value of the rendered depth map. If the rendered depth value is larger than the ground-truth depth by a certain margin, we concluded that the corresponding joint is occluded. The margins values were empirically determined at 5mm and 2mm for wrist and the other skeletal joints, respectively. Figure 2 shows examples of the generated ground-truth joint visibility maps while Fig. 3 visualizes the resulting 2D hand heatmap estimation pipeline.

Table 2: Error rates (MPJPE; in mm) of our system trained with (Ours [Final]) and without (Ours [$-L^{\mathrm{D}}$]) the supervision L^{D} from the joint hand pose discriminator on ten different training and testing splits of *InterHand2.6M*: (a) results on a subset consisting of only interacting hands and (b) results on the entire dataset.

	Split	1	2	3	4	5	6	7	8	9	10	Avg.
(a)	Ours ($-L^{\mathrm{D}}$)	12.17	12.49	12.27	12.24	12.09	12.31	12.07	12.04	12.28	12.25	12.22
	Ours (Final)	11.75	12.02	11.76	11.72	11.43	11.85	11.61	11.48	11.78	11.65	11.70
(b)	Ours ($-L^{\mathrm{D}}$)	11.35	11.53	11.45	11.34	11.22	11.50	11.18	11.23	11.44	11.34	11.36
	Ours (Final)	11.25	11.41	11.33	11.21	11.06	11.39	11.07	11.10	11.32	11.19	11.23

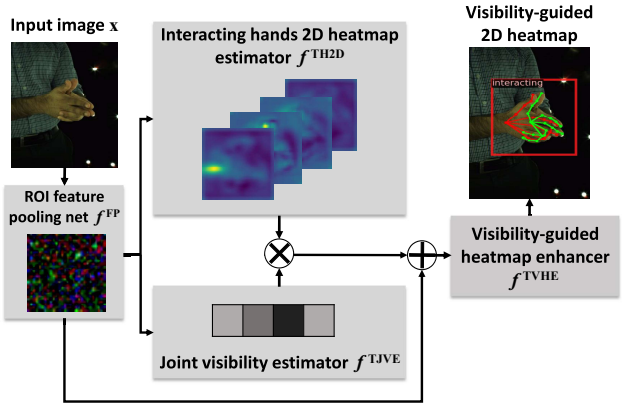


Figure 3: An illustration of the proposed 2D hand heatmap estimation pipeline.

5. Network architectures

Tables 3–10 present the details of our network architectures.

6. Additional examples

Figure 5 demonstrates that estimating and exploiting the joint visibility helps improve the joint estimation accuracy. Figures 6–10 show example images and the corresponding hand pose estimation results of Moon et al.'s algorithm [2] (using their code) and ours on single-hand as well as interacting hands cases. Both Moon et al.'s approach and our algorithm generated accurate pose estimates for single-hand cases. However, for interacting hands, severe occlusions and interference can pose significant challenges even for state-of-the-art Moon et al.'s approach (fourth to sixth rows in Fig. 7 and second and third rows in Fig. 9, and first and forth rows in Fig. 10). By exploiting the dependence of interacting hands, our approach can provide higher quality estimates.

References

[1] Gyeongsik Moon and Kyoung Mu Lee. NeuralAnnot: Neural annotator for in-the-wild expressive 3d human pose and mesh training sets. *arXiv preprint arXiv:2011.11232*, 2020. 2

[2] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. InterHand2.6M: A new large-scale dataset and baseline for 3D single and interacting hand pose estimation from a single RGB image. In *ECCV*, 2020. 3, 6, 7, 8, 9, 10

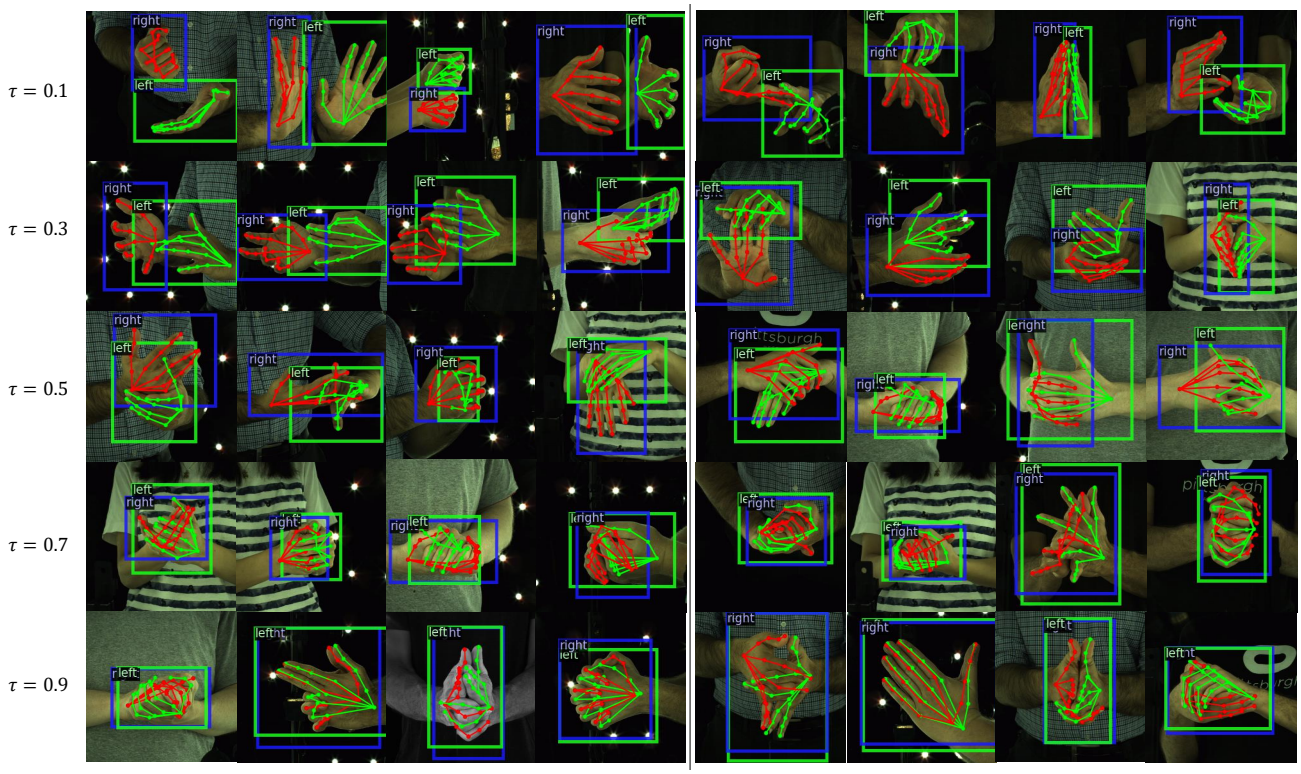
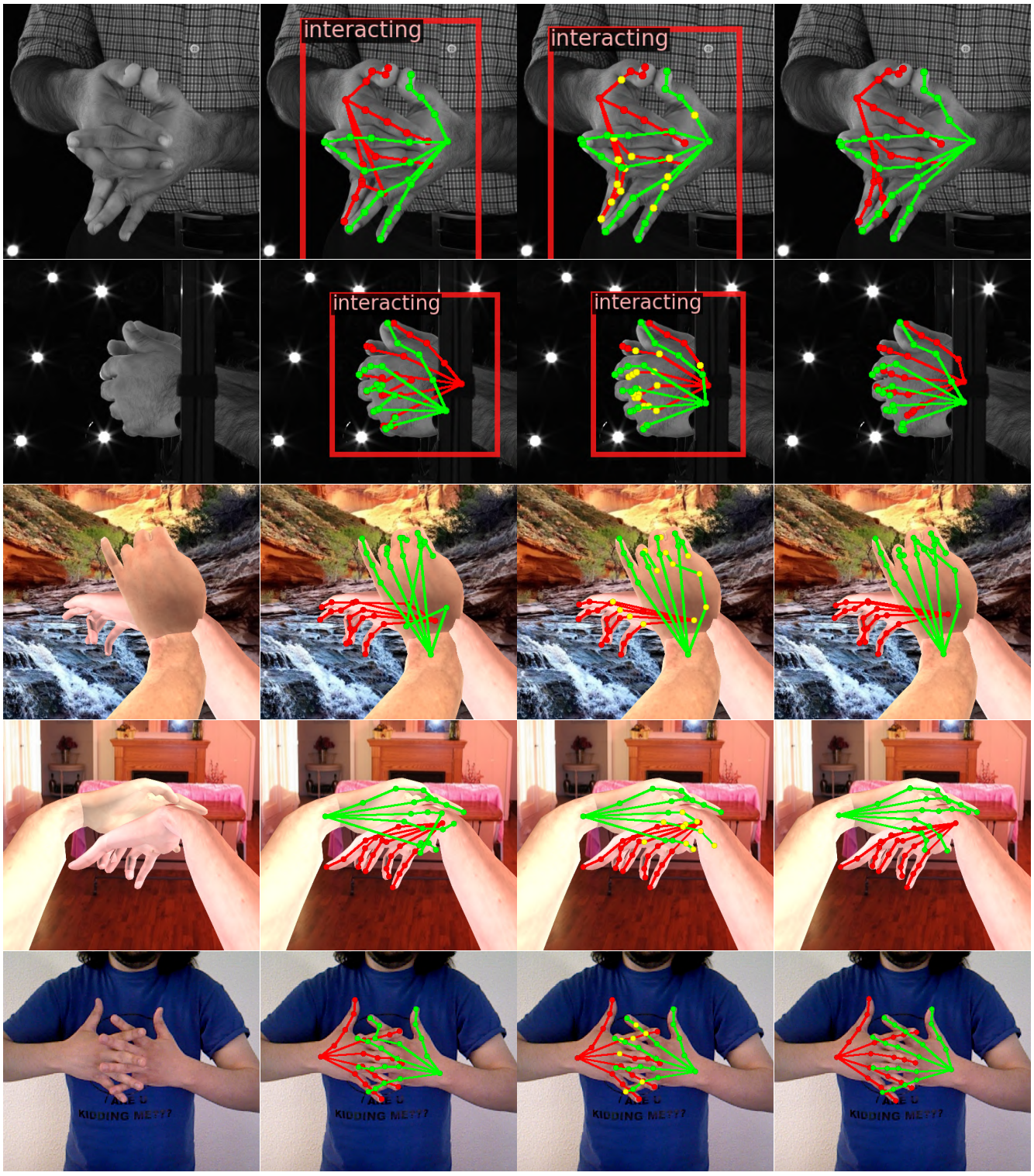
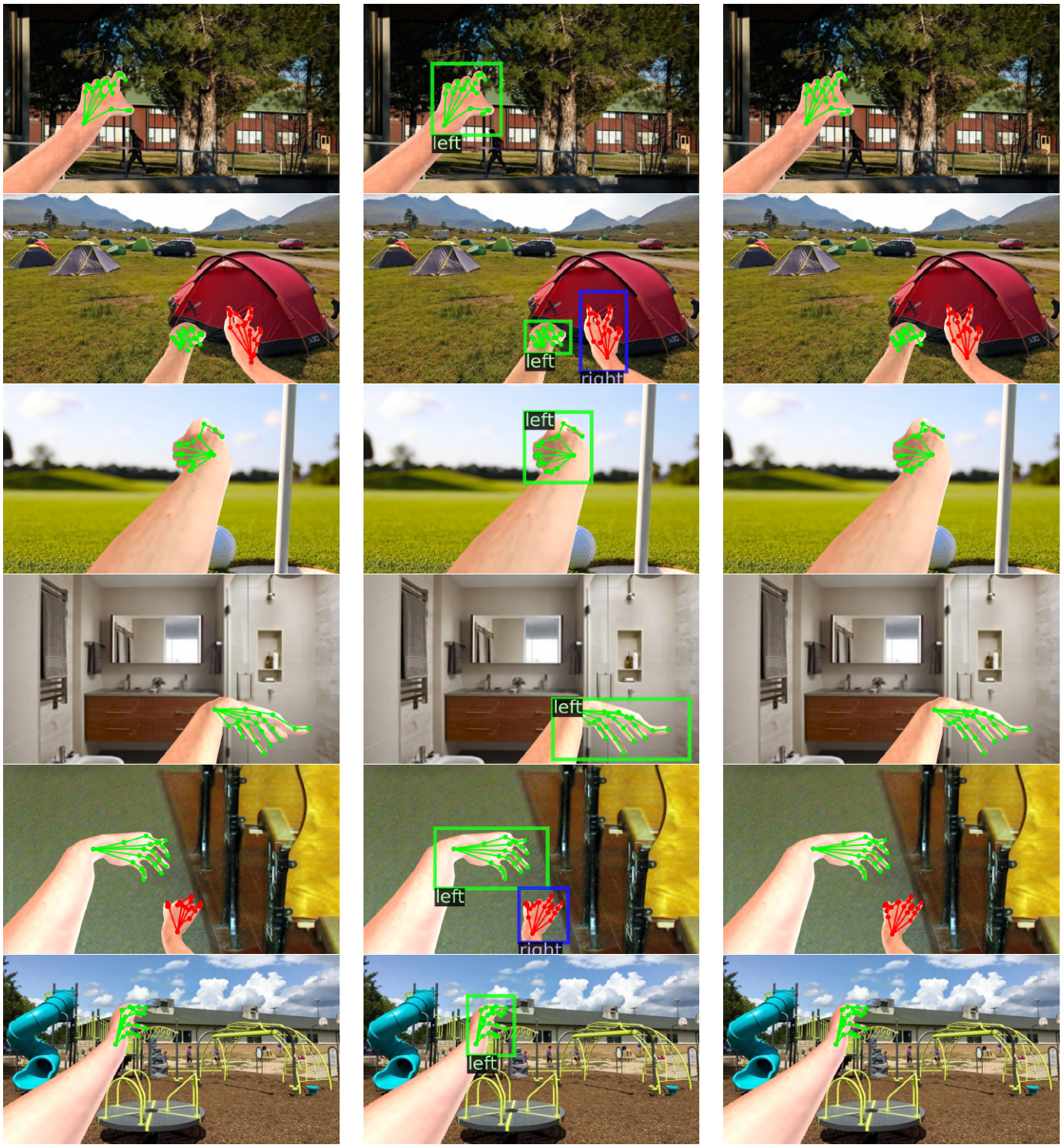


Figure 4: Example classification of training images under varying IOU thresholds τ: (Left 4 columns) single hands whose IOU is $\tau - \epsilon$; (Right 4 columns) interacting hands whose IOU is $\tau + \epsilon$ where $\epsilon \approx 0.05$.



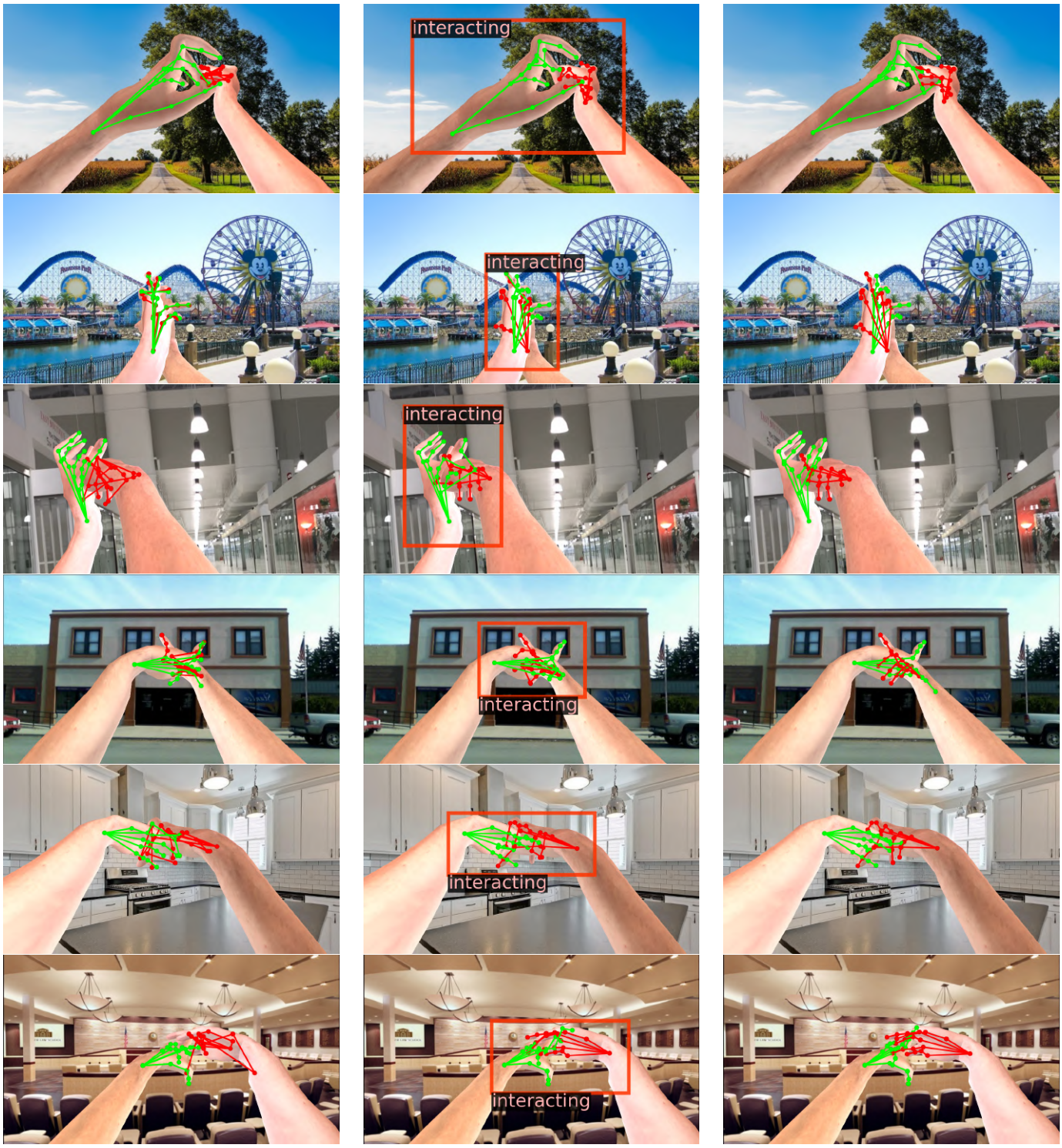
(a) (b) (c) (d)

Figure 5: Hand pose estimation examples on *InterHand2.6M* (Rows 1-2), *Ego3D* (Rows 3-4) and *Tzionas* (Row 5): (a) input images; (b) results of our algorithm without using the visibility-guided heatmap enhancement network; (c) final results of our algorithm including visibility-guided heatmap enhancement: The yellow points highlight occluded joints. Explicitly estimating and exploiting the joint visibility helps improve the joint estimation accuracy (rows 2-5); (d) ground-truths.



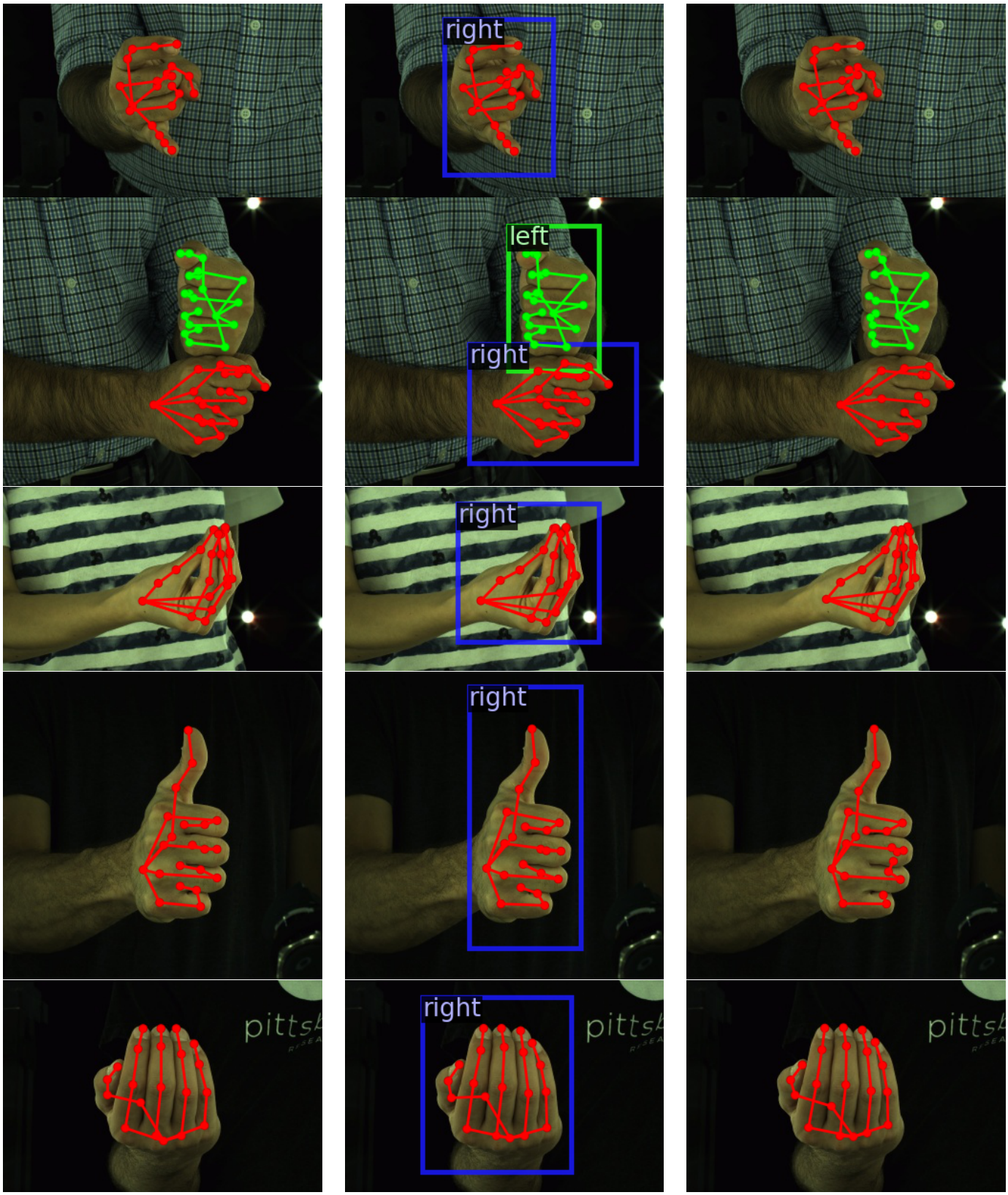
(a) Moon et al. [2] (b) Ours (c) Ground-truths

Figure 6: Hand pose estimation examples on *Ego3D* (single-hand cases).



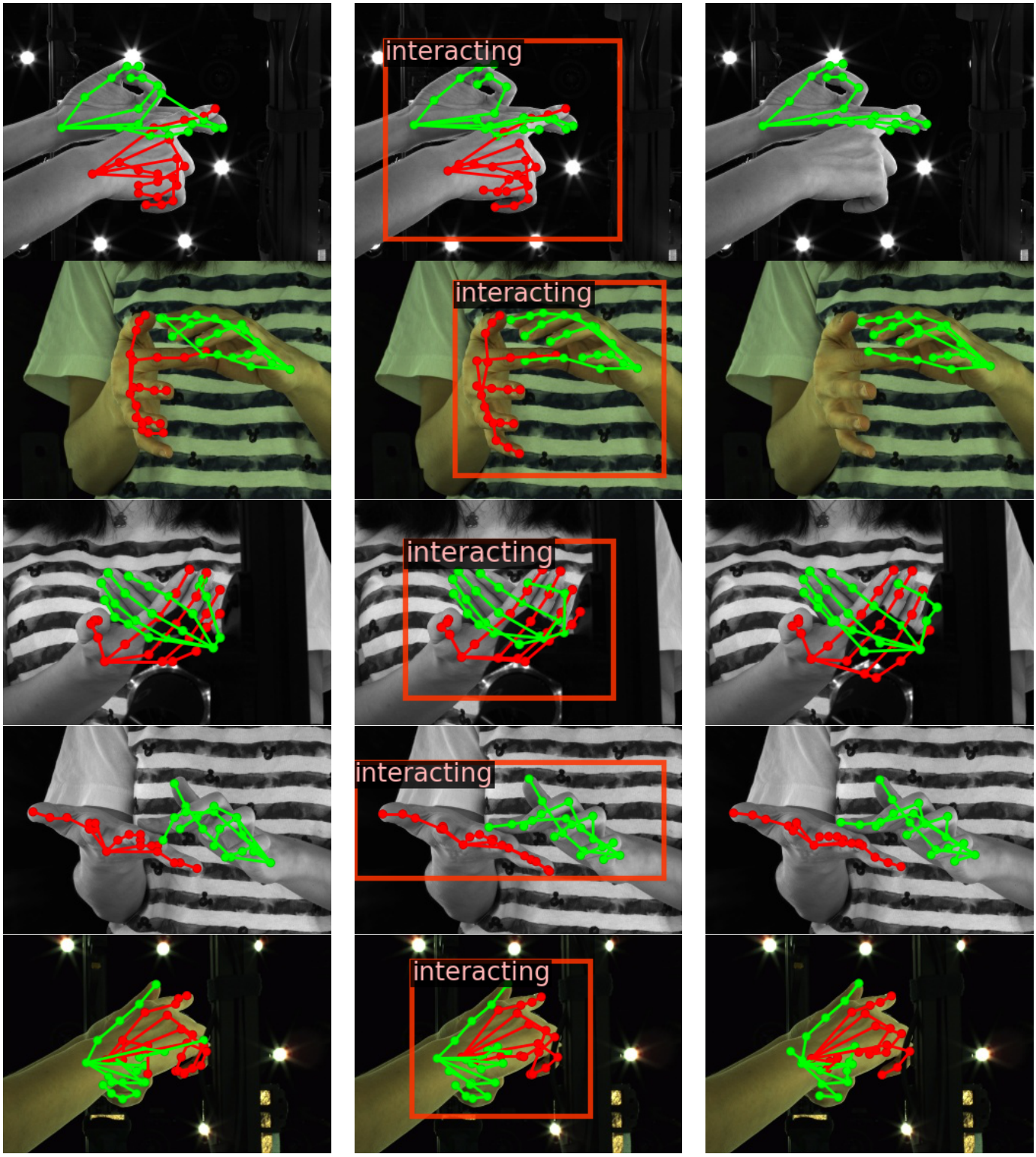
(a) Moon et al. [2] (b) Ours (c) Ground-truths

Figure 7: Hand pose estimation examples on *Ego3D* (interacting hands cases).



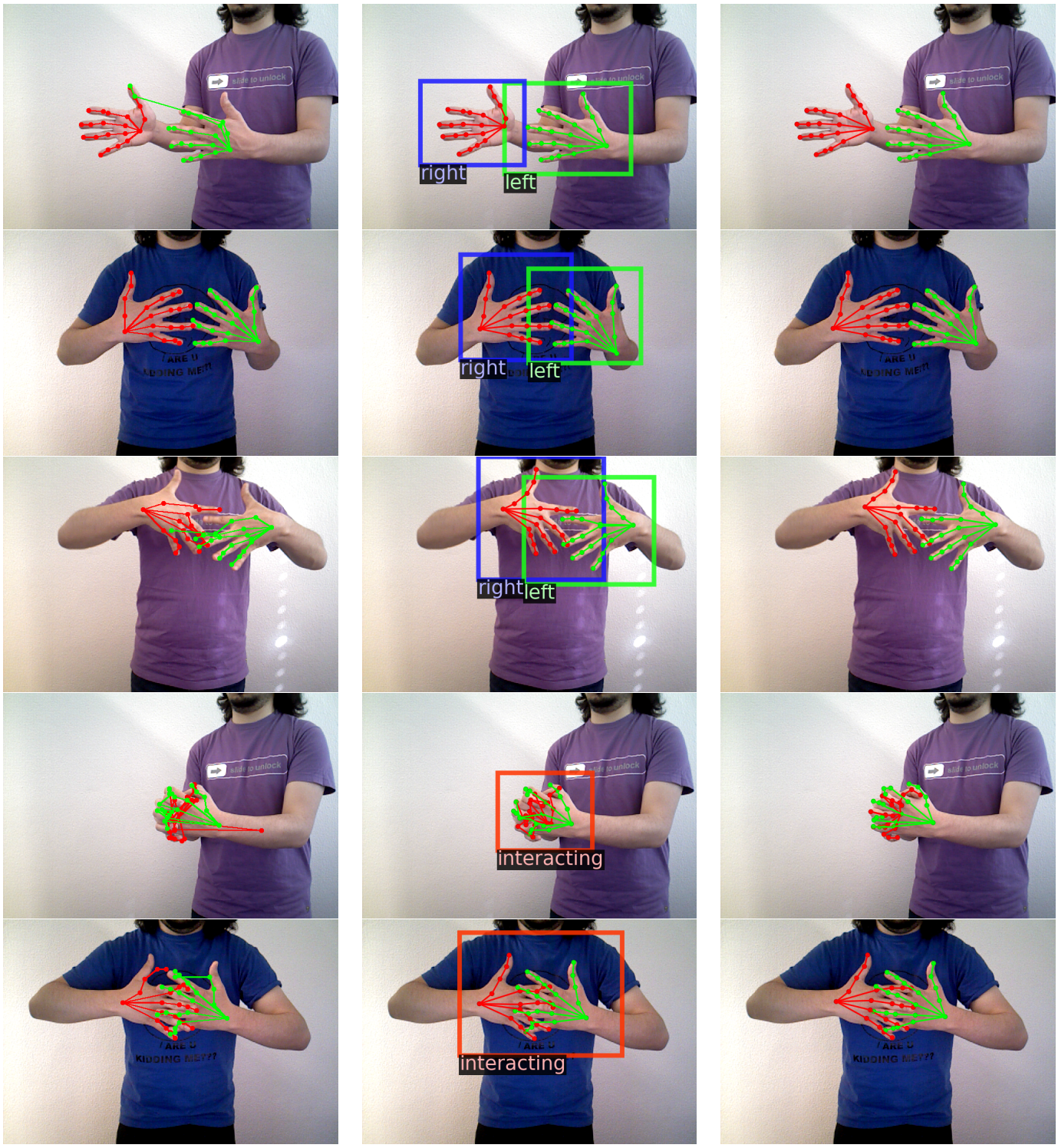
(a) Moon et al. [2] (b) Ours (c) Ground-truths

Figure 8: Hand pose estimation examples on *InterHand2.6M* (single-hand cases).



(a) Moon et al. [2] (b) Ours (c) Ground-truths

Figure 9: Hand pose estimation examples on *InterHand2.6M* (interacting hands cases).



(a) Moon et al. [2] (b) Ours (c) Ground-truths

Figure 10: Hand pose estimation examples on *Tzionas*.

Table 3: Architecture of single-hand 2D heatmap estimation network f^{SH2D}

Layer	Operation	Kernel	Dimensionality
	Input: ROI pooled features	-	$28 \times 28 \times 256$
1	Conv. + ReLU	3×3	$28 \times 28 \times 512$
2	Conv. + ReLU	3×3	$28 \times 28 \times 512$
3	Conv. + ReLU	3×3	$28 \times 28 \times 512$
4	Conv. + ReLU	3×3	$28 \times 28 \times 512$
5	Conv. + ReLU	3×3	$28 \times 28 \times 512$
6	Conv. + ReLU	3×3	$28 \times 28 \times 512$
7	Conv. + ReLU	3×3	$28 \times 28 \times 512$
8	Conv. + ReLU	3×3	$28 \times 28 \times 512$
9	Conv. + ReLU	3×3	$28 \times 28 \times 21$

Table 4: Architecture of interacting hand 2D heatmap estimation network f^{TH2D}

Layer	Operation	Kernel	Dimensionality
	Input: ROI pooled features	-	$28 \times 28 \times 256$
1	Conv. + ReLU	3×3	$28 \times 28 \times 768$
2	Conv. + ReLU	3×3	$28 \times 28 \times 768$
3	Conv. + ReLU	3×3	$28 \times 28 \times 768$
4	Conv. + ReLU	3×3	$28 \times 28 \times 768$
5	Conv. + ReLU	3×3	$28 \times 28 \times 768$
6	Conv. + ReLU	3×3	$28 \times 28 \times 768$
7	Conv. + ReLU	3×3	$28 \times 28 \times 768$
8	Conv. + ReLU	3×3	$28 \times 28 \times 768$
9	Conv. + ReLU	3×3	$28 \times 28 \times 42$

Table 5: Architecture of Joint visibility estimation network f^{TJVE}

Layer	Operation	Kernel	Dimensionality
	Input: Feature map	-	$28 \times 28 \times 256$
1	Global Average pooling	28×28	$1 \times 1 \times 256$
2	Flatten	-	256
3	Linear + GroupNorm(16) + ReLU	-	512
4	Linear	-	42
6	Sigmoid	-	42

Table 6: Architecture of visibility-guided heatmap enhancement network f^{TVHE}

Layer	Operation	Kernel	Dimensionality
	Input: (Feature map + heatmap) combinations	-	$28 \times 28 \times 298$
1	Conv. + GroupNorm(16) + ReLU	3×3	$28 \times 28 \times 512$
2	Conv. + GroupNorm(16) + ReLU	3×3	$28 \times 28 \times 512$
3	Conv. + ReLU	3×3	$28 \times 28 \times 42$

Table 7: Architecture of single-hand 3D depth value estimation net f^{SZ3D}

Layer	Operation	Kernel	Dimensionality
	Input: (Feature map + heatmap) combinations	-	$28 \times 28 \times 277$
1	Conv. + ReLU	3×3	$14 \times 14 \times 512$
2	Conv. + ReLU	3×3	$7 \times 7 \times 512$
3	Flatten	-	$25,088$
4	Linear	-	512
5	Linear	-	21
6	Sigmoid	-	21

Table 8: Architecture of interacting-hands 3D depth value estimation net f^{TZ3D}

Layer	Operation	Kernel	Dimensionality
	Input: (Feature map + 2 heatmap) combinations	-	$28 \times 28 \times 298$
1	Conv. + ReLU	3×3	$14 \times 14 \times 512$
2	Conv. + ReLU	3×3	$7 \times 7 \times 512$
3	Flatten	-	$25,088$
4	Linear	-	512
5	Linear	-	42
6	Sigmoid	-	42

Table 9: Architecture of interacting-hands 2D heatmap discriminator d^{TH2D}

Layer	Operation	Kernel	Dimensionality
	Input: 2D heatmaps	-	$56 \times 56 \times 42$
1	Conv. + LeakyReLU(0.2)	4×4	$28 \times 28 \times 128$
2	Conv. + LeakyReLU(0.2)	4×4	$14 \times 14 \times 256$
3	Conv. + LeakyReLU(0.2)	4×4	$7 \times 7 \times 512$
4	Conv. + LeakyReLU(0.2)	4×4	$3 \times 3 \times 512$
5	Conv.	3×3	$1 \times 1 \times 1$
6	Sigmoid		

Table 10: Architecture of interacting-hands 3D pose discriminator d^{TZ3D}

Layer	Operation	Kernel	Dimensionality
	Input: 3D skeletal joints	-	126
1	Linear + LeakyReLU(0.2)	-	512
2	Linear + LeakyReLU(0.2)	-	1024
3	Linear + LeakyReLU(0.2)	-	2048
4	Linear + LeakyReLU(0.2)	-	4096
5	Linear	-	1
6	Sigmoid		