Supplementary Material for "HandOccNet: Occlusion-Robust 3D Hand Mesh Estimation Network"

JoonKyu Park[1*] Yeonguk Oh[1*] Gyeongsik Moon[1*] Hongsuk Choi[1] Kyoung Mu Lee[1,2]
[1]Dept. of ECE & ASRI, [2]IPAI, Seoul National University, Korea
jkpark0825@snu.ac.kr, namepllet1@gmail.com, {mks0601, redarknight, kyoungmu}@snu.ac.kr

In this supplementary material, we first describe the specifications of FIT and SET in Section S1. In Section S2, we show quantitative results of our HandOCCNet before procrustes alignment to further justify our model. In Section S3, we show comparisons on the Dex-YCB dataset, which not only presents severe hand-object occlusion but also contains larger data. In Section S4, we provide additional visual comparisons of hand mesh estimation with the proposed HandOccNet and other state-of-the-art methods.

S1. Specifications of FIT and SET

FIT injects hand information into the correlated occlusion region, and SET refines $\mathbf{F}_{\text{FIT}}$ by referencing the distant information from $\mathbf{F}_{\text{FIT}}$. In this section, we cover the detailed specification of each Transformer-based module.

S1.1. FIT

We show the inference process of FIT in Algorithm 1. From the secondary feature $\mathbf{F}_{\text{S}}$ and primary feature $\mathbf{F}_{\text{P}}$, the query and key features, q_{soft} and k_{soft} are computed by 1×1 convolution, respectively. The query and key are matrix multiplied to produce an attention map, $\mathbf{C}_{\text{soft}}$. To attenuate undesirable high attention scores from low matrix multiplication output in $\mathbf{C}_{\text{soft}}$, we compute an additional attention map $\mathbf{C}_{\text{sig}}$ from matrix multiplication of additional query q_{sig} and key k_{sig}. All elements of $\mathbf{C}_{\text{sig}}$ range from 0 to 1. $\mathbf{C}_{\text{soft}}$ and $\mathbf{C}_{\text{sig}}$ are multiplied to produce a scaled attention map $\mathbf{C}$, and matrix multiplied with a value feature from the primary feature $\mathbf{F}_{\text{P}}$ to get the residual feature $\mathbf{R}_{\text{FIT}}$. Note that we do not use any residual connection to get $\mathbf{R}_{\text{FIT}}$. The final output of FIT is obtained by feeding $\mathbf{R}_{\text{FIT}}$ into a feed-forward module with the residual connection between the module's input and output. We also add a residual connection between its output and primary feature $\mathbf{F}_{\text{P}}$.

S1.2. SET

In Algorithm 2, we show the inference process of our proposed SET. Following the self-attention scheme in Transformer [11], we constrain the sum of attention values to be 1 by only adopting a softmax-based attention module. We also follow the same pipeline of previous Transform-

Algorithm 1 Pseudocode of FIT in a PyTorch-style

```
class FIT(nn.Module):
    def __init__():
        Q_soft = nn.Conv2d(256, 256, 1)
        K_soft = nn.Conv2d(256, 256, 1)
        Q_sig = nn.Conv2d(256, 256, 1)
        K_sig = nn.Conv2d(256, 256, 1)
        V = nn.Conv2d(256, 256, 1)
        η : R^{256×32×32} → R^{1024×256} # reshape
        ψ : R^{1024×256} → R^{256×32×32} # reshape
        softmax = nn.Softmax(dim=-1)
        pool = nn.AvgPool1d(1024, dim=-1)
        LN = nn.LayerNorm(256, dim=-1)
        MLP = nn.Sequential(
                    nn.Linear(256, 256*4),
                    nn.Linear(256*4, 256))
        d_{k_soft}, d_{k_sig} = 256, 256
    def forward(F_S, F_P): # F_S and F_P ∈ R^{256×32×32}
        # get queries, keys, and value
        q_soft = η(Q_soft(F_S))
        q_sig = η(Q_sig(F_S))
        k_sig = η(K_sig(F_P))
        k_sig = η(K_sig(F_P))
        v = η(V(F_P))
        # softmax-based attention module
        C_soft = softmax(matmul(q_soft, k_soft^T)/sqrt(d_{k_soft}))
        # sigmoid-based attention module
        C_sig = sigmoid(pool(matmul(q_sig, k_sig^T)/sqrt(d_{k_sig})))
        C = elemmul(C_soft, C_sig)
        # get residual feature R_FIT
        R_FIT = matmul(C, v)
        # feed-forward module
        F_FIT = F_P + ψ(R_FIT) + ψ(MLP(LN(R_FIT)))
        return F_FIT
```

* Authors contributed equally.

Algorithm 2 Pseudocode of SET in a PyTorch-style

```
class SET(nn.Module):
    def __init__()
        Q′ = nn.Conv2d(256, 256, 1)
        K′ = nn.Conv2d(256, 256, 1)
        V′ = nn.Conv2d(256, 256, 1)
        η : ℝ^{256×32×32} → ℝ^{1024×256} # reshape
        ψ : ℝ^{1024×256} → ℝ^{256×32×32} # reshape
        softmax = nn.Softmax(dim=-1)
        LN = nn.LayerNorm(256, dim=-1)
        MLP = nn.Sequential(
                        nn.Linear(256, 256*4),
                        nn.Linear(256*4, 256))
        d_{k′} = 256
    def forward(F_FIT): # F_FIT ∈ ℝ^{256×32×32}
        # get query, key, and value
        q′ = η(Q′(F_FIT))
        k′ = η(K′(F_FIT))
        v′ = η(V′(F_FIT))
        # softmax-based attention module
        C′ = softmax(matmul(q′, k′^T)/√d_{k′})
        # get residual feature R_SET
        R_SET = matmul(C′, v′) + q′
        # feed-forward module
        F_SET = ψ(R_SET) + ψ(MLP(LN(R_SET)))
        return F_SET
```

ers [11] to get the residual feature $\mathbf{R}_{\text{SET}}$ and the final output of SET, $\mathbf{F}_{\text{SET}}$.

S2. Evaluation: Before Procrustes Alignment

As results of hand mesh estimation before procrustes alignment are also significantly important in the literature, we further compare our HandOccNet to the state-of-the-art methods on the HO3D dataset in Table S1. As can be seen, our method still achieves better MPJPE before procrustes alignment.

models	Pose2Mesh [2]	Hasson *et al. et al.* [6]	I2L-MeshNet [9]	Liu *et al.* [8]]	**HandOccNet**
MPJPE	33.2	55.2	26.8	30.0	**24.9**

Table S1. MPJPE before procrustes alignment comparison with state-of-the-art method on HO-3D.

S3. Results on the larger dataset, Dex-YCB [1]

We further compare our model to [8], the second highest performing model on HO3D and FPHA datasets, in Table S2. Dex-YCB consists of 582K RGB-D frames over 1,000 sequences of 10 subjects grasping 20 different objects from 8 independent views. Therefore, evaluation on the Dex-YCB dataset could further justify our model's robustness to the situation where the hand is severely occluded.

method	METRO [7]	Spurr *et al.* [10]	Liu *et al.* [8]	**HandOccNet**
MPJPE	15.24	17.34	15.28	**14.04**
PA-MPJPE	6.99	6.83	6.58	**5.80**

Table S2. Comparison with state-of-the-art methods on Dex-YCB dataset.

S4. Qualitative comparisons

Figure S1 shows more qualitative comparisons on HO-3D [4]. Figure S2 further shows the qualitative comparisons on severely occluded images on HO-3D. The figure shows that our HandOccNet can robustly estimate the 3D hand mesh when hands are severely occluded by objects. This is due to our feature injection mechanism, which injects hand information into the occluded region and utilizes the injected information for the 3D hand mesh estimation. For example, in the first row and fourth row in Figure S2, thumbs are better reconstructed by injecting the relevant hand information into the occluded region. Figure S3 shows that our HandOccNet produces better results than Hasson *et al.* [6] on FPHA [3].

License of the Used Assets

- HO-3D dataset [4] is a publicly available dataset released under GNU GENERAL PUBLIC LICENSE v3.0.
- FPHA dataset [3] is academically released dataset under Imperial College London.
- I2L-MeshNet [9] codes are released for academic research only and it is free to researchers from educational or research institutes for non-commercial purposes.
- Pose2Mesh [2] codes are released for academic research only and it is free to researchers from educational or research institutes for non-commercial purposes.
- Hasson *et al.* [5] codes are released for academic research only and it is free to researchers from educational or research institutes for non-commercial purposes.
- Hasson *et al.* [6] codes are released for academic research only and it is free to researchers from educational or research institutes for non-commercial purposes.
- Liu *et al.* [8] codes are released for academic research only and it is free to researchers from educational or research institutes for non-commercial purposes.
- METRO [7] codes are released under the MIT license.

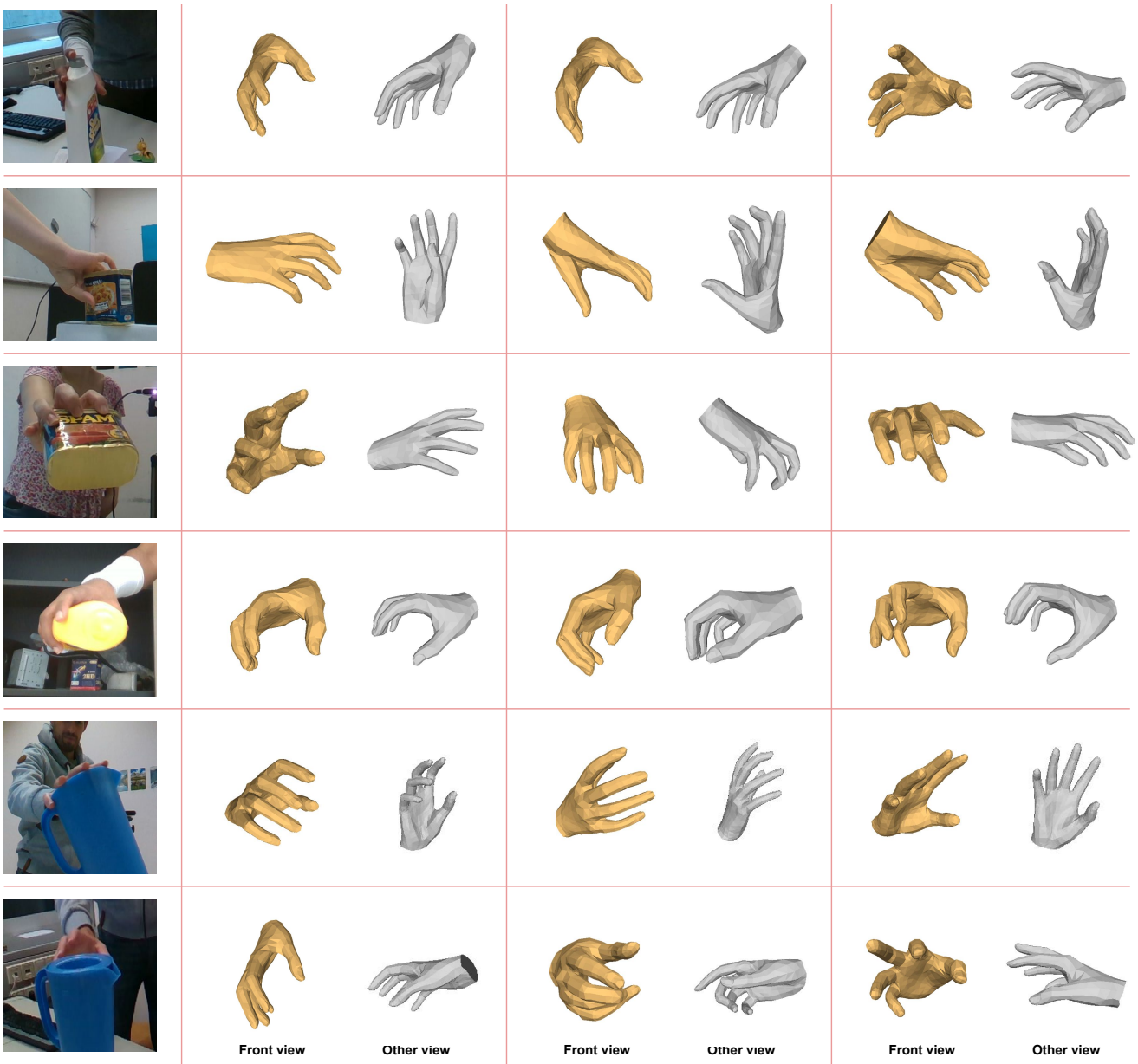

Figure S1. Qualitative comparisons of the proposed HandOccNet and state-of-the-art 3D hand mesh estimation methods [5, 8] on HO-3D [4].

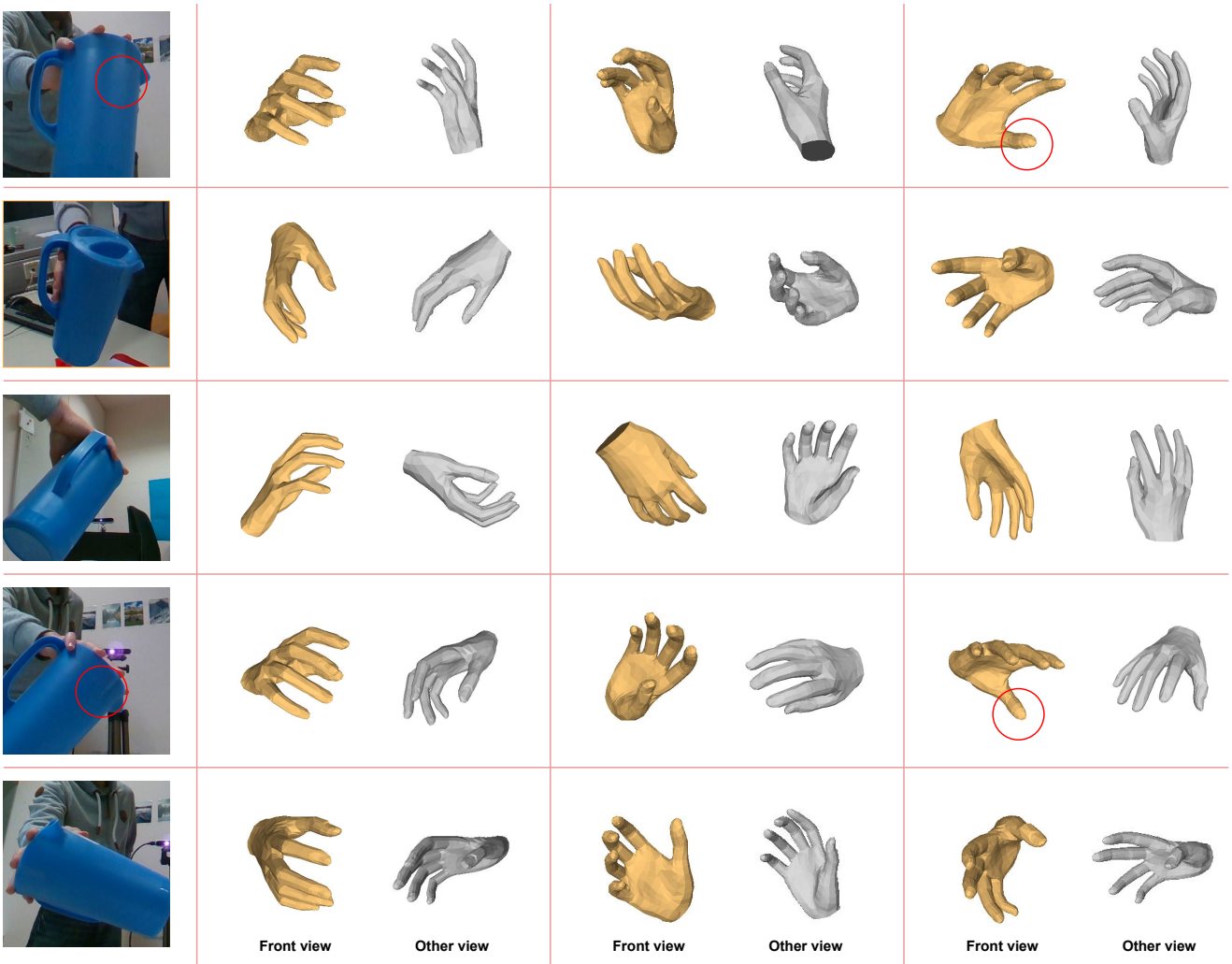

Figure S2. Qualitative comparisons of the proposed HandOccNet and state-of-the-art 3D hand mesh estimation methods [5, 8] on images of HO-3D [4] that contain severe occlusions.

Figure S3. Qualitative comparisons of the proposed HandOccNet and state-of-the-art 3D hand mesh estimation methods [6] on FPHA [3].

References

[1] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. DexYCB: A benchmark for capturing hand grasping of objects. In *CVPR*, 2021. 2

[2] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2Mesh: Graph convolutional network for 3D human pose and mesh recovery from a 2D human pose. In *ECCV*, 2020. 2

[3] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with RGB-D videos and 3D hand pose annotations. In *CVPR*, 2018. 2, 5

[4] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. HOnnotate: A method for 3d annotation of hand and object poses. In *CVPR*, 2020. 2, 3, 4

[5] Yana Hasson, Bugra Tekin, Federica Bogo, Ivan Laptev, Marc Pollefeys, and Cordelia Schmid. Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction. In *CVPR*, 2020. 2, 3, 4

[6] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *CVPR*, 2019. 2, 5

[7] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *CVPR*, 2021. 2

[8] Shaowei Liu, Hanwen Jiang, Jiarui Xu, Sifei Liu, and Xiaolong Wang. Semi-supervised 3D hand-object poses estimation with interactions in time. In *CVPR*, 2021. 2, 3, 4

[9] Gyeongsik Moon and Kyoung Mu Lee. I2L-MeshNet: Image-to-lixel prediction network for accurate 3D human pose and mesh estimation from a single RGB image. In *ECCV*, 2020. 2

[10] Adrian Spurr, Umar Iqbal, Pavlo Molchanov, Otmar Hilliges, and Jan Kautz. Weakly supervised 3D hand pose estimation via biomechanical constraints. In *ECCV*, 2020. 2

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1, 2