HOISDF: Constraining 3D Hand-Object Pose Estimation with Global Signed Distance Fields

Supplementary Material

We first provide additional details on the architecture design of HOISDF with respect to the image feature extraction and hand pose regression. Then, we provide additional details for the ablation experiments. Finally, we conduct additional experiments to assess the effectiveness of HOISDF.

1. Architecture details

1.1. Image Feature Extraction

Here, we detail the regressed objectives and the corresponding losses for the image backbone mentioned in Sec. 3.1. Following standard practice [19, 33, 49], we regress 2D heatmaps and hand/object segmentation masks as additional 2D predictions. Specifically, for simplicity, we regress a single-channel 2D hand keypoints heatmap $\mathbf{H}_h$ [19]. To obtain the ground-truth heatmap $\mathbf{H}^*_h$, we convolve all the 2D joint locations with a 2D Gaussian kernel and sum them in the same channel. Furthermore, we regress the hand and object segmentation maps ($\mathbf{H}_s$ and $\mathbf{O}_s$) as two additional channels. To learn $\mathbf{H}_h$, $\mathbf{H}_s$, and $\mathbf{O}_s$, we minimize the loss

$$\mathcal{L}_{img} = \|\mathbf{H}^*_h - \mathbf{H}_h\| + \mathcal{CE}(\mathbf{H}_s, \mathbf{H}^*_s) + \mathcal{CE}(\mathbf{O}_s, \mathbf{O}^*_s), \quad (10)$$

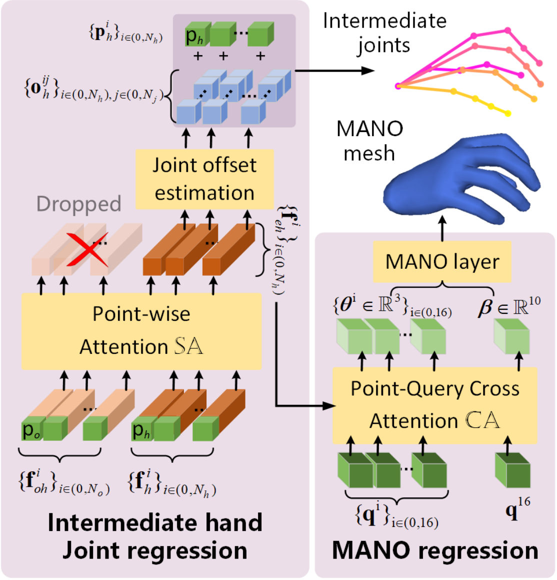
where $\mathcal{CE}$ represents the cross-entropy loss, and $\mathbf{H}^*_s$ and $\mathbf{O}^*_s$ are obtained by rendering the ground-truth 3D hand and object meshes.

1.2. Hand pose regression

In Sec. 3.2, we show that the field-guided pose regression module uses the point-wise features augmented by the field information to predict the hand object poses. Here, we give more details about the hand pose estimation component.

As is shown in Figure 5, with the set of hand query point features $\{\mathbf{f}^i_h\}_{i\in(0,N_h)}$ illustrated in Sec. 3.2.2 and the set of cross-hand query point features $\{\mathbf{f}^i_{oh}\}_{i\in(0,N_o)}$ illustrated in Sec. 3.2.3, we conduct point-wise attention $\mathbb{SA}$ between all the point features. The resulting features from $\{\mathbf{f}^i_h\}_{i\in(0,N_h)}$ are denoted as enhanced hand point features $\{\mathbf{f}^i_{eh}\}_{i\in(0,N_h)}$, while the resulting features from $\{\mathbf{f}^i_{oh}\}_{i\in(0,N_o)}$ are dropped since the object clues are already passed to $\{\mathbf{f}^i_{eh}\}_{i\in(0,N_h)}$ through $\mathbb{SA}$ (illustrated in Sec. 3.2.4).

We then conduct cross-attention between $\{\mathbf{f}^i_{eh}\}_{i\in(0,N_h)}$ and learnable queries $\{\mathbf{q}^i\}_{i\in(0,17)}$, where the last query is used to regress MANO shape parameters $\beta \in \mathbb{R}^10$ and the rest queries are used to regress MANO pose parameters $\{\boldsymbol{\theta}^i \in \mathbb{R}^3\}_{i\in(0,16)}$ (Eq. 8).



$\{\mathbf{p}^i_h\}_{i\in(0,N_h)}$
$\mathbf{p}_h$
$\{\mathbf{o}^{ij}_h\}_{i\in(0,N_h),j\in(0,N_j)}$
Intermediate joints
MANO mesh
Joint offset estimation
Dropped
$\{\mathbf{f}^i_{eh}\}_{i\in(0,N_h)}$
Point-wise Attention $\mathbb{SA}$
MANO layer
$\{\boldsymbol{\theta}^i \in \mathbb{R}^3\}_{i\in(0,16)}$
$\boldsymbol{\beta} \in \mathbb{R}^{10}$
Point-Query Cross Attention $\mathbb{CA}$
$\mathbf{p}_o$
$\mathbf{p}_h$
$\{\mathbf{f}^i_{oh}\}_{i\in(0,N_o)}$
$\{\mathbf{f}^i_h\}_{i\in(0,N_h)}$
$\{\mathbf{q}^i\}_{i\in(0,16)}$
$\mathbf{q}^{16}$
Intermediate hand Joint regression
MANO regression

Figure 5. **Details of hand pose regression of HOISDF.**

Meanwhile, similarly to Hampali et al. [19], we also regress the intermediate hand pose objective to guide the final predictions. However, since the features $\{\mathbf{f}^i_{eh}\}$ already contains rich 3D information, we directly regress 3D hand joints instead of 2D joints as in Hampali et al.[19] and use the query points as dense local regressors [31, 51]. Specifically, we use a joint offset regression head to predict the offsets $\{\mathbf{o}^{ij}_h\}_{i\in(0,N_h),j\in(0,N_j)}$ from a hand query point $\mathbf{p}^i_h$ to all the pose joints $\{\mathbf{h}^{*j}_p\}$, where j represents the pose joint index and N_j is the number of the hand pose joints. We use a smooth-L1 loss [43] to supervise the learning of the offsets. However, if $\mathbf{p}^i_h$ is far away from the pose joint $\mathbf{h}^{*j}_p$, the predicted $\mathbf{o}^{ij}_h$ could be inaccurate. Therefore, instead of regressing all the joint offsets, we use a joint visibility term to determine if $\mathbf{p}^i_h$ is close to $\mathbf{h}^{*j}$. We empirically set the joint class $\mathbf{v}^{*ij}_h$ to one if the distance between $\mathbf{p}^i_h$ and $\mathbf{h}^j$ is smaller than 4 cm, and to zero otherwise. The joint visibility information is not accessible during inference. Therefore, we introduce a joint classification head to learn it. To train it, we minimize the cross entropy loss $\mathcal{CE}$ between the predicted joint visibility $\mathbf{v}^{ij}_h$ and the ground truth $\mathbf{v}^{*ij}_h$. During inference, the predicted joint visibility $\{\mathbf{v}^{ij}_h\}_i$ is sent to the SoftMax function [3] to weigh the joint predictions. Altogether, this yields the training loss

$$\mathcal{L}_{\text{off}} = \sum_{i}^{N_h}\sum_{j}^{N_j} SmoothL1(\mathbf{p}_h^i + \mathbf{o}_h^{ij}, \mathbf{h}{*}_p^j) \cdot \mathbf{v}{*}_h^{ij} + \mathcal{CE}(\mathbf{v}_h^{ij}, \mathbf{v}{*}_h^{ij}). \quad (11)$$

2. Ablation Details

2.1. Comparison of different intermediate representations

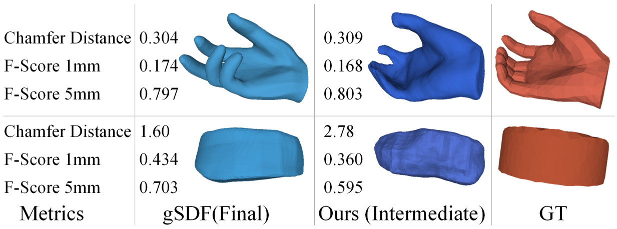
As discussed in Sec. 4.4, we replace the 3D field learning module (Sec. 3.1) with 2D keypoint learning, 2D segmentation learning, and 3D mesh learning. Here, we give more details for the model designs of using other intermediate representations. Specifically, for 2D keypoint learning, we borrow the model design of Hampali et al. [19] to regress identity-aware but part-agnostic keypoints in the intermediate step to serve as query points. For 2D segmentation learning, we use the pixel locations with segmentation scores larger than 0.3 as query points. The keypoint confidence and the segmentation score are used to multiply with the query point features separately to mimic our feature regularization (Sec. 3.2.2) in the above two baselines. For 3D mesh learning, we follow Tse et al. [47] to regress MANO parameters in the intermediate stage and use the MANO hand vertices to serve as hand query points. Meanwhile, we regress the object rotation and translation in the intermediate stage to obtain object vertices as object query points.

We find that the SDF representation outperforms the 2D representations by a large margin, especially in MJE and object metrics that exploit more global information (Table 5). We attribute this to the 2D intermediate representations gathering less 3D shape information in the initial step. Furthermore, we observe that using 3D vertices as intermediate representation performs better than 2D representations (Table 5). This supports our claim that implicit 3D shape representations are better than explicit 3D meshes.

2.2. Comparison to other SDF-based methods

The key difference between HOISDF and other SDF-based methods [12, 13, 54] is the role of the SDF module. Previous methods rely on the SDF module to reconstruct fine-grained hand-object surfaces. Predicting the SDF is the endpoint of the models. The resulting SDF values are used to generate meshes directly. By contrast, HOISDF shows that SDFs are a great intermediate representation for hand-object pose estimation (Table 5). The extracted SDF values are sent to the field-guided pose regression module to provide 3D global shape information for hand-object pose estimation. In comparison, we obtained better pose estimates than previous SOTA SDF methods [12, 13] (Table 2).

Due to different roles, the design choices of the SDF module in HOISDF and other SDF methods thus differ. To improve the quality of the reconstructed surfaces, previous methods [12, 13, 54] add intermediate pose regression modules. *The generated hand-object poses are used to pre-align the local parts with the canonical space. The SDF module can thus focus on fine-grained details without being disturbed by hand-object poses.* However, we aim to let the SDF module encode global pose information to guide the subsequent pose regression. We have evidence that adding a pose regression module before will convey unreliable pose information to the input of the SDF module and will pollute the global information captured by the SDF module (e.g., the little finger of gSDF's hand mesh in Figure 6). Meanwhile, additional pose regression and canonicalization steps would also decrease the running speed of HOISDF and make the module unable to be end-to-end trained [13, 54].



Metrics	gSDF(Final)	Ours (Intermediate)	GT
Chamfer Distance	0.304	0.309	
F-Score 1mm	0.174	0.168	
F-Score 5mm	0.797	0.803	
Chamfer Distance	1.60	2.78	
F-Score 1mm	0.434	0.360	
F-Score 5mm	0.703	0.595	

Figure 6. **Comparisons between HOISDF's intermediate results and gSDF's [13] final results on DexYCB testset.** The SDF module in HOISDF cares more about global plausibility, while the one in gSDF cares more about fine-grained surface reconstruction.

To support our design choices, we directly use the intermediate SDF module to reconstruct hand-object meshes and compare them with gSDF's [13] final outputs (Figure 6)). Note that HOISDF also yields 3D hand and object meshes in the final outputs and obtains SOTA results (Table 3 and Figure 4). Regarding our intermediate SDF module, we expect to have worse results since mesh reconstruction is not the goal of our SDF module. Surprisingly, however, it performs similarly to gSDF on hand metrics (Fig. 6). We attribute this to the fact that our SDF module captures better global shape information. Therefore, even though the mesh reconstruction quality is lower, the overall distance to the GT hand mesh is acceptable. In comparison, the poses of the meshes produced by gSDF are influenced by its pose regression module and might yield large pose errors. As expected, our intermediate SDF module performs worse than gSDF on object metrics because of worse surface reconstruction. However, the general pose of our intermediate object reconstruction remains satisfactory. Note that gSDF is trained for 1600 epochs, while HOISDF is only trained for 40. We also replace our SDF module with gSDF initialized by their trained weights. The results (MJE: 11.2, PAMJE: 5.83, OCE: 19.6, MCE: 29.4, ADD-S: 14.3) show that despite more computational complexity, gSDF is less effective as an intermediate module.

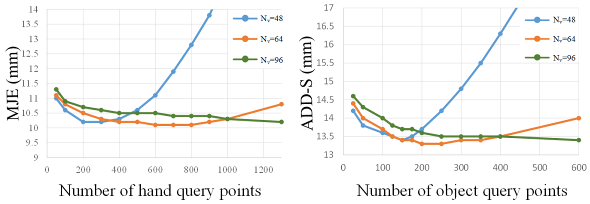


Figure 7. **Hand object performance curve according to the numbers of sampled query points on DexYCB testset.** HOISDF is robust with a wide range of sampled query points under different discretization sizes.

2.3. Ablations for the Field-guided Pose Regression Module

As discussed in Sec. 4.5, we verify the effectiveness of the components in our field-guided pose regression module by comparing each component with multiple variants. Here, we show the detailed designs of the variants.

Effectiveness of the field-informed point sampling. As discussed in Sec. 3.2.1, we sample query points close to the hand/object surfaces for the subsequent pose estimation. During inference, we sample query points with the smallest absolute distances to achieve the same goal. Here, we compare to three alternative point sampling strategies. The first one is to sample query points randomly in the 3D spaces. The second one is to sample query points inside the hand object meshes and sample points with the smallest signed distances during inference. The final one still samples points close to the hand-object surfaces. However, during the inference, we follow Zhou et al. [56] to compute the gradient of the SDF module according to a certain sampled query point. Then we multiply the gradient with the signed distance and use them as an offset to move the original sampled query point. This moves the query point even closer to the surfaces. Random sampling and signed distance sampling perform much worse than our absolute distance sampling, because the sampled points cannot reflect the general shapes of the hand and object and query irrelevant image features that will harm the pose estimation (Table 7). Applying field gradient to obtain the query points has almost the same performance as ours. However, computing the gradients for all the query points takes much more time compared to directly sampling points based on absolute distances. Therefore, in comparison our sampling strategy is the most efficient one.

Effectiveness of field-based point feature augmentation. As described in Sec. 3.2.2, we convert the point signed distance into a volume density and then multiply it with the point image feature to augment the feature. Since the cross hand object interaction (Sec. 3.2.3) also uses the feature augmentation and will influence the performance, we remove the cross field attention and implement three variants to verify the effectiveness of the feature augmentation (Table 8). Removing the SDF feature augmentation (*w/o SDF regularization*), concatenating rather than multiplying the volume density with the image feature (*w density concatenation*), and concatenating the distance value with the image feature (*w distance concatenation*). Removing the SDF regularization yields an accuracy drop. Directly concatenating the distance values makes the model struggle to extract useful information. Directly concatenating the density value boosts the performance compared to *w/o SDF regularization*. However, since it only has one dimension, it is hard to influence the whole feature representation.

Effectiveness of hand-object feature enhancement. As discussed in Sec. 3.2.3, we augment the object query point features with the cross-hand signed distances. The resulting cross-hand query point features are then used to conduct cross-attention with the original hand query point features to enhance the hand feature representation (Eqn. 7). Here, we conduct ablations to verify the effectiveness of our hand-object feature enhancement with three variants (Table 9): Removing the cross feature enhancement completely (denoted as w/o cross feature enhancement), cross attention with cross target image features f_{img} without feature augmentation (denoted as w cross image feature), cross attention with cross target features f_h and f_o (denoted as *w cross target feature*). Compared to *w/o cross feature enhancement*, both hand and object benefit from the cross target cues and improve the pose estimation performance. The variant *W cross image feature* only obtains very few improvements for the object pose estimation while has a side influence on the hand pose estimation. The object usually takes a larger space than the hand in the image. The various object features from different pixel locations will mislead the hand pose estimation without the guidance of the cross-hand signed distances. *W cross target feature* obtains the worst results for both hand and object pose estimations since the features are still augmented with the original signed distances instead of the cross-target signed distances, which are not helpful in transferring clues to the other target.

Robustness with various pose regression components. As mentioned in Sec. 3.2.5, we use learnable queries to conduct cross-attention with enhanced hand query point features $\{\mathbf{f}^i_{eh}\}$ and regress the MANO parameters. Note, however, that the strong hand pose estimation performance is mainly because of the field-based feature enhancement rather than the design of the hand pose regressor. To verify that, we also implement three other hand pose regressors (Table 10). The first one removes the intermediate hand joint regression. The second one removes the cross-attention layer and directly uses the intermediate hand joints as the final result. The last one only uses the cross-attention layer to regress the MANO shape parameters. The MANO pose parameters are inferred from the intermediate

Methods **Metrics in [mm]**	HOISDF (ours) OCE	 MCE	 ADD-S	Wang *et al.* [49] OCE	 MCE	 ADD-S
002_master_chef_can	**15.9**	**20.2**	**10.2**	21.8	25.5	12.8
003_cracker_box	**29.4**	40.2	18.5	33.3	**37.8**	**17.8**
004_sugar_box	**17.1**	**29.7**	**14.2**	24.6	32.3	14.7
005_tomato_soup_can	**17.9**	**20.8**	**10.3**	29.4	31.7	15.0
006_mustard_bottle	**13.6**	**18.1**	**9.1**	20.4	24.5	11.1
007_tuna_fish_can	**15.4**	**17.3**	**8.9**	23.6	24.5	12.5
008_pudding_box	**13.3**	**19.5**	**9.5**	21.0	24.5	12.1
009_gelatin_box	**14.8**	**20.8**	**9.8**	25.4	28.3	13.9
010_potted_meat_can	**13.9**	**19.8**	**10.5**	24.7	26.7	12.4
011_banana	**19.5**	**41.7**	**20.6**	28.1	42.2	21.0
019_pitcher_base	**27.9**	**39.5**	**18.8**	37.3	44.4	21.5
021_bleach_cleanser	**19.0**	40.9	18.6	34.4	**39.7**	**17.8**
024_bowl	**17.7**	**21.5**	**12.0**	28.5	30.2	16.1
025_mug	**16.5**	**17.9**	**9.5**	27.1	27.3	12.3
035_power_drill	**20.5**	31.2	16.1	26.8	**30.8**	**14.5**
036_wood_block	**27.9**	**35.3**	**17.1**	35.8	46.4	21.7
037_scissors	**25.4**	49.0	21.3	33.5	**47.8**	**22.8**
040_large_marker	**14.9**	**24.2**	**12.9**	25.1	31.8	18.3
052_extra_large_clamp	**23.7**	48.3	**22.4**	31.2	**45.8**	22.7
061_foam_brick	**13.7**	**16.3**	**8.0**	24.3	25.1	11.4
Mean	**18.4**	**27.4**	**13.3**	27.3	32.6	15.9

Table 11. **Per-object performance on DexYCB testset.** Our HOISDF can outperform Wang *et al.* [49] for most of the objects, demonstrating HOISDF is robust to various objects.

Methods **Metrics in [mm]**	HOISDF (ours) OME	 ADD-S	Wang *et al.* [49] OME	 ADD-S
006_mustard_bottle	42.6	**11.8**	**36.5**	16.3
010_potted_meat_can	**39.7**	**14.5**	48.6	22.1
021_bleach_cleanser	**29.5**	**15.1**	44.7	20.7
Mean	**35.5**	**14.4**	45.5	20.8

Table 12. **Per-object performance on HO3Dv2 testset.** HOISDF can outperform Wang *et al.* [49] on HO3Dv2 dataset as well.

hand joints using inverse kinematics adopted from Chen et al. [13]. We can observe removing the intermediate joint regression only drops very little on the performances. Removing the MANO regression drops slightly more in PAMJE since there is no constraint for the hand shape in the intermediate joints regression. To improve that, we add the MANO shape regression in the last variant and use the inverse kinematics to compute MANO pose parameters from the intermediate joints, which can are passed into MANO network to regress the hand mesh. We can see the performance is almost comparable with our current regressor.

Comparable performance with some variants. Here, we want to emphasize that the design logic is the most important contribution of each component in our field learning module. The comparable variants share the same key ideas with our module design. For example, *Field gradient* also samples points near the surface (Table 7), while *w density concatenation* also introduces distance-to-density [38] for SDF information encoding (Table 8). They were (our) intermediate designs to the final proposed module and lacked either efficiency or performance.

Robustness with different numbers of sampled points. As mentioned in Sec. 4.2, we sample $N_v^2/n_h = 600$ hand query points and $N_v^2/n_o = 200$ object query points with a discretization size of $N_v = 64$. Here, we sample different numbers of query points with different discretization sizes to verify that HOISDF is robust to a wide range of point sampling numbers (Fig. 7). We found that HOISDF is robust for reasonable numbers of query points. When increasing the number of query points for a discretization size of 48 one will sample many points that are far away from the hand/object, which results in large errors.

3. Inference Speed

Benefiting from the efficient way of using the field information in our field-guided pose regression module, our model can achieve real-time inference speed (30.7 FPS) on a single NVIDIA TITAN RTX GPU, which includes 10.6ms for image feature extraction, 11.5ms for query points sampling, and 10.9ms for pose attention and regression.

4. Additional results

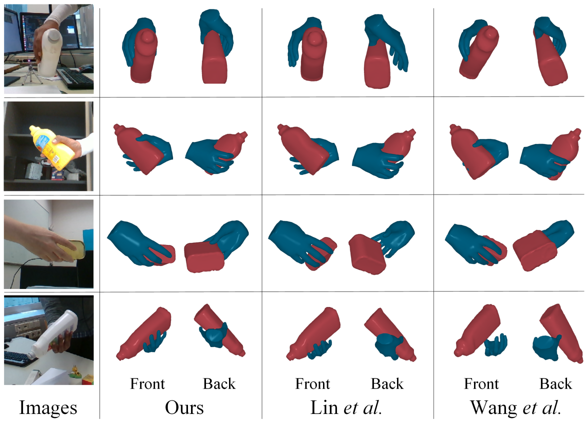


Figure 8. **Qualitative comparisons** on the HO3Dv2 test set with Lin et al. [33] and Wang et al. [49]. HOISDF can produce better hand-object poses under various hand object interactions.

4.1. Qualitative comparison on HO3Dv2 dataset

We visualize qualitative comparison with SOTA methods ([33, 49]) on the DexYCB dataset in Sec. 4.3. To further verify the effectiveness of HOISDF, we also show the qualitative comparison with the SOTA methods ([33, 49]) on the HO3Dv2 dataset (Figure 8). We can observe consistent improvements in HOISDF over the SOTA methods.

4.2. Per-object performances

We compare HOISDF with Wang et al.[49] that has SOTA object performances for every object category on DexYCB test set (Table 11) and HO3Dv2 test set (Table 12). We can observe that HOISDF outperforms Wang et al. [49] on almost all the object categories and all the metrics, which proves the effectiveness of our model for various objects.

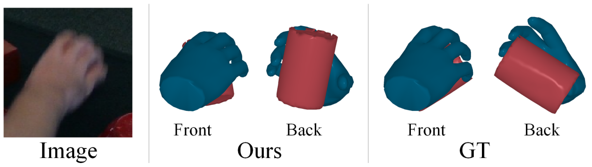


Figure 9. **Failure case of HOISDF.** Physical plausibility could be improved. For severely occluded scenarios, the predicted hand and object meshes might intersect with each other.

4.3. Failure cases and limitations

Although HOISDF obtains the SOTA results, it still has limitations. For severely occluded scenarios, the predicted hand and object meshes might intersect with each other (Figure 9). Therefore, some physical constraints could be modeled during hand object pose estimation to further improve the performance.