Enhancing 3D Hand Pose Estimation via Dense Ordinal Regression Network

Yamin Mao[1]
yamin18.mao@samsung.com
Zhihua Liu[1]
zhihua.liu@samsung.com
Weiming Li[1]
weiming.li@samsung.com
SoonYong Cho[2]
soonyong.cho@samsung.com
Qiang Wang[1]
qiang.w@samsung.com
Xiaoshuai Hao*[1]
xshuai.hao@samsung.com

[1] Samsung R&D Institute China–Beijing
Beijing, China
[2] Multimedia System TU
SAIT
SEC, Korea

Abstract

Depth-based 3D hand pose estimation is an important but challenging task in robotics and autonomous driving. Recently, more attention has been given to dense regression methods for this task. These methods offer a good balance between accuracy and computational efficiency through the densely regressing hand joint offset maps. Despite the benefits, large-scale regression offset values are often affected by noise and outliers, leading to a significant drop in accuracy. To address this issue, we re-formulate 3D hand pose estimation as a dense ordinal regression problem and introduced a new Dense Ordinal Regression 3D Pose Network (DOR3D-Net). Specifically, we first decompose offset value regression into sub-tasks of binary classifications with ordinal constraints. Then, each binary classifier can predict the probability of a binary spatial relationship relative to joint, which is easier to train and yield much lower level of noise. The estimated hand joint positions are inferred by aggregating the ordinal regression results at local positions with a weighted sum. Furthermore, both joint regression loss and ordinal regression loss are used to train our DOR3D-Net in an end-to-end manner. Extensive experiments on public datasets (ICVL, MSRA, NYU and HANDS2017) show that our design provides significant improvements over SOTA methods.

1 Introduction

High-quality hand pose estimation provides an important way of user interaction for various applications, such as robotics [8, 20], virtual reality [6, 33] and autonomous driving [32]. With the development of deep learning, hand pose estimation methods from RGB images [1, 13, 15, 24, 34] and depth images [2, 22, 23, 30] have attracted much attention.

* Corresponding author.

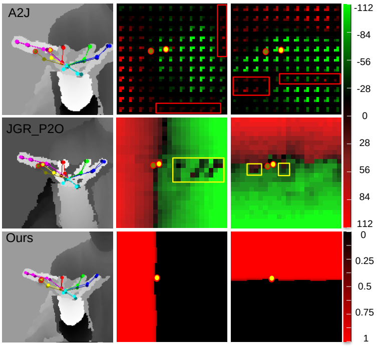

Figure 1: Visualization results of SOTA methods and our method. Row 1 and 2 show predictions from A2J [30] and JGR-P2O [5] respectively and row 3 shows our predictions. Column 1 shows the final result of prediction for an exemplar hand joint and column 2 and 3 show the *x*-offset and *y*-offset maps respectively. Notice that several error areas present in the offset maps from A2J and JGR-P2O (highlighted with red and yellow boxes). In contrast, our probability map is clean.

This paper focuses on improving depth-based 3D hand pose estimation task, which aims to output hand joint coordinates in 3D space from an input depth image.

Existing hand pose estimation methods with dominant performance employ deep learning models of different structures. Some models [3, 33] extract deep representations and then directly regress the joint coordinates or other forms of hand model parameters. Differently, more recent models explore dense predictions. They usually use a dense grid and for each grid position predict an offset vector that points to a joint. The densely predicted offset vectors form an offset map and then are used to infer joint coordinates. For example, A2J [30] proposes dense anchors and aggregates the offsets of these anchors to estimate each joint in the image plane. JGR-P2O [5] regresses joints by weighted averaging over all pixels' offsets in both image plane and depth space. For these methods, depending on the distance between a grid position and its target joint, the offset value varies in a large interval, especially for high-resolution images. However, large-scale regression offset values are often affected by noise and outliers. These flaws are difficult to completely remove and will propagate to subsequent steps resulting in degradation in the estimated joint accuracy. In this paper, we explore ordinal constraints to improve dense prediction methods for hand pose estimation. Specifically, as a point traverses in space along the scanline, the spatial relationship between the point's position to a target joint should vary smoothly with strict ordinal constraints. A closely related previous work [18] is ordinal regression which converts a regression task into a series of binary classifies with ordinal constraints. The ordinal regression has been proven to be useful for several tasks such as age estimation [18] and depth estimation [7].

To our best knowledge, we are the first re-formulate 3D hand pose estimation as a dense ordinal regression problem and propose a novel Dense Ordinal Regression 3D Pose Network (DOR3D-Net). Specifically, the design of DOR3D-Net includes: (1) The problem of hand joint regression in 3D is decomposed into sub-tasks of binary classifications. Each binary classifier is associated to a grid in 3D with different interval distributions in image and depth dimensions. Each binary classifier predicts probability of a binary spatial relationship between the position and a joint point. (2) The ordinal regression results at different local positions are aggregated to infer joint positions with weighted sum. This allows us using a joint position loss together with the ordinal regression loss to supervise DOR3D-Net.

The experiments show that the binary classifiers in our design are easy to train and yield much lower level of noise. Fig. 1 visualizes the offset maps predicted by A2J [30], JGR-P2O [5] and our probability map in image plane respectively. The first row shows the offset maps of A2J. Anchor offset values are wrong in several local areas (highlighted by red boxes). The boundary of zero offset value appears as a curve, which severely deviates from its ideal form as a straight line. For JGR-P2O, the learned offset maps also include apparent errors (highlighted by yellow boxes). In contrast, the probability maps generated by DOR3D-Net are much cleaner and well approximate those of their ideal forms. The main contributions are summarized as follows:

- We are the first to re-formulate the 3D hand pose estimation as a dense ordinal regression problem and propose a novel Dense Ordinal Regression 3D Pose Network.
- Specifically, we propose Ordinal Regression (OR) module to decompose offset value regression into sub-tasks of binary classifications with less noises and outliers. Furthermore, both joint regression loss and ordinal regression loss are used to train our DOR3D-Net in an end-to-end manner.
- DOR3D-Net is remarkably superior to SOTAs on existing methods, revealing the effectiveness of our approach.

2 Related Work

2.1 Depth Image Based Hand Pose Estimation

This paper focuses on the depth image-based 3D hand pose estimation task. According to summary of a large-scale public challenge HANDS2017 [31], state-of-the-art hand pose estimation methods can be roughly divided into two categories: regression-based methods and detection-based methods. Regression-based method directly regress hand joint parameters with extracted global feature representation. DenseRecurrent [3] uses PointNet network to extract features and iteratively refines the estimated hand pose with a point cloud representation. Detection-based methods generate dense pixel-wise estimations with heatmaps or offset vectors from local features. V2V-PoseNet [17] uses 3D CNN network to extract a feature-based volumetric representation and estimates volumetric heatmaps. DenseReg [29] decompose 3D hand pose as 3D heatmaps and 3D joint offsets and estimates these parameters by dense pixel-wise regression. Compared with heatmap-based method with relatively high computational burden, offset-based methods achieve a better trade-off between accuracy and efficiency and can be adapted in resource-constrained platforms. A2J [30] predicts per-joint pixel-wise offset through a dense set of anchor points on the input image. JGR-P2O [5] proposes a pixel-to-offset prediction network to address the trade-off between accuracy and efficiency for hand pose estimation. HandFoldingNet [2] inputs 3D hand point cloud and acquires the hand joint locations based on point-wise regression. SRN [21] regresses the joint position through multiple stacked network modules to capture spatial information. TriHorn-Net[23] computes two complementary attention maps of each joint and uses appearance-based data augmentation to improve the accuracy of hand pose estimation.

2.2 Ordinal Regression

The ordinal regression method maps direct regression into multiple binary classifiers and learns to predict ordinal labels. By preserving the natural order and supervising with multiple rank labels, the ordinal regression methods [7, 12, 14] have been proven to achieve much higher accuracy and faster convergence than the direct regression. DORN [7] converts the depth prediction into an ordinal regression problem, which discretizes depth value into several intervals and obtains ordinal labels to improve depth estimation accuracy. Furthermore, [19] also proposes the definitions of ordinal depth, which are based on comparing the relative depth between different joints. However, [19] is different from our DOR3D-Net. In this paper, we reformulate the pose estimation problem into an ordinal regression problem and compare binary spatial relationships between a sampling position with respect to a joint point. Our proposed dense ordinal supervision guarantees the probability maps are ordinal, which reduces the depth noise effect and improves 3D joint pose estimation accuracy.

3 Methods

In this section, we first introduce the feature extractor module, which use a transformer-based feature extractor to learn dense local feature representations. Then, we elaborate on the details of dense ordinal regression module, which design to output pixel-wise probability maps and regress hand joints. Finally, we introduce the overall training procedure.

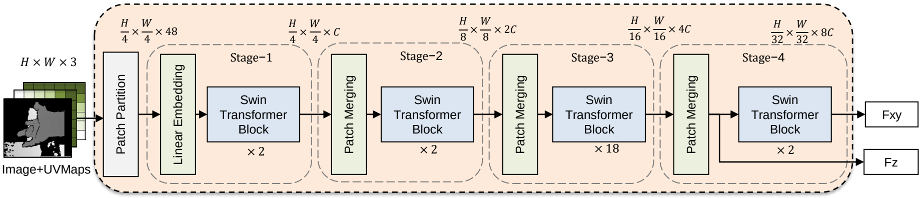

Figure 2: The pipeline of our transformer-based feature extractor. It contains patch partition and four Swin Transformer stages. Two feature maps from the last stage are sent into the dense ordinal regression module for 3D hand pose prediction.

3.1 Feature Extractor

In the transformer-based feature extractor module (Fig. 2), the input image plane is split into multiple 4×4 patches with the patch partition module. Each patch is treated as a 'token'. Four Swin Transformer stages are used to learn attention among patch tokens for capturing long-range contextual information. These stages consist of linear embedding layers, patch merging layers, and Swin Transformer blocks with their structure details specified in [16].

Since the Swin Transformer structure contains only relative positional embedding, we modify the input by adding U, V coordination maps (UVMap) and concat them together with the depth map to provide global absolute spatial information. U and V maps are generated by linear scaling, which corresponds to the in-plain coordinate map of each pixel. Here, $U(i,j) = j/W, i \in [0,H), j \in [0,W)$; $V(i,j) = i/H, i \in [0,H), j \in [0,W)$.

Considering the in-plane xy regression and depth-plane z regression are quite different, following the design of A2J [30], two feature maps F_{xy} and F_z are output from 'Stage-4' with the same dimensions $\frac{H}{32} \times \frac{W}{32} \times 8C$. Then, both decoded feature maps are regressed with the dense ordinal regression module for 3D joint prediction.

3.2 Dense Ordinal Regression Module

Fig. 3 illustrates the pipeline for the dense ordinal regression module. With input as the learned feature maps, we utilize separate branches to estimate the three-dimensional coordinates of hand joints independently and output the predicted hand joint pose.

Normal Discretization. Since the hand joints reside in three dimensional space, we decouple the 3D solution space and quantize it by representative discrete values along each of the three dimensions. In image plane x-axis and y-axis directions, the intervals are $[0,W)$ and $[0,H)$ respectively. Uniform discretization (UD) is adopted to divide the image plane. Assuming that the intervals are discretized into K_x, K_y sub-intervals along x-axis and y-axis

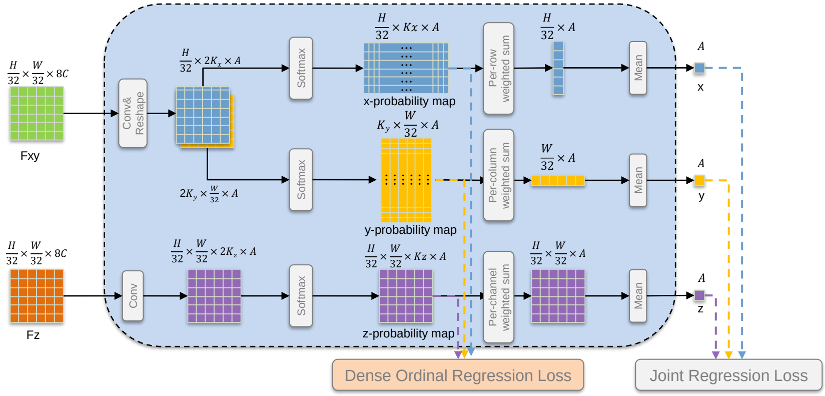
$\frac{H}{32} \times \frac{W}{32} \times 8C$
Fxy
Conv& Reshape
$\frac{H}{32} \times 2K_x \times A$
$2K_y \times \frac{W}{32} \times A$
Softmax
Softmax
$\frac{H}{32} \times Kx \times A$
x-probability map
$K_y \times \frac{W}{32} \times A$
y-probability map
Per-row weighted sum
Per-column weighted sum
$\frac{H}{32} \times A$
$\frac{W}{32} \times A$
Mean
Mean
A
x
A
y
$\frac{H}{32} \times \frac{W}{32} \times 8C$
Fz
Conv
$\frac{H}{32} \times \frac{W}{32} \times 2K_z \times A$
Softmax
$\frac{H}{32} \times \frac{W}{32} \times Kz \times A$
z-probability map
Per-channel weighted sum
$\frac{H}{32} \times \frac{W}{32} \times A$
Mean
A
z
Dense Ordinal Regression Loss
Joint Regression Loss

Figure 3: The pipeline of our proposed dense ordinal regression module. The inputs are two feature maps. With the reshape and softmax operators, we obtain binary probability maps. With weighted sum, the binary probabilities at local positions are aggregated to infer hand keypoints along each of the three dimensions respectively.

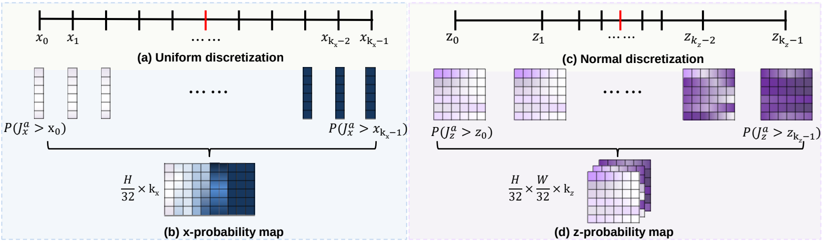

Figure 4: Visualization of the proposed x- and z-discretization process. x-axis uses uniform discretization and z-axis applies normal discretization. For the x-probability map, each column represents the probability that the keypoint is larger than the corresponding discretization threshold. For the z-probability map, each map represents the probability that the keypoint is larger than the corresponding discretization threshold.

directions respectively, the UD can be formulated as: $x_i = i * W/K_x$, $y_j = j * H/K_y$, where x_i, y_j are sampling points and then form well-ordered sets $S_x = x_0, \ldots, x_{K_x-1}$, $S_y = y_0, \ldots, y_{K_y-1}$. Here, we set $K_x = W/2$, $K_y = H/2$.

In the z-axis direction, we analyze statistics of the joints z coordinate distribution and notice that it is close to normal distribution. Following this, the sampling interval of normal discretization (ND) becomes smaller as the sampling position becomes closer to the distribution center. In our specific implementation, we first divide the z-axis $[0, D)$ into several sub-intervals evenly, and then increase the frequency of sampling points by the exponential power of 2 for the consecutive sub-intervals to the midpoint $D/2$. Sampling points for the other half of the z-interval can be obtained by symmetry. These sampling points form a well-ordered set $S_z = z_0, \ldots, z_{K_z-1}$. Here, we set $K_z = D/4$, D refers to the depth range of the cropped image. Fig. 4 (a) and (c) visualize the distributions along x and z respectively.

Ordinal Regression. After obtaining the discrete ordered classification sets S_x, S_y, S_z, we cast the hand pose estimation problem into an ordinal regress problem to learn the network. These well-ordered sampling points along x-axis, y-axis, and z-axis directions construct multiple binary classification sub-problems. For each predicted coordinate of joint a, these binary classifiers are used to predict whether the hand joint location is larger than these discretization thresholds respectively and then form probability map $Prob_x$, $Prob_y$, and $Prob_z$. Here, joint location is represented as $J^a = (J^a_x, J^a_y, J^a_z)$.

$$Prob_x(i,j,a) = P(J^a_x > x_j),\ Prob_y(i,j,a) = P(J^a_y > y_i),\ Prob_z(i,j,k,a) = P(J^a_z > z_k). \quad (1)$$

Fig. 3 illustrates the probability map generation process. Let $F_{xy} \in R^{\frac{H}{32}\times\frac{W}{32}\times 8C}$ denote the feature maps in the image plane, after convolution, reshape and softmax operators, the feature map F_{xy} is converted into the probability map $Prob_x \in R^{\frac{H}{32}\times K_x \times A}$ and $Prob_y \in R^{K_y \times \frac{W}{32}\times A}$. For the feature map $F_z \in R^{\frac{H}{32}\times\frac{W}{32}\times 8C}$, after convolution and softmax operators, it is mapped into the probability map $Prob_z \in R^{\frac{H}{32}\times\frac{W}{32}\times K_z \times A}$. A is the number of joints. To guarantee the accuracy of classification, all these probability maps are densely supervised with the ground truth (GT) probability maps and introduced in section 3.3. Obviously, the ordinal regression solution compares the binary spatial relationship between the keypoint and every discretization threshold. In comparison with dense offset regression methods, the solution space is reduced from a large interval to binary values which is easy to get the optimal solution and insensitive to the noise and outliers.

Following the ordinal regression methods [12, 18], after obtaining the reliable probability maps $Prob_x$, we weighted the probability with its corresponding classification interval length and get the hand joint prediction vector in Eq. 2. Here, IL refers to the interval length. For x and y, we set $IL = 2$. For z, the interval length changes according to the sampling position, $IL(k) = (z_{k+1} - z_k)$, z_k are sampling points. The final predicted coordinate value $\hat{x}$ is the mean value of $\bar{x}(i)$. In the same way, we obtain the coordination $\hat{y}$ and $\hat{z}$ from $\bar{y}(j)$, $\bar{z}(i,j)$ in Eq. 2, respectively.

$$\begin{aligned} \bar{x}(i,a) &= IL \cdot \textstyle\sum_j Prob_x(i,j,a), \\ \bar{y}(j,a) &= IL \cdot \textstyle\sum_i Prob_y(i,j,a), \\ \bar{z}(i,j,a) &= \textstyle\sum_k IL(k) \cdot Prob_z(i,j,k,a). \end{aligned} \quad (2)$$

We concatenate $\hat{x}$, $\hat{y}$ and $\hat{z}$ as the final results.

3.3 Loss Function

The network is jointly supervised by two loss items: dense ordinal regression loss and joint regression loss. The binary probability maps along x, y and z-axis are supervised to guarantee that the learned features are robust to the low-quality depth image.

Dense Ordinal Regression (DOR) Loss. To supervise the binary classifiers, we generate the GT of middle results (GT binary probability maps $Prob^{gt}_x$, $Prob^{gt}_y$ and $Prob^{gt}_z$):

$$Prob^{gt}_x(i,j,a) = \begin{cases} 1,\ if\ J^{a*}_x \geq x_j,\ \forall\ i \\ 0,\quad otherwise.\ \forall\ i \end{cases} \quad (3)$$

where J^{a*}_x is the GT coordinate of joint a. $Prob^{gt}_y$ and $Prob^{gt}_z$ are defined similarly to $Prob^{gt}_x$.

Take the $Prob_x \in R^{\frac{H}{32}\times W \times A}$ for an example, the DOR loss is defined as the cross-entropy loss to densely supervise all binary classification probability maps.

$$L_{ord}\left(Prob_x, Prob^{gt}_x\right) = -\frac{1}{\frac{HA}{32}} \sum \left(Prob^{gt}_x \log\left(Prob_x\right) + \left(1 - Prob^{gt}_x\right)\log\left(1 - Prob_x\right)\right). \quad (4)$$

The DOR losses in three dimensions are defined in Eq.5.

$$L_{ord\_loss} = L_{ord}\left(Prob_x, Prob^{gt}_x\right) + L_{ord}\left(Prob_y, Prob^{gt}_y\right) + L_{ord}\left(Prob_z, Prob^{gt}_z\right). \quad (5)$$

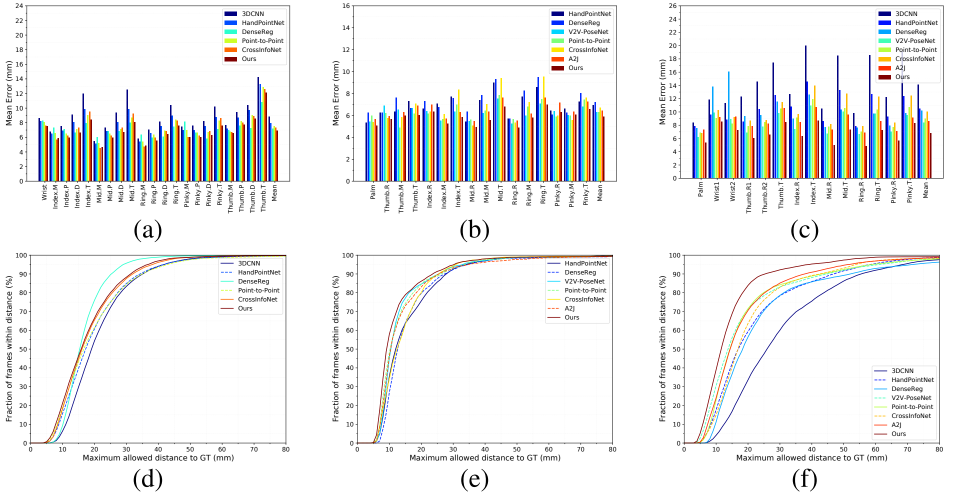

Figure 5: Comparison with the state-of-the-art methods: (a-c) 3D per-joint mean error on MSRA, ICVL and NYU; (d-e) Percentages of success frames on MSRA, ICVL and NYU.

Joint Regression Loss. The GT hand pose supervises the network to generate an accurate hand pose at the final stage. The loss is formulated as $L_{joint\_loss} = \sum_{a \in A} L_{smooth}\left(J^a - J^{a*}\right)$, where J^a and J^{a*} are the predicted coordinate and the GT of joint a, respectively.

Total Loss Function. The total loss is defined as: $L = \lambda_1 L_{joint\_loss} + \lambda_2 L_{ord\_loss}$, where λ_1 and λ_2 are hyperparameters for balancing these terms. Here, λ_1 = 3 and λ_2 = 2.

4 Experiments

4.1 Experimental Setting

Datasets. There are four widely adopted datasets for 3D hand pose estimation task, including HANDS2017 [31], MSRA [26], ICVL [27], and NYU [28]. HANDS2017 dataset is composed of Big Hand 2.2M dataset [31] and the First-person Hand Action Dataset (FHAD) [9]. It contains 957K training and 295K testing depth images. MSRA dataset contains 9 subjects with each subject performs 17 hand gestures, and each hand gesture contains about 500 frames. ICVL dataset contains 22K training and 1.5k testing depth images with 3D annotations for 16 joints. The raw images with annotations are augmented to 330K samples by in-plane rotations. NYU dataset contains 72K training and 8.2K testing depth images labeled with 3D annotations for 36 joints. Following A2J [30], we only use 14 of the 36 joints from frontal view for both training and testing. Since the number of raw images in HANDS2017 dataset is over ten times bigger than any of the rest three datasets, we conduct ablation study on the HANDS2017 dataset.

Implementation Details. DOR3D-Net is trained with 2 NVIDIA V100 GPUs. We adopt the AdamW optimizer and the learning rate is 3.5×10^{-4} with a weight decay of 10^{-4}. The batch size for MSRA, ICVL and NYU datasets is 32, and the batch size for HANDS2017 is 64. For all datasets, the learning rate decays by 0.2 every 7 epoches. Similar to A2J [30] and DenseRecurrent [3], we use hand center point to crop the hand region from an depth image and resize the image to 224 × 224.

Evaluation Metric. We use two standard metrics to evaluate pose estimation performance. One is the mean 3D Euclidean distance error (Mean Error) [30]. Another is the

Table 1: Comparison with the state-of-the-art methods.

Method	**Mean Error [mm] ↓**			
	MSRA	**NYU**	**ICVL**	**HANDS17**
HandPointNet [10]	8.51	10.54	6.94	-
DenseReg [29]	7.23	10.21	7.30	-
V2V-PoseNet [17]	7.59	8.42	6.28	9.95
Point-to-Point [11]	7.71	8.99	-	9.82
CrossInfoNet [4]	7.86	10.08	6.73	
A2J [30]	-	8.61	6.46	8.57
JGR-P2O [5]	7.55	8.29	6.02	-
HandFoldingNet [2]	7.34	8.58	5.95	-
SRN [21]	7.16	7.78	6.26	8.39
DenseRecurrent [3]	7.01	6.85	6.05	-
TriHorn-net [23]	7.13	7.68	**5.73**	-
DOR3D-Net (Ours)	**6.93**	**6.71**	5.87	**6.99**

Table 2: Comparison of heatmap-based methods on NYU dataset [28].

Method	**Backbone**	**Mean Error [mm]↓**
DOR3D-Net (w/ Heatmap Regression)	Resnet50	8.63
DOR3D-Net (w/ Ordinal Regression)	**Resnet50**	**8.25**

percentage of success frames in which the worst joint 3D distance error is below a threshold [30]. Note that, the results of our method are all predicted by a single model.

4.2 Comparison with the State-of-the-art Methods

We compare our method with the state-of-the-art depth image-based hand pose estimation methods, i.e., HandPointNet [10], DenseReg [29], V2V-PoseNet [17], Point-to-Point [11], CrossInfoNet [4], A2J [30], JGR-P2O [5], HandFoldingNet [2], SRN [21], DenseRecurrent [3], and TriHorn-net [23]. Fig. 5 shows the result of 3D per-joint mean error and the percentages of success frames over different error thresholds. Meanwhile, Tab. 1 shows the overall performance of DOR3D-Net and all the methods on four datasets, respectively. It can be seen that ours outperforms all the other methods on the MSRA, NYU and HANDS17 datasets. On the ICVL dataset, our method is ranked second, with a slightly lower accuracy than the TriHorn-net [23], which uses an innovative data agumentation approach.

Moreover, to further valid the effectiveness of the dense ordinal regression module, we compare it with the heatmap-based method [25] with the same backbone Resnet50 on NYU dataset [28]. Tab. 2 shows that our ordinal regression module surpasses the heatmap-based method. In a nutshell, DOR3D-Net shows significant superiority over other existing methods, indicating the benefit of the dense ordinal regression module.

4.3 Ablation Study

To demonstrate the effectiveness of each module in our method, we conduct extensive ablation studies on the HANDS2017 dataset. Tab. 3, Tab. 4, Tab. 5 and Tab. 6 show the experimental results in detail.

Effectiveness of dense ordinal regression module. To verify the effectiveness of the dense ordinal regression module, we replace the regression module with an offset-based regression module and show the results in the Tab. 3. Specifically, we design the following model variants: (1) DOR3D-Net (w/ offset-based regression) : we train the model with the offset-based regression module; (2) DOR3D-Net (w/ ordinal-based regression) : we train the

Table 3: Effectiveness of dense ordinal regression module.

Method	Mean Error [mm] ↓			
	x	y	z	all
DOR3D-Net (w/ offset-based regression)	3.31	3.33	4.32	7.34
DOR3D-Net (w/ ordinal-based regression)	**3.12**	**3.19**	**4.13**	**6.99**

Table 4: Effectiveness of normal discretization.

Method	Mean Error [mm] ↓			
	x	y	z	all
DOR3D-Net (w/ uniform distribution)	3.14	3.19	4.18	7.06
DOR3D-Net (w/ normal distribution)	**3.12**	**3.19**	**4.13**	**6.99**

model with the dense ordinal regression module; For a fair comparison, we uses the same backbone and experimental configuration. As shown in Tab. 3, our ordinal-based regression method significantly outperforms the offset-based regression method, and the 3D mean error is reduced by 4.77% on HANDS2017 dataset. There are two main reasons: (1) Offset-based regression module regresses the hand joints in a large 3D solution space, which is hard to obtain the optimal solution. (2) The dense ordinal regression module predicts probability maps that vary smoothly with ordinal constraints and are insensitive to noise and outliers.

Effectiveness of normal discretization. To verify the effectiveness of normal discretizatione, we verified two discretization strategies to quantize the z-axis interval and show the results in Tab. 4. Specifically, DOR3D-Net (w/ uniform distribution) uses uniform distribution strategy. Similarly, DOR3D-Net (w/ normal distribution) uses normal distribution strategy. The results of DOR3D-Net (w/ uniform distribution) is worse than DOR3D-Net (w/ normal distribution), demonstrating verifying the effectiveness of normal distribution strategy. Moreover, we analyze statistics of the hand z coordinate distribution in public four datasets and notice that it is close to normal distribution (ND).

Effectiveness of feature extractor module. In Sec. III-A, we proposes transformer-based feature extractor with three designs: (1) UVMap: since the transformer structure contains only relative positional embedding, we design to include UVMap in the input to provide global absolute spatial information; (2) Introduce transformer structure: it has the powerful capability to learn the long-range relationship of dense features; (3) Output feature design: considering the in-plane xy regression and depth-plane z regression are quite different, we output two features F_{xy} and F_z from transformer to regress the xy and z coordinates, respectively. We conduct an ablation study on the components in feature extractor and summarize our results in Tab. 5. The ablation study show that each of designs in transformer-based

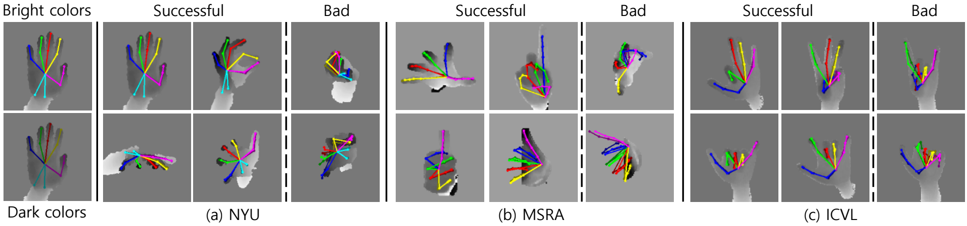

Figure 6: Joints prediction on NYU, MSRA, and ICVL. Bright colors represent predicted results and dark colors show ground truth. The visualization results superimpose the predicted results on the ground truth. Both successful and bad cases are displayed.

Table 5: Effectiveness of feature extractor module.

Method	Module	Mean Error[mm]↓	Params[MB]↓	Speed[FPS]↑
Input	DOR3D-Net (w/o UVMap)	7.10	86.9	**50**
	DOR3D-Net(w/ UVMap)	**6.99**	**86.9**	47
Backbone	DOR3D-Net (Resnet50-based)	7.67	**32.7**	**110**
	DOR3D-Net (Transformer-based)	**6.99**	86.9	47
Design	DOR3D-Net (w/ F_{xyz})	7.04	86.9	47
	DOR3D-Net (w/ F_{xy}&F_z)	**6.99**	**86.9**	**47**

Table 6: Effectiveness of DOR Loss. 'DOR loss' refers to the dense ordinal regression loss.

Method	Module	Mean Error[mm]↓	Params[MB]↓	Speed[FPS]↑
Loss	DOR3D-Net (w/o DOR loss)	7.31	86.9	47
	DOR3D-Net (w/ DOR loss)	**6.99**	**86.9**	**47**

feature extractor provides a meaningful contribution to improving model performance.

Effectiveness of DOR Loss. In this ablation, we investigate the effectiveness of dense ordinal regression loss (DOR loss) in our method. We design the following model variants: (1) DOR3D-Net (w/o DOR loss) : we train the model without the dense ordinal regression loss; (2) DOR3D-Net (w/o DOR loss) : we train the model with the dense ordinal regression loss; As can be seen from the Tab. 6, without DOR loss, the error increased by 0.32mm. This also validates that dense probability supervision plays an important role in our DOR3D-Net to learn representative features and improve hand pose accuracy.

Speed and Parameters. We test our model on a single NVIDIA V100 GPU. The speed of the Resnet50-based backbone and transformer-based backbone is 110 FPS and 47 FPS, respectively. The full model meets the real-time requirement for practical applications. The parameters of Resnet50-based backbone and Transformer-based backbone are 32.7MB and 86.9MB, respectively.

Qualitative Results. Fig. 6 visualizes the successful and failure cases of our full model performance on MSRA dataset [26], NYU dataset [28] and ICVL dataset [27]. It can be seen that our method predicts well in most cases, while fails in cases of severe occlusion, large areas of missing pixels, and challenging viewpoint.

5 Conclusion

In this paper, we propose a DOR3D network that reformulates the 3D hand pose estimation as a dense ordinal regression problem. In comparison with offset-based regression methods, this formulation simplifies the solution space from a large interval to binary values which enables the network to learn easily and find out the optimal solution. Furthermore, a transformer-based feature extractor is utilized to enhance dense feature presentation and additional UV coordination maps are generated to provide absolute spatial information. Our DOR3D-net has achieved the SOTA performance on the HANDS2017, MSRA, NYU and ICVL datasets. This method provides high accuracy pose estimation which could be useful in the augmented reality/virtual reality, autonomous driving and robotics. In the future, we will improve the depth-based hand pose estimation by improving the samples with the challenging viewpoint. Moreover, we will to improve the model efficiency for porting to mobile platforms for low-latency user interaction.

References

[1] Haoming Chen, Runyang Feng, Sifan Wu, Hao Xu, Fengcheng Zhou, and Zhenguang Liu. 2d human pose estimation: A survey. *Multimedia Systems*, 29(5):3115–3138, 2023.

[2] Wencan Cheng, Jae Hyun Park, and Jong Hwan Ko. Handfoldingnet: A 3d hand pose estimation network using multiscale-feature guided folding of a 2d hand skeleton. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11260–11269, 2021.

[3] Xiaoming Deng, Dexin Zuo, Yinda Zhang, Zhaopeng Cui, Jian Cheng, Ping Tan, Liang Chang, Marc Pollefeys, Sean Fanello, and Hongan Wang. Recurrent 3d hand pose estimation using cascaded pose-guided 3d alignments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):932–945, 2022.

[4] Kuo Du, Xiangbo Lin, Yi Sun, and Xiaohong Ma. Crossinfonet: Multi-task information sharing based hand pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9896–9905, 2019.

[5] Linpu Fang, Xingyan Liu, Li Liu, Hang Xu, and Wenxiong Kang. Jgr-p2o: Joint graph reasoning based pixel-to-offset prediction network for 3d hand pose estimation from a single depth image. In *European Conference on Computer Vision*, pages 120–137. Springer, 2020.

[6] Seyed Amirhossein Farjadi, M-R Akbarzadeh-T, and Kamaledin Ghiasi-Shirazi. Rgb image-based hand pose estimation: A survey on deep learning perspective. In *2024 20th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP)*, pages 1–7. IEEE, 2024.

[7] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2002–2011, 2018.

[8] Qing Gao, Zhaojie Ju, Yongquan Chen, Qiwen Wang, Yinan Zhao, and Shiwu Lai. Parallel dual-hand detection by using hand and body features for robot teleoperation. *IEEE Transactions on Human-Machine Systems*, 53(2):417–426, 2023.

[9] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 409–419, 2018.

[10] Liuhao Ge, Yujun Cai, Junwu Weng, and Junsong Yuan. Hand pointnet: 3d hand pose estimation using point sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8417–8426, 2018.

[11] Liuhao Ge, Zhou Ren, and Junsong Yuan. Point-to-point regression pointnet for 3d hand pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 475–491, 2018.

[12] Tianchu Guo, Hui Zhang, ByungIn Yoo, Yongchao Liu, Youngjun Kwak, and Jae-Joon Han. Order regularization on ordinal loss for head pose, age and gaze estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1496–1504, 2021.

[13] Xiaoshuai Hao, Yi Zhu, Srikar Appalaraju, Aston Zhang, Wanqian Zhang, Bo Li, and Mu Li. Mixgen: A new multi-modal data augmentation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 379–389, 2023.

[14] Wanhua Li, Xiaoke Huang, Jiwen Lu, Jianjiang Feng, and Jie Zhou. Learning probabilistic ordinal embeddings for uncertainty-aware regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13896–13905, 2021.

[15] Xingyu Liu, Pengfei Ren, Yuanyuan Gao, Jingyu Wang, Haifeng Sun, Qi Qi, Zirui Zhuang, and Jianxin Liao. Keypoint fusion for rgb-d based 3d hand pose estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 3756–3764, 2024.

[16] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[17] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *Proceedings of the IEEE conference on computer vision and pattern Recognition*, pages 5079–5088, 2018.

[18] Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. Ordinal regression with multiple output cnn for age estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4920–4928, 2016.

[19] Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal depth supervision for 3d human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7307–7316, 2018.

[20] Jing Qi, Li Ma, Zhenchao Cui, and Yushu Yu. Computer vision-based hand gesture recognition for human-robot interaction: a review. *Complex & Intelligent Systems*, 10 (1):1581–1606, 2024.

[21] Pengfei Ren, Haifeng Sun, Qi Qi, Jingyu Wang, and Weiting Huang. Srn: Stacked regression network for real-time 3d hand pose estimation. In *BMVC*, page 112, 2019.

[22] Pengfei Ren, Haifeng Sun, Jiachang Hao, Jingyu Wang, Qi Qi, and Jianxin Liao. Mining multi-view information: A strong self-supervised framework for depth-based 3d hand pose and mesh estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20555–20565, 2022.

[23] Mohammad Rezaei, Razieh Rastgoo, and Vassilis Athitsos. Trihorn-net: a model for accurate depth-based 3d hand pose estimation. *Expert Systems with Applications*, 223: 119922, 2023.

[24] Wenkang Shan, Zhenhua Liu, Xinfeng Zhang, Zhao Wang, Kai Han, Shanshe Wang, Siwei Ma, and Wen Gao. Diffusion-based 3d human pose estimation with multi-hypothesis aggregation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14761–14771, 2023.

[25] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5693–5703, 2019.

[26] Xiao Sun, Yichen Wei, Shuang Liang, Xiaoou Tang, and Jian Sun. Cascaded hand pose regression. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 824–832, 2015.

[27] Danhang Tang, Hyung Jin Chang, Alykhan Tejani, and Tae-Kyun Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3786–3793, 2014.

[28] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (ToG)*, 33(5):1–10, 2014.

[29] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Dense 3d regression for hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5147–5156, 2018.

[30] Fu Xiong, Boshen Zhang, Yang Xiao, Zhiguo Cao, Taidong Yu, Joey Tianyi Zhou, and Junsong Yuan. A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 793–802, 2019.

[31] Shanxin Yuan, Qi Ye, Bjorn Stenger, Siddhant Jain, and Tae-Kyun Kim. Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4866–4874, 2017.

[32] Andrei Zanfir, Mihai Zanfir, Alex Gorban, Jingwei Ji, Yin Zhou, Dragomir Anguelov, and Cristian Sminchisescu. Hum3dil: Semi-supervised multi-modal 3d humanpose estimation for autonomous driving. In *Conference on Robot Learning*, pages 1114–1124. PMLR, 2023.

[33] Ce Zheng, Wenhan Wu, Chen Chen, Taojiannan Yang, Sijie Zhu, Ju Shen, Nasser Kehtarnavaz, and Mubarak Shah. Deep learning-based human pose estimation: A survey. *ACM Computing Surveys*, 56(1):1–37, 2023.

[34] Feng Zhou, Pei Shen, Ju Dai, Na Jiang, Yong Hu, Yu-Kun Lai, and Paul L Rosin. Ahrnet: Attention and heatmap-based regressor for hand pose estimation and mesh recovery. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3000–3004. IEEE, 2024.