# JGR-P2O: Joint Graph Reasoning based Pixel-to-Offset Prediction Network for 3D Hand Pose Estimation from a Single Depth Image

Anonymous ECCV submission

Paper ID 4860

**Abstract.** State-of-the-art single depth image-based 3D hand pose estimation methods are based on dense predictions, including voxel-to-voxel predictions, point-to-point regression, and pixel-wise estimations. Despite the good performance, those methods have a few issues in nature, such as the poor trade-off between accuracy and efficiency, and plain feature representation learning with local convolutions. In this paper, a novel pixel-wise prediction-based method is proposed to address the above issues. The key ideas are two-fold: a) explicitly modeling the dependencies among joints and the relations between the pixels and the joints for better local feature representation learning; b) unifying the dense pixel-wise offset predictions and direct joint regression for end-to-end training. Specifically, we first propose a graph convolutional network (GCN) based joint graph reasoning module to model the complex dependencies among joints and augment the representation capability of each pixel. Then we densely estimate all pixels' offsets to joints in both image plane and depth space and calculate the joints' positions by a weighted average over all pixels' predictions, totally discarding the complex post-processing operations. The proposed model is implemented with an efficient 2D fully convolutional network (FCN) backbone and has only about 1.4M parameters. Extensive experiments on multiple 3D hand pose estimation benchmarks demonstrate that the proposed method achieves new state-of-the-art accuracy while running very efficiently with around a speed of 110fps on a single NVIDIA 1080Ti GPU.

**Keywords:** 3D hand pose estimation, depth image, graph neural network

## 1 Introduction

Vision-based 3D hand pose estimation aims to locate hand joints in 3D space from input hand images, which serves as one of the core techniques in contactless human computer interaction applications, such as virtual reality, augmented reality and robotic gripping [23, 11]. Recent years have witnessed significant advances [6, 22, 48, 20, 8] in this area with the availability of consumer depth cameras, such as Microsoft Kinect and Intel RealSense, and the success of deep learning technology in the computer vision community. However, accurate and

real-time 3D hand pose estimation is still a challenging task due to the high articulation complexity of the hand, severe self-occlusion between different fingers, poor quality of depth images, etc.

In this paper, we focus on the problem of 3D hand pose estimation from a single depth image. At present, the state-of-the-art approaches to this task rely on deep learning technology, especially deep convolutional neural networks (CNNs). The main reasons are two-fold. On one hand, public available large datasets [49, 37, 39, 34] with fully labeled 3D hand poses provide a large number of training data for these data-hungry methods. On the other hand, CNNs with well-designed network structures provide very effective solutions to challenging visual learning tasks and have been demonstrated to outperform traditional methods by a large margin in various computer vision tasks, including 3D hand pose estimation.

Best performing deep learning-based methods are detection-based, which formulate 3D hand pose parameters as volumetric heat-maps or extended 3D heat-maps together with offset vector fields and estimate them in a dense prediction manner with fully convolutional networks (FCNs) or PointNet [30, 31]. Contrary to their regression-based counterparts that directly map the depth images to 3D hand pose parameters and severely suffer from the problem of highly non-linear mapping, the detection-based methods can learn better feature representations by pose reparameterization and have proven to be more effective for both human pose estimation [29, 4] and hand pose estimation [10, 22, 43].

By analyzing previous detection-based methods, we find that they suffer from several drawbacks in nature, which can be improved to boost performance. First, they bear the problem of poor trade-off between accuracy and efficiency. For example, the V2V [22] uses 3D CNNs to estimate volumetric heat-maps, which is very parameter-heavy and computationally inefficient. The pixel-wise and point-wise prediction-based methods [43, 10] take the advantages of 2D CNNs or PointNet to regress dense 3D estimations. In spite of the higher efficiency, these methods achieve lower estimation precision empirically, and the complex post-processing operations still degrade the computational efficiency. Second, they consist of non-differentiable post-processing operations, such as taking maximum and taking neighboring points, preventing fully end-to-end training and causing inevitable quantization errors. In addition, the models are trained with non-adaptive Gaussian heat-maps or joint-centered heat-maps, which may be suboptimal. Finally, the feature representation for each element (e.g., a voxel, a pixel or a point) is only learned by local convolutions ignoring the global context information. However, modeling the dependencies among joints and the relations between the elements and the joints helps to learn more abundant contextual information and better local feature representations.

To cope with these problems, we propose a novel joint graph reasoning based pixel-to-offset prediction network (JGR-P2O) for 3D hand pose estimation, which aims at directly regressing joints' positions from single depth images. Specifically, we decompose the 3D hand pose into joints' 2D image plane coordinates and depth values, and estimate these parameters in an ensemble way, fully

exploiting the 2.5D property of depth images. The proposed method consists of two key modules, i.e., GCN-based joint graph reasoning module and pixel-to-offset prediction module. The joint graph reasoning module aims at learning a better feature representation for each pixel, which is vital for dense prediction. First, the features of joints are generated by summarizing the global information encoded in local features. Second, the dependencies among joints are modeled by graph reasoning to obtain stronger feature representations of joints. Finally, the evolved joints' features are mapped back to local features accordingly enhancing the local feature representations. The pixel-to-offset prediction module densely estimates all the pixels' offsets to joints in both image plane and depth space. And the joints' positions in both image plane space and depth space are calculated by a weighted average over all the pixels' predictions. In this way, we discard the complex post-processing operations used in [43, 10], which improves not only the computational efficiency but also the estimation robustness.

Note that our JGR-P2O can obtain joints' positions directly from single depth images without extra post-processing operations. It generates intermediate dense offset vector fields, and can also be fully end-to-end trained under the direct supervision of joints' positions, fully sharing the merits of both detection-based and regression-based methods. It also explicitly models the dependencies among joints and the relations between the pixels and the joints to augment the local feature representations. The whole model is implemented with an efficient 2D FCN backbone and has only about 1.4M parameters. It generally outperforms the previous detection-based methods on effectiveness and efficiency simultaneously. Overall, the proposed method provides some effective solutions to the problems encountered by previous detection-based methods.

To sum up, the main contributions of this paper are as follows:

- We formulate the problem of 3D hand pose estimation from a single depth image as dense pixel-to-offset predictions leveraging the 2.5D property of depth images and unifying the dense pixel-wise offset predictions and direct joint regression for end-to-end training.
- We propose a GCN-based joint graph reasoning module to explicitly model the dependencies among joints and the relations between the pixels and the joints to augment the local feature representations that are vital for dense predictions.
- We conduct extensive experiments on multiple most common 3D hand pose estimation benchmarks (i.e., ICVL [37], NYU [39], and MSRA [34]). The results demonstrate that the proposed method achieves new state-of-the-art accuracy with only about 1.4M parameters while running very efficiently with around a speed of 110fps on single NVIDIA 1080Ti GPU.

## 2 Related Work

This paper focuses on the problem of estimating 3D hand pose from a single depth image. The approaches to this problem can be categorized into discriminative methods [44, 34, 7, 9], generative methods [40, 14, 32] and hybrid meth-

ods [38, 27, 50, 47, 42, 25, 28]. In this section, we focus on the discussions of the deep learning-based discriminative and hybrid methods related closely to our work. These methods can be further classified into regression-based methods, detection-based methods, hierarchical and structured methods. Please refer to [48, 35, 36] for more detailed review. Furthermore, we also introduce some GCN-based works that related to our method.

**Regression-based Methods.** Regression-based methods [26, 25, 7, 9, 3, 12] aim at directly regressing 3D hand pose parameters such as 3D coordinates or joint angles. Oberweger et al. [25, 26] exploit a bottleneck layer to learn a pose prior for constraining the hand pose. Guo et al. [12] propose a tree-structured Region Ensemble Network (REN) to regressing joints' 3D coordinates directly. Instead of using depth images as inputs, other works focus on 3D input representations, fully utilizing the depth information. Ge et al. [9, 7] apply 3D CNNs and PointNet [30, 31] for estimating 3D hand joint positions directly, which use 3D volumetric representation and 3D point cloud as inputs respectively. Despite the simplicity, the global regression manner within the fully-connected layers incurs highly non-linear mapping, which may reduce the estimation performance. However, our method adopts the dense prediction manner to regress the offsets from pixels to joints, effectively maintaining the local spatial context information.

**Detection-based Methods.** Detection-based methods [8, 22, 10, 43] work in dense local prediction manner via setting a heat map for each joint. Early works [39, 8] firstly detect the joints' positions in 2D plain based on the estimated 2D heat-maps and then translate them into 3D coordinates by complex optimization-based post-processing. Recent works [22, 10, 43] directly detect 3D joint positions from 3D heat-maps with much more simple post-processing. Moon et al. [22] propose a Voxel-to-Voxel prediction network (V2V) for both 3D hand and human pose estimation. Wan et al. [43] and Ge et al. [10] formulate 3D hand pose as 3D heat-maps and unit vector fields and estimate these parameters by dense pixel-wise and point-wise regression respectively. Despite the good performance, these methods have some drawbacks, such as the poor trade-off between accuracy and efficiency and local feature representations. With the proposed pixel-to-offset prediction module and GCN-based joint graph reasoning module, our method can effectively solve these problems.

**Hierarchical and Structured Methods.** These methods aim at incorporating hand part correlations or pose constraints into the model. Hierarchical methods [20, 2, 6] divide the hand joints into different subsets and use different network branches to extract local pose features for each subset. Then all the local pose features are combined together forming the global hand pose representation for final pose estimation. Structured methods [25, 26, 20, 50] impose physical hand motion constraints into the model, which are implemented by embedding constraint layers in the CNN model [25, 26, 50] or adding specific items in the loss function [50]. Different from these methods, the proposed GCN-based joint graph reasoning module aims at augmenting the local feature representation by learning the dependencies among joints and the relations between the pixels and the joints.

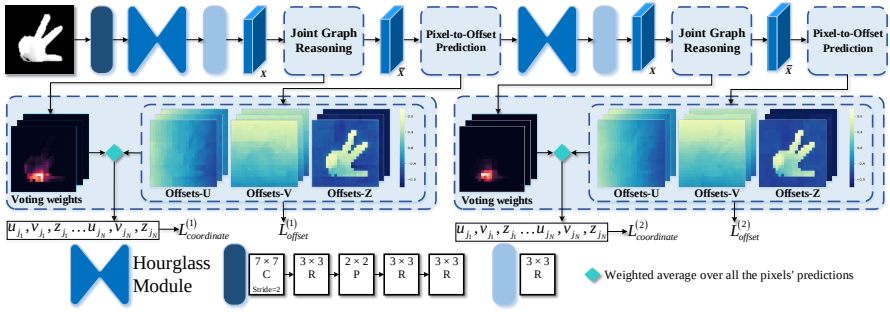


**Fig. 1.** An overview of our JGR-P2O. The abbreviations C, P, R indicate convolutional layer with BN and ReLU, pooling layer and residual module respectively. Given a hand depth image, the backbone module first extracts the intermediate local feature representation $\boldsymbol{X}$, which is then augmented by the proposed GCN-based joint graph reasoning module producing the augmented local feature representation $\bar{\boldsymbol{X}}$. Finally, the proposed pixel-to-offset prediction module predicts three offset maps for each joint where each pixel value indicates the offset from the pixel to the joint along one of the axes in the UVZ coordinate system. The joint's UVZ coordinates are calculated as the weighted average over all the pixels' predictions. Two kinds of losses, coordinate-wise regression loss $L_{coordinate}$ and pixel-wise offset regression loss $L_{offset}$, are proposed to guide the learning process. We stack together two hourglasses to enhance the learning power, feeding the output from the previous module as the input into the next while exerting intermediate supervision at the end of each module.

**Related GCN-based Works.** Graph CNNs (GCNs) generalize CNNs to graph-structured data. Approaches in this field are often classified into two categories: spectral based methods [5, 15] that start with constructing the frequency filtering, and spatial based methods [21, 41] that generalize the convolution to a patch operator on groups of node neighbors. Recently, some works use GCNs for 3D pose estimation [1] and skeleton-based action regnition [46, 16, 33]. And some works [19, 18] use GCN-based methods to augment the local feature representation for dense prediction. Inspired by these works, we also define the connections between hand joints as a graph and apply a GCN to learn their dependencies. Moreover, we design several different joint graph structures for comprehensive comparison studies.

# 3 The Proposed Method

## 3.1 Overview

The proposed JGR-P2O casts the problem of 3D hand pose estimation as dense pixel-to-offset predictions, fully exploiting the 2.5D property of depth images. It takes a depth image as input and outputs the joints' positions in image plain (i.e., uv coordinates) and depth space (i.e., z coordinates) directly. An overview of the JGR-P2O can be found in Figure 1. We use the high-efficient hourglass

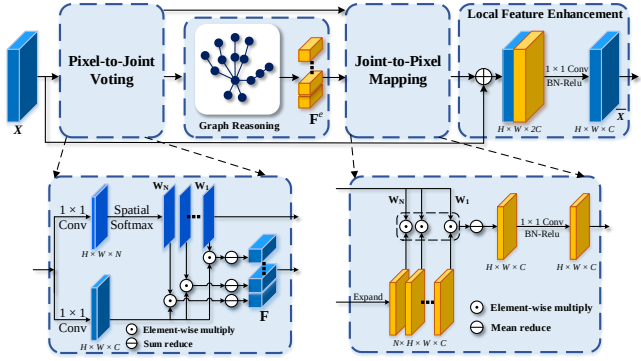


**Fig. 2.** Flowchart of our proposed GCN-based joint graph reasoning module. Given the intermediate feature representation $\boldsymbol{X}$ extracted from the backbone module, it first generates the joints' feature representation $\mathbf{F}$ by a pixel-to-joint voting mechanism where each joint is represented as the weighted average over all the local features. Then we define the connections between joints as a graph and map the joints' features to the corresponding graph nodes. The joints' features are propagated within the graph by graph reasoning, obtaining the enhanced joints' feature $\mathbf{F}^e$. Next, the $\mathbf{F}^e$ is mapped back to local features by a joint-to-pixel mapping mechanism that is the inverse operation of the pixel-to-joint voting, generating the joint context representations for all pixels. Finally, the original feature representation and the joint context representation are fused together obtaining enhanced local feature representation.

network [24] as the backbone to extract intermediate local feature representation. Then the proposed joint graph reasoning module models the dependencies among joints and the relations between the pixels and the joints enhancing the intermediate local feature representation. Finally, the pixel-to-offset module estimates the offsets from pixels to joints and aggregates all the pixels' predictions to obtain final joints' positions. Following [24], we stack together two hourglasses to enhance the learning power, feeding the output from the previous module as the input into the next while exerting intermediate supervision at the end of each module. Next, we will first introduce the GCN-based joint graph reasoning module and the pixel-to-offset prediction module in detail. And then introduce the the training strategy of the whole network architecture.

### 3.2 GCN-based Joint Graph Reasoning Module

This module aims at augmenting the intermediate local feature representation for each pixel, which is vital for local prediction. Given the extracted feature map from the backbone, we first generate the joints' features by summarizing the global context information encoded in local features. Specifically, joints are represented as the weighted average over all the local features through a pixel-to-joint voting mechanism. Then a joint-to-joint undirected graph $\boldsymbol{G} =< \mathcal{N}, \mathcal{E} >$ is defined, where each node in $\mathcal{N}$ corresponds to a joint and each edge $e_{i,j} \in \mathcal{E}$

encodes relationship between two joints. And the joints' features are propagated with the defined structure of $\boldsymbol{G}$ to capture the dependencies among joints and enhance their representation capabilities further. Finally, the evolved joints' features are mapped back to local features through a joint-to-pixel mapping mechanism, obtaining the pixel-wise joint context representations which are combined with original local features to enhance the local feature representations. The detailed pipeline of this module can be found in Figure 2.

**Pixel-to-Joint Voting** We seek to obtain joints' visual representations based on the global context information encoded in local features. Specifically, each joint has its informative pixels, the representations of which are aggregated to form the joint's feature. In this paper, we compute the joints' features by a pixel-to-joint voting mechanism. Given the feature map $\boldsymbol{X} \in \mathbb{R}^{H \times W \times C}$ after the backbone network, where $H$, $W$ and $C$ denote the height, width and number of channels of the feature map respectively. First, the voting weights from pixels to joints are computed as:

$$\boldsymbol{W} = \boldsymbol{\Phi}\left(\phi\left(\boldsymbol{X}\right)\right), \tag{1}$$

where $\phi\left(\cdot\right)$ is a transformation function implemented by a $1 \times 1$ convolution, $\boldsymbol{\Phi}$ is the spatial softmax normalization, and $\boldsymbol{W} \in \mathbb{R}^{H \times W \times N}$ is the voting tensor where the $kth$ channel $\mathbf{W}_k \in \mathbb{R}^{H \times W}$ represents the voting matrix for joint $k$. Then the feature representation for joint $k$ is calculated as the weighted average over all the transformed pixel-wise representations:

$$\mathbf{f}_k = \sum_i w_{ki} \varphi\left(\mathbf{x}_i\right), \tag{2}$$

where $\mathbf{x}_i$ is the representation of pixel $p_i$, $\varphi\left(\cdot\right)$ is a transformation function implemented by a $1 \times 1$ convolution layer, and $w_{ki}$, an element of $\mathbf{W}_k$, is the voting weight for pixel $p_i$. We also define the whole representation of all $N$ joints as $\mathbf{F} = \left[\mathbf{f}_1^T; \ldots; \mathbf{f}_N^T\right]$.

**Graph Reasoning** Given the joints' features and the defined joint-to-joint undirected graph $\boldsymbol{G}$, it is natural to use a GCN to model the dependencies among joints and augment the joints' feature representations further. Following GCN defined in [15], we perform graph reasoning over representation $\mathbf{F}$ of all joints with matrix multiplication, resulting the evolved joint features $\mathbf{F}^e$:

$$\mathbf{F}^e = \sigma\left(\mathbf{A}^e \mathbf{F} \mathbf{W}^e\right), \tag{3}$$

where $\mathbf{W}^e \in \mathbb{R}^{C \times C}$ is a trainable transformation matrix, $\mathbf{A}^e \in \mathbb{R}^{N \times N}$ is the connection weight matrix defined according to the edge connections in $\mathcal{E}$, and $\sigma\left(\cdot\right)$ is a nonlinear function (we use ReLU function for $\sigma\left(\cdot\right)$ in this paper). To demonstrate the generalization capability of the GCN-based joint graph reasoning module, we try three different methods to construct graph structure (i.e., the definition of $\mathbf{A}^e$) in this paper.

**Skeleton Graph.** The most intuitive method is to define the edge connections as hard weights (i.e., $\{0, 1\}$) based on the physical connections between joints in the hand skeleton. Then the connection weight matrix is defined as the normalized form as in [15]: $\mathbf{A}^e = \tilde{\mathbf{D}}^{-\frac{1}{2}} (\mathbf{A} + \mathbf{I}_N) \tilde{\mathbf{D}}^{-\frac{1}{2}}$, where $\mathbf{A}$ is the adjacency matrix defined in the hand skeleton, $\mathbf{I}_N$ is the identity matrix, and $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$ defines a undirected graph with added self-connections. $\tilde{\mathbf{D}}$ is the diagonal node degree matrix of $\tilde{\mathbf{A}}$ with $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$.

**Feature Similarity.** The connection weight between two joints can be calculated as the similarity of their visual representations: $a^e_{ij} = \frac{exp\big(\upsilon(\mathbf{f}_i)^T \psi(\mathbf{f}_j)\big)}{\sum_{j=1}^{N} exp\big(\upsilon(\mathbf{f}_i)^T \psi(\mathbf{f}_j)\big)}$, where $\upsilon$ and $\psi$ are two liner transformation functions implemented by two fully-connected layers. Note that each sample has a unique graph learned by this data-dependent method.

**Parameterized Matrix.** In this way, $\mathbf{A}^e$ is defined as a parameterized matrix whose elements are optimized together with the other parameters in the training process, that is, the graph is completely learned according to the training data.

**Joint-to-Pixel mapping** The evolved joint features can be used to augment the local feature representations. Specifically, the pixel-wise joint context representations are first calculated by mapping the evolved joint features back to local features and then combined with original pixel-wise representations to compute the augmented local feature representations. We use the inverse operation of pixel-to-joint voting, i.e., joint-to-pixel mapping, to calculate the pixel-wise joint context representation. For pixel $p_i$, we first compute its context representation of joint $k$ as: $\mathbf{c}_{ik} = w_{ik}\mathbf{f}^e_k$, where $\mathbf{f}^e_k$ is the evolved feature of joint $k$, and $w_{ik}$ is the mapping weight from joint $k$ to pixel $p_i$, which is the same as the voting weight $w_{ki}$ in formula (2). Then the mean of set $\{\mathbf{c}_{ik}; k = 1, \dots N\}$ is used to calculate the final pixel-wise joint context representation for pixel $p_i$:

$$\mathbf{c}_i = \rho\left(\frac{1}{N}\sum_k \mathbf{c}_{ik}\right), \tag{4}$$

where $\rho$ is a transformation function implemented by a $1 \times 1$ convolution with BN and ReLU.

**Local Feature Enhancement** Finally, we aggregate the original feature representation $\mathbf{x}_i$ and joint context representation $\mathbf{c}_i$ to obtain the augmented feature representation for pixel $p_i$:

$$\bar{\mathbf{x}}_i = \tau\left([\mathbf{c}_i^T, \mathbf{x}_i^T]^T\right), \tag{5}$$

where $\tau$ is a transformation function used to fuse the original feature representation and joint context representation, and implemented by a $1 \times 1$ convolution with BN and ReLU. The combination of the augmented features of all the pixels constitutes the augmented feature map $\bar{\boldsymbol{X}}$, which is used as the input to the pixel-to-offset prediction module.

### 3.3 Pixel-to-Offset Prediction Module

A depth image consists of pixels' 2D image plane coordinates and depth values (i.e., UVZ coordinates), which are the most direct information for determining the positions of hand joints. In this paper, we also decompose the 3D hand pose into joints' 2D image plane coordinates and depth values, and estimate these parameters in an ensemble way. More concretely, a pixel's UVZ coordinates and its offset vector to a joint can determine the joint's position in the UVZ coordinate system. That is, instead of predicting the joint's UVZ coordinates directly, we can detour estimate the offset vector from the pixel to the joint since the pixel's UVZ coordinates can be obtained from the depth image directly. To achieve robust estimation, we aggregate the predictions of all the pixels to obtain the position of the joint. Formally, for a certain joint $k$, we predict three offset values for each pixel representing the offset vector in the UVZ coordinate system from the pixel to joint $k$, resulting in three offset maps. Then the UVZ coordinates $(u_{j_k}, v_{j_k}, z_{j_k})$ of joint $k$ is obtained by a weighted average over all the pixels' predictions:

$$\begin{cases} u_{j_k} = \sum_i w_{ki} \left(u_{p_i} + \Delta u_{ki}\right) \\ v_{j_k} = \sum_i w_{ki} \left(v_{p_i} + \Delta v_{ki}\right), \\ z_{j_k} = \sum_i w_{ki} \left(z_{p_i} + \Delta z_{ki}\right) \end{cases} \tag{6}$$

where $(u_{p_i}, v_{p_i}, z_{p_i})$ indicate the UVZ coordinates of pixel $p_i$, $(\Delta u_{ki}, \Delta v_{ki}, \Delta z_{ki})$ represent the predicted offset values from pixel $p_i$ to joint $k$. $w_{ki}$ is the normalized prediction weight of pixel $p_i$, indicating its importance for locating the joint $k$, which is set to be same as the voting weight introduced in Section 3.2.1. The pixel-to-offset prediction module is implemented by a $1 \times 1$ convolution layer that takes the augmented local feature representation $\bar{\boldsymbol{X}}$ as input and output $3N$ offset maps for all the $N$ joints directly.

Note that our P2O module is much simpler than the estimation scheme of A2J [45] where two different branches are design to estimate the joints' UV coordinates and Z coordinates, respectively. In addition, A2J uses a single feature in high-level feature maps to predict multiple estimations for a set of anchor points, which may distract the model's representation learning as well as increasing the parameters. The experimental results also demonstrates the superiority of our method over the A2J.

### 3.4 Training Strategy

The predicted joints' coordinates are calculated as in formula (6). According to the ground truth 3D hand pose, we can construct a coordinate-wise regression loss:

$$L_{coordinate} = \sum_k \sum_c L_\delta \left(c_{j_k} - c^*_{j_k}\right), \tag{7}$$

where $c_{j_k}$ is one of the predicted UVZ coordinates of joint $k$, and $c^*_{j_k}$ is the corresponding ground truth coordinate. We choose the Huber loss function $L_\delta$ as the

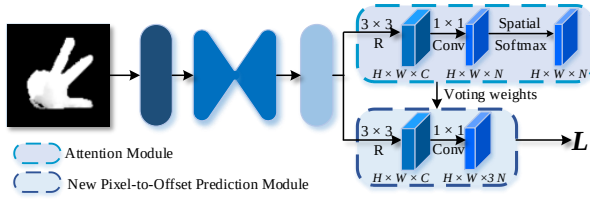


**Fig. 3.** The flowchart of the attention-based baseline model. The JGR module is replaced with an attention module to calculate the weights of pixels for locating the joints. The attention module consists of a 3x3 Residual block, a 1x1 Conv layer, and a spatial softmax operation. We also add a 3x3 Residual block to the original pixel-to-offset prediction module. This figure only depicts the one-stage version of the attention-based baseline model. In practice, we employ the two-stage version for comparison.

regression loss function since it is less sensitive to outliers in data than squared error loss function. Moreover, we can also explicitly supervise the generation process of offset maps by constructing a pixel-wise offset regression loss:

$$L_{offset} = \sum_{k} \sum_{i} \sum_{c} L_{\delta} \left( \Delta c_{ki} - \Delta c_{ki}^{*} \right), \tag{8}$$

where $\Delta c_{ki}$ is the offset value from pixel $p_i$ to joint $k$ along one of axes in the UVZ coordinate system, and $\Delta c_{ki}^{*}$ is the corresponding ground truth offset value. The pixel-wise offset regression loss can be seen as a regularization term for learning better local feature representation. Note that we normalize the ground truth coordinates and offset values to be within the range $[-1, 1]$, the pixel's UV coordinates and Z coordinates (i.e., depth values) are also normalized to be within the range $[0, 1]$ and $[-1, 1]$ respectively. Therefore, the estimated offset maps and joints' coordinates are also the normalized versions. We use a downsampled input depth image with the same resolution as the predicted offset map to calculate these parameters. Following [24], we boost the learning capability of the network architecture by stacking multiple hourglasses with identical structures, feeding the output from the previous module as the input into the next while exerting intermediate supervision at the end of each module. The final loss for the whole network is defined as follows:

$$L = \sum_{s=1}^{S} \alpha L_{coordinate}^{(s)} + \beta L_{offset}^{(s)}. \tag{9}$$

where $L_{coordinate}^{(s)}$ and $L_{offset}^{(s)}$ are the coordinate-wise regression loss and pixel-wise offset regression loss at the $sth$ stage, $\alpha = 1$ and $\beta = 0.0001$ are the weight factors that are used to balance the proposed two kinds of losses, and $S = 2$ is the total number of the stacked hourglasses. The whole network architecture is trained in an end-to-end style with the supervision of this loss.

**Table 1.** Comparison of different graph structures in the proposed JGR module. #Params indicates the number of parameters of the whole model.

| Graph Structures | Mean error (mm) | #Params |
|---|---|---|
| Skeleton Graph | **8.29** | 1.37M |
| Feature Similarity | 8.45 | 1.43M |
| Parameterized Matrix | 8.36 | 1.37M |

**Table 2.** Effectiveness of individual components of the proposed method.

| Component | | | Mean error (mm) |
|---|---|---|---|
| P2O | Offset Loss | JGR | |
| ✓ | | | 10.83 |
| ✓ | ✓ | | $10.54^{-0.29}$ |
| ✓ | ✓ | ✓ | $\mathbf{8.29}^{-2.25}$ |

# 4 Experiments

## 4.1 Datasets and Settings

We evaluate our proposed JGR-P2O on three common 3D hand pose estimation datasets: ICVL dataset [37], NYU dataset [39], and MSRA dataset [34]. The ICVL dataset contains 330K training and 1.5K testing depth images that are captured with an Intel Realsense camera. The ground truth hand pose of each image consists of $N = 16$ joints. The NYU dataset was captured with three Microsoft Kinects from different views. Each view consists of 72K training and 8K testing depth images. There are 36 joints in each annotated hand pose. Following most previous works, we only use view 1 and $N = 14$ joints for training and testing in all experiments. The MSRA dataset consists of 76K training images captured from 9 subjects with 17 gestures, using Intel's Creative Interactive Camera. Each image is annotated with a hand pose with $N = 21$ joints. We use the leave-one-subject-out cross-validation strategy [34] for evaluation.

We employ two most commonly used metrics to evaluate the performance of 3D hand pose estimation. The first one is the mean 3D distance error (in mm) averaged over all joints and all test images. The second one is the percentage of success frames in which the worst joint 3D distance error is below a threshold.

All experiments are conducted on a single server with four NVIDIA 1080Ti GPU using Tensorflow. For inputs to the JGR-P2O, we crop a hand area from the original image using a method similar to the one proposed in [25] and resize it to a fixed size of 96x96. The depth values are normalized to [-1, 1] for the cropped image. For training, Adam with weight decay of 0.00005 and batch size of 32 is used to optimize all models. Online data augmentation is used, including in-plane rotation ([-180, 180] degree), 3D scaling ([0.9, 1,1]), and 3D translation ([-10, 10] mm). The initial learning rate is set to be 0.0001, reduced by a factor of 0.96 every epoch. We train 8 epochs for the ICVL training set and 58 epochs for the other training sets.

**Table 3.** Comparison of different numbers of stacked hourglass module.

| #Hourglasses | Mean error (mm) | #Params |
|---|---|---|
| 1 | 8.63 | 0.72M |
| 2 | 8.29 | 1.37M |
| 3 | 8.27 | 2.02M |

**Table 4.** Comparison with different baselines on NYU.

| Model | Mean error (mm) | #Params |
|---|---|---|
| Baseline with Attention Module | 8.72 | 1.42M |
| Baseline with DHM Module | 8.69 | 1.37M |
| Ours | **8.29** | 1.37M |

### 4.2 Ablation Studies

We firstly conduct ablation studies to demonstrate the effectiveness of various components of the proposed JGR-P2O. The ablation studies are conducted on the NYU dataset since it is more challenge than the other two.

**Comparison of different graph structures.** Table 1 reports the performance of different graph structures in the proposed joint graph reasoning module. It can be seen that different graph structures can obtain similar estimation precision, indicating that the proposed joint graph reasoning module has strong generalization capability. In the following experiments, we choose the skeleton graph as the default graph structure for the joint graph reasoning module since it is more interpretable and best-performed.

**Effectiveness of individual components.** The results in Table 2 show how much each component improves the estimation performance along with the combinations of other components. The simplest baseline that combines the backbone network and a P2O module, denoted as P2O in Table 2, estimates the joint's positions with the average summation over all pixels' predictions and obtains highest estimation error. Adding the pixel-wise offset regression loss for training decreases estimation error by 0.29mm. Finally, the JGR module helps to greatly decrease the estimation error by 2.25mm.

**Number of hourglass modules.** The results of using different numbers of hourglass modules are reported in Table 3. It can be seen that with only one hourglass, the proposed JGR-P2O would achieve relatively low mean 3D distance errors (8.63mm) on the NYU dataset. Increasing the number of hourglasses can improve the estimation precision, but three hourglasses can only obtain negligible improvement. In this paper, we stack only two hourglasses to balance accuracy and efficiency.

### 4.3 Comparison with different baselines

To demonstrate the effectiveness of the proposed JGR and P2O module, we compare them with related baseline methods. The results are shown in Table 4.

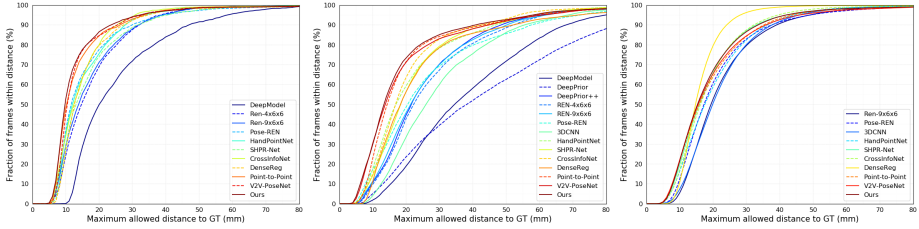


**Fig. 4.** Comparison with previous state-of-the-art methods. The percentages of success frames over different error thresholds are presented in this figure. Left: ICVL dataset, Middle: NYU dataset, Right: MSRA dataset.

**JGR vs. Attention.** To verify the effectiveness of the proposed JGR module, we design an attention-based baseline model where the JGR module is replaced with an attention module to calculate the weights of pixels for locating the joints. The flowchart of the baseline model can be found in Figure 3. As shown in Table 4, the JGR module outperforms the attention module by reducing the mean 3D distance error by 0.43mm on the NYU dataset, while having fewer parameters. It demonstrates that the JGR module is indeed useful for better local feature learning.

**P2O vs. Differentiable heat-map (DHM).** To demonstrate the effectiveness of the proposed P2O module, we compare our model with a model by replacing the P2O module with the DHM module proposed in [13]. DHM implicitly learns the joints' depth maps and heatmap distributions, while our P2O explicitly estimates the offsets from pixels to joints. It can be seen from Table 4 that our P2O module surpasses the DHM module by reducing the mean 3D distance error from 8.69mm to 8.29mm on NYU, which demonstrates the superiority of the proposed P2O module.

### 4.4 Comparison with state-of-the-art

We compare our proposed JGR-P2O with state-of-the-art deep learning-based methods, including both dense prediction-based methods: dense regression network (DenseReg) [43], Point-to-Point [10], Point-to-Pose Voting [17], A2J [45], and V2V [22], and direct regression-based methods: model-based method (DeepModel) [50], DeepPrior [26], improved DeepPrior (DeepPrior++) [25], region ensemble network (Ren-4x6x6 and Ren-9x6x6 [12]), Pose-guided REN (Pose-Ren) [2], 3DCNN [9], HandPointNet [7] , SHPR-Net [3] and CrossInfoNet [6]. The percentages of success frames over different error thresholds and mean 3D distance errors are shown in Figure 4 and Table 5, respectively.

It can be seen that dense prediction-based methods are generally superior to direct regression-based methods. As shown in Table 5, our method can achieve the lowest mean estimation errors (6.02mm and 8.29mm) on the ICVL and NYU dataset. Figure 4 also shows that the proportions of success frames of our method are highest when the error thresholds are lower than 30mm and 50mm on the

**Table 5.** Comparison with previous state-of-the-art methods on the ICVL, NYU and MSRA dataset. Mean error indicates the average 3D distance error. Type DR and DP indicate the direct regression-based method and dense prediction-based method, respectively. #Params indicates the parameter quantity of the whole network. Speed indicates the running speed during testing.

| Method | Mean error (mm) | | | Type | #Params | Speed (fps) |
|---|---|---|---|---|---|---|
| | ICVL | NYU | MSRA | | | |
| DeepModel[50] | 11.56 | 17.04 | - | DR | - | - |
| DeepPrior[26] | 10.40 | 19.73 | - | DR | - | - |
| DeepPrior++[25] | 8.10 | 12.24 | 9.50 | DR | - | 30.0 |
| REN-4x6x6[12] | 7.63 | 13.39 | - | DR | - | - |
| REN-9x6x6[12] | 7.31 | 12.69 | 9.70 | DR | - | - |
| Pose-REN[2] | 6.79 | 11.81 | 8.65 | DR | - | - |
| 3DCNN[9] | - | 14.1 | 9.60 | DR | 104.9M | 215 |
| HandPointNet[7] | 6.94 | 10.54 | 8.50 | DR | 2.58M | 48.0 |
| SHPR-Net[3] | 7.22 | 10.78 | 7.76 | DR | - | - |
| CrossInfoNet[6] | 6.73 | 10.08 | 7.86 | DR | 23.8M | 124.5 |
| DenseReg[43] | 7.30 | 10.2 | **7.20** | DP | 5.8M | 27.8 |
| Point-to-Point[10] | 6.30 | 9.10 | 7.70 | DP | 4.3M | 41.8 |
| V2V-PoseNet[22] | 6.28 | 8.42 | 7.59 | DP | 457.5M | 3.5 |
| Point-to-Pose Voting[17] | - | 8.99 | - | DP | - | 80.0 |
| A2J[45] | 6.46 | 8.61 | - | DP | 44.7M | 105.1 |
| JGR-P2O(Ours) | **6.02** | **8.29** | 7.55 | DP | 1.4M | 111.2 |

ICVL and NYU dataset, respectively. Our method obtains the second-lowest estimation error (7.55mm) on the MSRA dataset, which is only 0.35mm higher than the estimation error (7.20mm) of DenseReg [43].

Table 5 also shows that our method has the minimum model size and fastest running speed, compared with state-of-the-art dense prediction-based methods. Specifically, the total parameter quantity of our network is only 1.4M, and the running speed of our method is 111.2fps, including 2.0ms for reading and pre-processing image, and 7.0ms for network inference on a NVIDIA 1080Ti GPU.

More experimental analysis including qualitative results can be found in the supplementary material.

## 5 Conclusions

In this work, we propose a new prediction network (JGR-P2O) for 3D hand pose estimation from single depth images. Within JGR-P2O the GCN-based joint graph reasoning module can help to learn better local feature representation by explicitly modeling the dependencies among joints and the relations between pixels and joints, and the pixel-to-offset prediction module unifies the dense pixel-wise offset predictions and direct joint regression for end-to-end training, fully exploiting the 2.5D property of depth images. Extensive experiments demonstrate the superiority of the JGR-P2O concerning for both accuracy and efficiency.

## References


1. Cai, Y., Ge, L., Liu, J., Cai, J., Cham, T.J., Yuan, J., Thalmann, N.M.: Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2272–2281 (2019)
2. Chen, X., Wang, G., Guo, H., Zhang, C.: Pose guided structured region ensemble network for cascaded hand pose estimation. Neurocomputing (2019)
3. Chen, X., Wang, G., Zhang, C., Kim, T.K., Ji, X.: Shpr-net: Deep semantic hand pose regression from point clouds. IEEE Access **6**, 43425–43439 (2018)
4. Chu, X., Yang, W., Ouyang, W., Ma, C., Yuille, A.L., Wang, X.: Multi-context attention for human pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1831–1840 (2017)
5. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in neural information processing systems. pp. 3844–3852 (2016)
6. Du, K., Lin, X., Sun, Y., Ma, X.: Crossinfonet: Multi-task information sharing based hand pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9896–9905 (2019)
7. Ge, L., Cai, Y., Weng, J., Yuan, J.: Hand pointnet: 3d hand pose estimation using point sets. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8417–8426 (2018)
8. Ge, L., Liang, H., Yuan, J., Thalmann, D.: Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3593–3601 (2016)
9. Ge, L., Liang, H., Yuan, J., Thalmann, D.: 3d convolutional neural networks for efficient and robust hand pose estimation from single depth images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1991–2000 (2017)
10. Ge, L., Ren, Z., Yuan, J.: Point-to-point regression pointnet for 3d hand pose estimation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 475–491 (2018)
11. Guleryuz, O.G., Kaeser-Chen, C.: Fast lifting for 3d hand pose estimation in ar/vr applications. In: 2018 25th IEEE International Conference on Image Processing (ICIP). pp. 106–110. IEEE (2018)
12. Guo, H., Wang, G., Chen, X., Zhang, C.: Towards good practices for deep 3d hand pose estimation. arXiv preprint arXiv:1707.07248 (2017)
13. Iqbal, U., Molchanov, P., Breuel Juergen Gall, T., Kautz, J.: Hand pose estimation via latent 2.5 d heatmap regression. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 118–134 (2018)
14. Khamis, S., Taylor, J., Shotton, J., Keskin, C., Izadi, S., Fitzgibbon, A.: Learning an efficient model of hand shape variation from depth images. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2540–2548 (2015)
15. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
16. Li, M., Chen, S., Chen, X., Zhang, Y., Wang, Y., Tian, Q.: Actional-structural graph convolutional networks for skeleton-based action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3595–3603 (2019)

17. Li, S., Lee, D.: Point-to-pose voting based hand pose estimation using residual permutation equivariant layer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11927–11936 (2019)
18. Li, Y., Gupta, A.: Beyond grids: Learning graph representations for visual recognition. In: Advances in Neural Information Processing Systems. pp. 9225–9235 (2018)
19. Liang, X., Hu, Z., Zhang, H., Lin, L., Xing, E.P.: Symbolic graph reasoning meets convolutions. In: Advances in Neural Information Processing Systems. pp. 1853–1863 (2018)
20. Madadi, M., Escalera, S., Baró, X., Gonzalez, J.: End-to-end global to local cnn learning for hand pose recovery in depth data. arXiv preprint arXiv:1705.09606 (2017)
21. Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., Bronstein, M.M.: Geometric deep learning on graphs and manifolds using mixture model cnns. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5115–5124 (2017)
22. Moon, G., Yong Chang, J., Mu Lee, K.: 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. pp. 5079–5088 (2018)
23. Mueller, F., Mehta, D., Sotnychenko, O., Sridhar, S., Casas, D., Theobalt, C.: Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1284–1293 (2017)
24. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: European conference on computer vision. pp. 483–499. Springer (2016)
25. Oberweger, M., Lepetit, V.: Deepprior++: Improving fast and accurate 3d hand pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 585–594 (2017)
26. Oberweger, M., Wohlhart, P., Lepetit, V.: Hands deep in deep learning for hand pose estimation. arXiv preprint arXiv:1502.06807 (2015)
27. Oberweger, M., Wohlhart, P., Lepetit, V.: Training a feedback loop for hand pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3316–3324 (2015)
28. Oberweger, M., Wohlhart, P., Lepetit, V.: Generalized feedback loop for joint hand-object pose estimation. IEEE transactions on pattern analysis and machine intelligence (2019)
29. Pavlakos, G., Zhou, X., Derpanis, K.G., Daniilidis, K.: Coarse-to-fine volumetric prediction for single-image 3d human pose. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7025–7034 (2017)
30. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 652–660 (2017)
31. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in neural information processing systems. pp. 5099–5108 (2017)
32. Remelli, E., Tkach, A., Tagliasacchi, A., Pauly, M.: Low-dimensionality calibration through local anisotropic scaling for robust hand model personalization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2535–2543 (2017)

33. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 12026–12035 (2019)
34. Sun, X., Wei, Y., Liang, S., Tang, X., Sun, J.: Cascaded hand pose regression. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 824–832 (2015)
35. Supancic, J.S., Rogez, G., Yang, Y., Shotton, J., Ramanan, D.: Depth-based hand pose estimation: data, methods, and challenges. In: Proceedings of the IEEE international conference on computer vision. pp. 1868–1876 (2015)
36. Supančič, J.S., Rogez, G., Yang, Y., Shotton, J., Ramanan, D.: Depth-based hand pose estimation: methods, data, and challenges. International Journal of Computer Vision **126**(11), 1180–1198 (2018)
37. Tang, D., Jin Chang, H., Tejani, A., Kim, T.K.: Latent regression forest: Structured estimation of 3d articulated hand posture. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3786–3793 (2014)
38. Tang, D., Ye, Q., Yuan, S., Taylor, J., Kohli, P., Keskin, C., Kim, T.K., Shotton, J.: Opening the black box: Hierarchical sampling optimization for hand pose estimation. IEEE transactions on pattern analysis and machine intelligence (2018)
39. Tompson, J., Stein, M., Lecun, Y., Perlin, K.: Real-time continuous pose recovery of human hands using convolutional networks. ACM Transactions on Graphics (ToG) **33**(5), 169 (2014)
40. Tzionas, D., Ballan, L., Srikantha, A., Aponte, P., Pollefeys, M., Gall, J.: Capturing hands in action using discriminative salient points and physics simulation. International Journal of Computer Vision **118**(2), 172–193 (2016)
41. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
42. Wan, C., Probst, T., Van Gool, L., Yao, A.: Crossing nets: Combining gans and vaes with a shared latent space for hand pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 680–689 (2017)
43. Wan, C., Probst, T., Van Gool, L., Yao, A.: Dense 3d regression for hand pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5147–5156 (2018)
44. Wan, C., Yao, A., Van Gool, L.: Hand pose estimation from local surface normals. In: European conference on computer vision. pp. 554–569. Springer (2016)
45. Xiong, F., Zhang, B., Xiao, Y., Cao, Z., Yu, T., Zhou, J.T., Yuan, J.: A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 793–802 (2019)
46. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
47. Ye, Q., Yuan, S., Kim, T.K.: Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation. In: European conference on computer vision. pp. 346–361. Springer (2016)
48. Yuan, S., Garcia-Hernando, G., Stenger, B., Moon, G., Yong Chang, J., Mu Lee, K., Molchanov, P., Kautz, J., Honari, S., Ge, L., et al.: Depth-based 3d hand pose estimation: From current achievements to future goals. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2636–2645 (2018)
49. Yuan, S., Ye, Q., Garcia-Hernando, G., Kim, T.K.: The 2017 hands in the million challenge on 3d hand pose estimation. arXiv preprint arXiv:1707.02237 (2017)

50. Zhou, X., Wan, Q., Zhang, W., Xue, X., Wei, Y.: Model-based deep hand pose estimation. arXiv preprint arXiv:1606.06854 (2016)