# Exploiting Learnable Joint Groups for Hand Pose Estimation

**Moran Li**[1*], **Yuan Gao**[1,2*], **Nong Sang**[1]
[1] Key Laboratory of Image Processing and Intelligent Control,
School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan, China
[2] Tencent AI Lab
moran_li@hust.edu.cn, ethan.y.gao@gmail.com, nsang@hust.edu.cn

## Abstract

In this paper, we propose to estimate 3D hand pose by recovering the 3D coordinates of joints in a *group-wise* manner, where less-related joints are automatically categorized into different groups and exhibit different features. This is different from the previous methods where all the joints are considered holistically and share the same feature. The benefits of our method are illustrated by the principle of multi-task learning (MTL), i.e., by separating less-related joints into different groups (as different tasks), our method learns different features for each of them, therefore efficiently avoids the negative transfer (among less related tasks/groups of joints). The key of our method is a novel binary selector that automatically selects related joints into the same group. We implement such a selector with binary values stochastically sampled from a *Concrete* distribution, which is constructed using *Gumbel softmax* on trainable parameters. This enables us to preserve the differentiable property of the whole network. We further exploit features from those less-related groups by carrying out an additional feature fusing scheme among them, to learn more discriminative features. This is realized by implementing multiple 1x1 convolutions on the concatenated features, where each joint group contains a unique 1x1 convolution for feature fusion. The detailed ablation analysis and the extensive experiments on several benchmark datasets demonstrate the promising performance of the proposed method over the state-of-the-art (SOTA) methods. Besides, our method achieves top-1 among all the methods that do not exploit the dense 3D shape labels on the most recently released FreiHAND competition at the submission date. The source code and models are available at https://github.com/moranli-aca/LearnableGroups-Hand.

## Introduction

3D hand pose estimation is essential to facilitate convenient human-machine interaction through touch-less sensors. Therefore, it receives increasingly interests in various areas including computer vision, human-computer interaction, virtual/augmented reality, and robotics. The input of 3D hand pose estimation differs with the touch-less sensors, ranging from a single 2D RGB image (Zimmermann and Brox 2017; Cai et al. 2018; Mueller et al. 2018; Spurr et al. 2018; Yang and Yao 2019; Boukhayma, Bem, and Torr 2019), stereo RGB images (Zhang et al. 2017), to depth maps (Moon, Yong Chang, and Mu Lee 2018; Wan et al. 2018; Ge, Ren, and Yuan 2018; Ge et al. 2018; Du et al. 2019; Xiong et al. 2019). This paper considers 3D hand pose estimation from *a single 2D RGB image*, as it can be easily acquired from an ordinary RGB camera, therefore being most widely applicable in practice.

Nevertheless, estimating 3D hand pose from a single 2D image is ambiguous. This is because multiple 3D poses correspond to the same 2D projection as the depth information is eliminated. Fortunately, the valid hand poses lie in a space with much lower dimensions, which can be learned to alleviate the ambiguities in a data-driven manner by deep learning technologies. Specifically, recent deep learning methods exploit the relationship of the hand joints/keypoints holistically to recover the valid hand poses. Recent representative methods typically realize this idea by using a fully connected network (FCN) on all the 2D joints to recover the 3D poses (Zimmermann and Brox 2017; Spurr et al. 2018)

It is arguable that not all joints are (equally) related. For example, joints that are far away from each other are less related (*i.e.*, less constrained by skeleton/nerves) than those that are closed. This suggests that learning shared features for all the joints and exploiting them by a FCN is less appropriate. More specifically, *from multi-task learning (MTL) perspective, recovering the 3D coordinates of each joint corresponds to a single task, this suggests that using shared features for less related tasks may lead to negative transfer and therefore results in degraded performance*.

The above discussion motivates us to learn different features for less related joints. This can be implemented by separating the joints into different groups, where related joints are contained in the same group and share the same features, representing one task in MTL. However, being different from the standard MTL, the group where the related joints should be categorized into (*i.e.*, the task) is unknown in 3D hand pose estimation. In this work, *we learn the groups of related joints in an end-to-end manner by introducing a novel differentiable binary joint selectors, which is stochastically sampled from a Concrete distribution* (Maddison, Mnih, and Teh 2016).

Moreover, features from different groups can be further exploited in our method without introducing undesirable

[*] Equal contributions

negative transfer. To do that, *we implement a feature embedding motivated by (Gao et al. 2019), which learns what to share automatically to exploit the useful features in an embedded feature space*. The proposed feature embedding scheme is formulated by feature concatenation and 1-by-1 convolution, which can also be trained end-to-end. The full design of our method is illustrated in Fig.1. In summary, the contributions of the proposed method include:

- We consider 3D hand pose estimation from a single RGB image as a multi-task learning problem, where we group related joints as one task and learns different features for different groups/tasks to avoid the negative transfer.
- We learn the groups of joints (i.e., the tasks) automatically by end-to-end trainable binary joint selectors. We learn such binary joint selectors by stochastically sampling from a *Concrete* distribution (Maddison, Mnih, and Teh 2016), which is constructed by performing *Gumbel softmax* (Jang, Gu, and Poole 2016) reparameterization on trainable continuous parameters.
- We further exploit features from different groups/tasks without introducing undesirable negative transfer, by learning feature embeddings among different groups also in an end-to-end manner.

The remaining of this paper is organized as follows. We first discuss the related works. Then, the details of the proposed method are presented, followed by the implementation details, experiment results, including detailed ablation analysis. Finally, we give conclusions. The FreiHAND competition including root recovery and qualitative results for the benchmark datasets are included in the supplementary materials.

## Related Works

**3D Hand Pose Estimation from a Single RGB Image.** Current works in this area can be roughly categorized into direct regression methods including single-stage methods (Spurr et al. 2018; Yang and Yao 2019; Yang et al. 2019) that directly regress 3D joints locations from single RGB images, two-stage ways (Zimmermann and Brox 2017; Cai et al. 2018; Mueller et al. 2018; Doosti et al. 2020) that first regress 2D joints locations then lift 2D to 3D , and latent 2.5D estimation (Iqbal et al. 2018; Spurr et al. 2020). Besides, some recent works focused on shape estimation (Malik et al. 2018; Boukhayma, Bem, and Torr 2019; Ge et al. 2019; Baek, Kim, and Kim 2019; Hasson et al. 2019; Zhang et al. 2019; Zimmermann et al. 2019; Kulon et al. 2020; Baek, Kim, and Kim 2020). These methods generally combine discriminate methods with some generative methods (*e.g.*, the MANO (Romero, Tzionas, and Black 2017)) to improve generalization. Although the rich prior information (*e.g.*, the geometrical/biological dynamic constraints of hands joints) embedded in the generative model can assist discriminate model learning, such methods usually need some extra expensive shape annotations and iterative refinements during training which are not efficient enough and needed careful initialization. All of those methods treat all joints equally and view joints estimation as a single-task problem. In this work, we treat joints regression from the multi-task perspective for the intuition that different joints play different roles and posse different interrelationships.

**Multi-task Learning.** Many recent works in detection, semantic segmentation, and human pose estimation have adopted MTL methods to boost performance with auxiliary labels or tasks (Liang et al. 2019; Lee, Na, and Kim 2019; Pham et al. 2019). Misra et al. (Misra et al. 2016) implemented a cross-stitch unit for cross tasks features sharing which requires elaborate network architecture design. GradNorm (Chen et al. 2017) automatically balanced the loss of different tasks during training to make the network focus more on difficult tasks. Liu et al. (Liu, Johns, and Davison 2019) used a soft-attention module for shared feature selection with dynamic tasks-specific weights adjustments needing extra parameters. Gao et al. (Gao et al. 2020, 2019) proposed an end-to-end "plug-and-play" $1 \times 1$ convolution embedding layer to better leverage all different tasks features with a negligible parameter increase. In this work, we treat the hand joints estimation problem from the MTL view. Since the difference among different hand joints is not as large as that among different tasks of the general MTL problems, the dynamic loss adjustment methods or some complex network architecture design are not appropriate enough for our problem. We adopt the lightweight and effective MTL methods proposed by (Gao et al. 2019).

**Joints Grouping.** Since, arguably, not all joints are equally related, some recent works (Madadi et al. 2017; Chen et al. 2019; Du et al. 2019; Zhou et al. 2018; Tang and Wu 2019) have implemented manually designed grouping. Du et al. (Du et al. 2019) divided hand joints into the palm and fingers groups considering much more flexibility of those fingers. Zhou et al. (Zhou et al. 2018) grouped hand joints into the thumb, index, and other fingers for the reason that the combination of thumb and index finger can generate some gestures without other fingers. These manually designed group strategies have some limitations: one is that different people have different intuitions for joins grouping with different explanations; the other is that it is hard to decide which grouping way is better without extra comprehensive experiments. More importantly, the manual grouping way only considers the bio-constrains/structure of hands and does not include the dataset-dependent information. Tang and Wu (Tang and Wu 2019) proposed a data-driven way to calculate joints spatial mutual information and using spectral clusters to generate joints groups, but this method needs extra pre-statistic analysis for each dataset and does not include the bio-structure of hands. Hence, we propose a novel automatically learnable grouping method to implicitly learn the dataset-dependent and bio-structure dependent grouping. Our method can avoid human intuition ambiguity and extra pre-analysis.

## Method

In this paper, we treat the 3D coordinates regression of multiple hand joints as a multi-task learning problem. By categorizing the most related joints into a group, we can formulate multiple groups as multiple tasks. This enables us to learn a unique feature set for each group, which avoids the

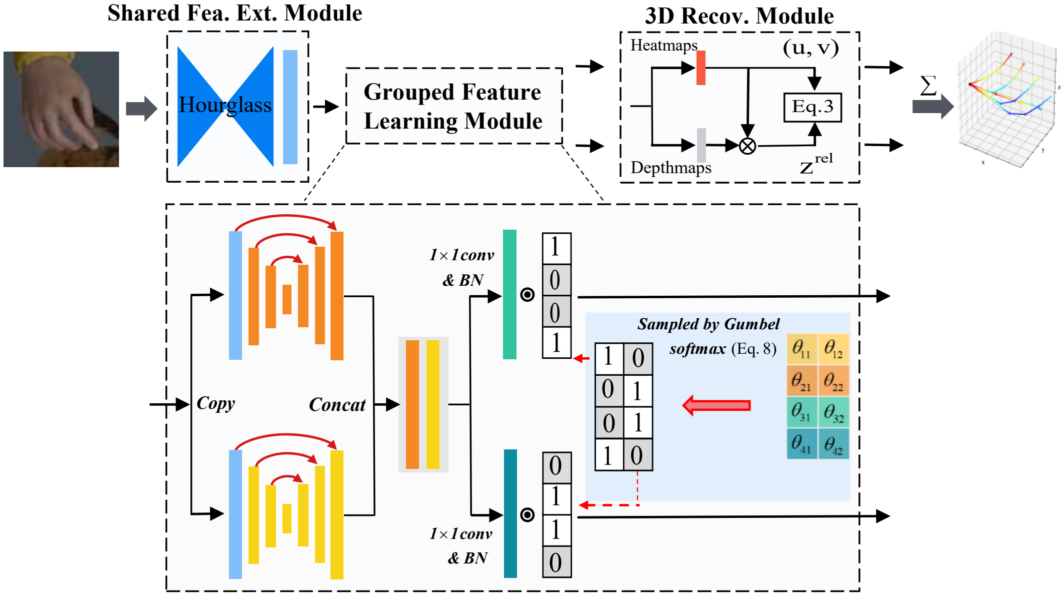


Figure 1: Overview of the proposed method. Our model consists of the *Shared Feature Extraction Module, Grouped Feature Learning Module*, and *3D Recovery Module*. $\odot$ denotes the element-wise product, and $\theta_{i,j}$'s are learnable network parameters for the binary selectors. We illustrate two groups for conciseness, while our method is able to categorize the joints into an arbitrary number of groups (we use $\#\text{Group} = 3$ in most of our experiments). (Best view in colors.)

potential negative transfer when exploiting a shared feature for all the joints (Zimmermann and Brox 2017; Spurr et al. 2018; Mueller et al. 2018; Cai et al. 2018; Iqbal et al. 2018; Yang and Yao 2019; Boukhayma, Bem, and Torr 2019; Ge et al. 2019; Zhang et al. 2019).

In the following, we first give an overview of the proposed method. Then, we detail the design of our grouped feature learning module, which automatically groups most related joints without violating the end-to-end training. The proposed grouped feature learning module further enables learning a discriminative feature embedding to exploit features from different groups. Finally, we give the loss functions to train the entire network.

## Overview

Our method consists of three modules, *i.e.*, a shared feature extraction module; a grouped feature learning module that categorizes the joints into multiple groups, and learns a unique feature set for each of them, to avoid negative feature transfer across groups; a 3D joints recovery module that regresses the 2D location and the relative depth of each joint, and finally recovers the 3D joint coordinates using the intrinsic camera parameters. We choose a similar design of the shared feature extraction module and the 3D joints recovery module to those used in (Iqbal et al. 2018), which benefits from i) leveraging well-developed 2D joints estimation (Newell, Yang, and Deng 2016), and ii) estimating relative depth between joints instead of much more difficult absolute depth. Note that the core of our method, *i.e.*, the grouped feature learning module, is general and can be applied to most (if not all) hand pose estimation models, such as (Ge et al. 2019; Boukhayma, Bem, and Torr 2019; Zhang et al. 2019; Yang et al. 2019).

The architecture of our method is shown in Fig. 1. We detail the design of the shared feature extraction module and the 3D joints recovery module in this section. We leave the core of our method, *i.e.*, the general-applicable grouped feature learning module, discussed in the next section.

**Shared Feature Extraction Module.** The hourglass network (Newell, Yang, and Deng 2016) illustrates the great potential in extracting features that are especially suitable to represent the joints/keypoints (Sun et al. 2018; Wan et al. 2018; Ge et al. 2019; Zhang et al. 2019). It is built on symmetric encoders and decoders. We use the hourglass network to extract the shared features from the input image, which are then fed to the grouped feature learning module to learn the unique features for each group of joints.

**3D Joints Recovery Module.** The group-unique features learned from the grouped feature learning module are fed to the 3D joints recovery module to recover the 3D coordinates of each joint.

Specifically, the group-unique features are first decoded to learn the 2D heatmaps and the (relative) depth maps simultaneously. Then, the 2D coordinates and the relative depth of *each joint* are obtained at the maxima location of each heatmap channel. After that, the 2D coordinates and the (relative) depth of each joint are exploited to recover the 3D joints in the camera coordinate, by using the intrinsic camera parameters and a depth root. Specifically, our goal is to estimate the 3D joint coordinates $(x_i, y_i, z_i)$ from the 2D estimated coordinates $(u_i, v_i)$ and the estimated relative depth $z_i$. Following the configuration of (Zimmermann and Brox 2017; Cai et al. 2018; Spurr et al. 2018; Iqbal et al. 2018; Yang and Yao 2019; Yang et al. 2019), we assume the global hand scale $s_0$, the depth of the root joint $z_{root}$, and the perspective camera intrinsic parameters $M = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix}$ are known[1], where $f_x, f_y$ are the focal lengths, $p_x, p_y$ are the principal point coordinates of the camera. Therefore, we have the following relationship:

$$[x_i, y_i, z_i]^T = sM^{-1}[u_i, v_i, 1]^T, \tag{1}$$

where $s$ is a scalar to balance the equation, and we have $s = z_i$ from the third equation of Eq. (1).

In addition, $z_i$ and $z_i^{rel}$ are only up to a global translation $z_{root}$ and a global hand scale $s_0$:

$$z_i = s_0 z_i^{rel} + z_{root}, \tag{2}$$

Therefore, we have:

$$[x_i, y_i, z_i]^T = (s_0 z_i^{rel} + z_{root})M^{-1}[u_i, v_i, 1]^T. \tag{3}$$

In the next section, we will detail how to learn a unique feature set for each joint group which finally decodes to the 3D coordinates.

## Grouped Feature Learning Module

We treat the 3D hand pose estimation as a multi-task learning problem by categorizing the joints into different groups as multiple tasks, where we construct multiple network branches to learn a unique feature set for each group, therefore efficiently avoid the negative transfer across different tasks (*i.e.*, joints groups).

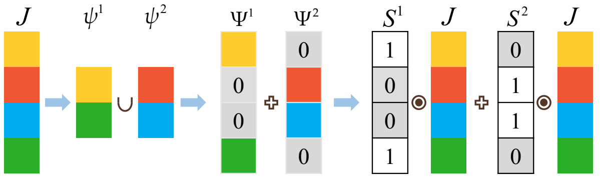


Figure 2: A simple illustration of our grouping operation. We show our grouping operation on four joints with two groups for conciseness. $\odot$ denotes the element-wise multiplication. The gray and the white blocks denote 0 and 1, respectively. (Best view in colors.)

To avoid ambiguities of group construction by different researchers (Zhou et al. 2018; Du et al. 2019), we learn the best group formation in a fully data-driven manner. This is implemented by learning several binary selectors so that each selects one group of joints. We restrict that each joint should be selected exactly once, *i.e.*, every joint should be selected to construct the groups, and different groups do not contain overlapped joints. Formally, we aim to group $N$ joints into $K$ groups, denoted as:

$$J = \bigcup_{k=1}^{K} \psi^k \quad \text{and} \quad \forall(i,j), \psi^i \cap \psi^j = \emptyset, \tag{4}$$

where $J = \{(x_i, y_i, z_i)\}_{i=1}^{N} \in \mathbb{R}^{N\times 3}$ is the coordinates of $N$ joints, $\psi^k \in \mathbb{R}^{G_k \times 3}$ is the coordinates of $G_k$ joints in the $k^{th}$ group with $\sum_{k=1}^{K} G_k = N$, $\emptyset$ is the null set. For ease of presentation, we augment each $\psi^g$ with 0's, as shown in Fig. 2, so that the augmented matrix $\Psi^g \in \mathbb{R}^{N\times 3}$ has the same dimension as $J$. Therefore, Eq. (4) can be rewritten as:

$$J = \bigcup_{k=1}^{K} \psi^k = \sum_{k=1}^{K} \Psi^k = \sum_{k=1}^{K} S^k \odot J, \tag{5}$$

where $\odot$ denotes element-wise product, $S^k = \{S_1^k, ..., S_N^k\} \in \mathbb{R}^{N\times 1}$ is a column vector representing the binary joints selector. Denoting $\mathbb{1} \in \mathbb{R}^{N\times 1}$ as the column vector with all 1's, the element of $S^k$, *i.e.*, $S_i^k$, has the following property:

$$S_i^k = \begin{cases} 1 & \text{if } J_i \in \Psi^k \\ 0 & \text{if } J_i \notin \Psi^k \end{cases} \quad \text{and} \quad \sum_{k=1}^{K} S^k = \mathbb{1}. \tag{6}$$

We learn each $S^k$ under constraints of Eq. (6) as trainable model parameters of a neural network. To do that, we further construct a matrix $\mathcal{S} \in \mathbb{R}^{N\times K}$ and reformulate the constraints accordingly as:

$$\begin{aligned} &\mathcal{S} = [S^1, ..., S^K] \in \mathbb{R}^{N\times K}, \\ \text{s.t.} \quad &\forall i, \sum \mathcal{S}_{i,\cdot} = 1, \text{and } \mathcal{S}_{i,j} \in \{0, 1\}, \end{aligned} \tag{7}$$

where $\mathcal{S}_{i,j}$ is the element in $i$-th row and $j$-th column of $\mathcal{S}$, and $\mathcal{S}_{i,\cdot}$ is the $i$-th row of $\mathcal{S}$.

Equation (7) shows that each row $\mathcal{S}_{i,\cdot}$ follows a *categorical* distribution. While we can easily impose the first constraint of Eq. (7) with a *softmax* operation, the second constraint which binarizes each element $S_{i,j}$ violates the differentiable property of the whole network.

To alleviate this issue, we reparameterize the *categorical* distribution of $\mathcal{S}_{i,\cdot}$ using a *Concrete* distribution to get the *continuous relaxation* of $\mathcal{S}_{i,j}$ (Maddison, Mnih, and Teh 2016), denoted as $\tilde{\mathcal{S}}_{i,j}$:

$$\tilde{\mathcal{S}}_{i,j} \sim \frac{\exp(\theta_{i,j} + O_{i,j})/\tau}{\sum_{k=1}^{K} \exp(\theta_{i,k} + O_{i,k})/\tau}, \tag{8}$$

where $\theta_{i,j}$ is a learnable parameter of the network, representing the *logits* of $\tilde{\mathcal{S}}_{i,j}$. $O_{i,j} = -\log(-\log(U_{i,j}))$ is the *Gumble* variable with $U_{i,j} \sim (0, 1)$ as a *uniform* distribution. $\tau = \tau(\text{training step})$ is the temperature parameter which is annealed to 0 with the training proceeds. It is shown in (Maddison, Mnih, and Teh 2016) that the continuous relaxation in Eq. (8), i.e., the *concrete* distribution, smoothly approaches to the *categorical* distribution when the temperature $\tau$ approaches to 0. Therefore, we can sample $\tilde{\mathcal{S}}_{i,j}$ from Eq. (8) as a good approximation for the binary selector $\mathcal{S}_{i,j}$.

[1] Typically, the camera intrinsic parameters $M$ can be obtained by the EXIF information of the image, and we can (almost) always recover the global hand scale $s_0$ and the depth of the root joints $z_{root}$ by the Procrustes alignment between the estimations and the ground-truth (Schönemann 1966).

The above analysis enables us to automatically categorize $N$ joints into $K$ groups, so that we can construct $K$ network branches, where each branch learns a unique feature set for each group. This efficiently alleviates the negative transfer across groups.

**Feature Fusing Across Groups.** We show that the different features learned by different groups can be further exploited to obtain more discriminative features, without introducing negative feature transferring. This is inspired by (Gao et al. 2019) via learning multiple discriminative feature embeddings on the concatenated features from all the groups.

Specifically, denoting features from the $l$-th layer of the $k$-th group as $\mathbf{F}_l^k \in \mathbb{R}^{B\times H\times W\times C}$, where $B, H, W, C$ are the batch size, the height, the width, and the number of channels of the feature map, we can learn a more discriminative feature embedding $\hat{\mathbf{F}}_l^k \in \mathbb{R}^{B\times H\times W\times C}$ for group $k$ exploiting features from all the groups:

$$\begin{aligned}\hat{\mathbf{F}}_l^k &= \texttt{BatchNorm}\Big([\mathbf{F}_l^1, ..., \mathbf{F}_l^K] \cdot [\alpha_1^k \mathbf{I}_C, ..., \alpha_K^k \mathbf{I}_C]^\top\Big) \\ &\xleftarrow{\text{Init. }\texttt{1x1conv}} \texttt{BatchNorm}\Big(\texttt{1x1conv}\big([\mathbf{F}_l^1, ..., \mathbf{F}_l^K]\big)\Big),\end{aligned} \tag{9}$$

where $[\mathbf{F}_l^1, ..., \mathbf{F}_l^k] \in \mathbb{R}^{B\times H\times W\times KC}$ is the concatenated feature from all the groups at layer $l$, and $[\alpha_1^k \mathbf{I}_C, ..., \alpha_K^k \mathbf{I}_C]^\top \in \mathbf{R}^{KC\times C}$ (with $\mathbf{I}_C \in \mathbf{R}^{C\times C}$ as the identity matrix) *is a (reshaped) learnable 1x1 convolution with size $KC \times 1 \times 1 \times C$, which generalizes the weighted sum of K features*.

In order to learn a good $\hat{\mathbf{F}}_l^k$ without introducing negative transfer, initially, it is reasonable to rely more on the original feature $F^k$ for the same group $k$, and the features from other groups $F^{i\neq k}$ also count for feature fusion, but with much smaller initial weights. Therefore, we carefully initialize $\alpha_k^k = 0.9$, which is the "generalized weighted-sum" weight for the original $\mathbf{F}_l^k$. We initialize the remaining $\alpha_{i\neq k}^k = (1 - \alpha_k^k)/(K-1)$ for $F_l^{i\neq k}$, so that ensuring $\sum_{i=1}^K \alpha_i^k = 1$. This was also adopted in (Gao et al. 2019).

## Loss

We use $\ell_1$ losses to train the network (Sun et al. 2018; Iqbal et al. 2018; Spurr et al. 2020). Denoting our estimation and the ground-truth 3D coordinates of the joints in the camera coordinate as $[X, Y, Z]^T = \{(x_i, y_i, z_i)\}_{i=1}^N \in \mathbb{R}^{N\times 3}$ and $[\hat{X}, \hat{Y}, \hat{Z}]^T = \{(\hat{x_i}, \hat{y_i}, \hat{z_i})\}_{i=1}^N \in \mathbb{R}^{N\times 3}$, respectively. We introduce a hyperparameter $\beta$ to balance the gradient magnitudes of XY and Z loss, hence the training loss is:

$$L = ||[X, Y]^\top - [\hat{X}, \hat{Y}]^\top||_1 + \beta||Z - \hat{Z}||_1 \tag{10}$$

# Experiment

## Datasets and Protocols

**RHD** (Rendered Hand Pose Dataset) (Zimmermann and Brox 2017) is a synthesized rendering hand dataset containing 41,285 training and 2,728 testing samples. Each sample provides an RGB image with resolution 320 × 320, hand mask, depth map, 2D and 3D joints annotation, and camera parameters. This dataset is built upon 20 different characters performing 39 actions with large variations in hand pose and viewpoints.

**STB** (Stereo Hand Pose Benchmark) (Zhang et al. 2017) is a real hand dataset containing 18,000 stereo pairs samples. It also provides RGB images with resolution 640 × 480, depth images, 2D and 3D joints annotations, and camera parameters for each sample. This dataset contains 6 different backgrounds, and each background has counting and random pose sequence. For each sample, we use one of the pair's images since the other contains the same pose and in almost the same viewpoint. We split this dataset into a training set with 15,000 images and an evaluation set with 3,000 images following (Zimmermann and Brox 2017).

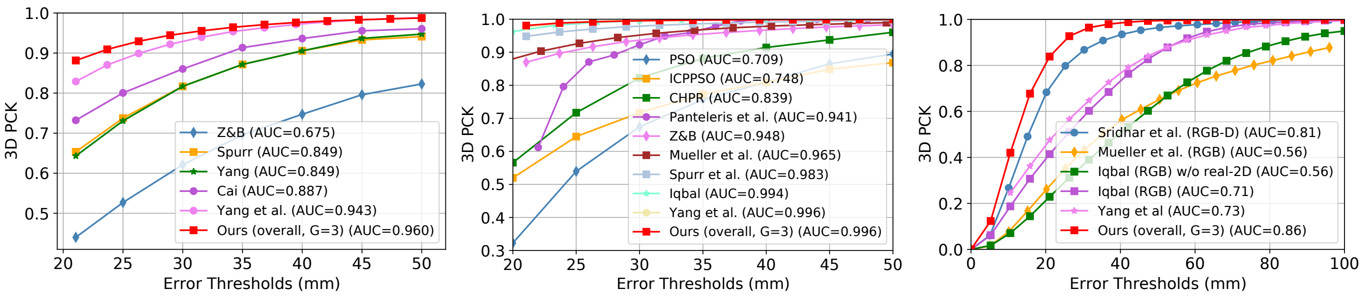
**Dexter + Object** (Dexter and Object) (Sridhar et al. 2016) is a real hand object interaction dataset consisting of six sequences with two actors (one female). For each sample, it provides an RGB image with resolution 640 × 480, depth image, camera parameters, and 3D annotation only for fingertips and three cubic corner joints of each object. We use this dataset as a cross-dataset evaluation similar to (Zimmermann and Brox 2017; Zhang et al. 2019).

**FreiHAND** (Zimmermann et al. 2019) is the latest released real hand dataset containing 130,240 training and 3,960 testing samples. Each training sample includes an RGB image with resolution 224 × 224, a 3D dense mesh ground-truth, and 3D joints annotations with hand scale and the camera parameters. The annotations of testing samples are not provided and the evaluation is conducted by submitting predictions to the online evaluation system. A lot of pose samples with severe occlusions contained in the FreiHAND makes it more challenging than other benchmark datasets.

**Evaluation Protocols** We use the common metrics to evaluate the accuracy of the estimated 3D hand poses including mean/median end-point-error (3D mean/median EPE), the area under the curve (AUC) of the percentage of correct keypoints (PCK) with different thresholds. We assume that the global hand scale and the root joint location are known for the RHD and the STB datasets, following a similar condition as used in (Zimmermann and Brox 2017; Cai et al. 2018; Spurr et al. 2018; Iqbal et al. 2018; Yang and Yao 2019; Yang et al. 2019). For experiments on the Dexter + Object dataset, we follow the same way as Yang et al. (Yang and Yao 2019). On the FreiHAND dataset, only the hand scale and camera parameters are given for the testing samples. To be consistent with previous works, we adopt the root recovery method similar to Spurr et al. (Spurr et al. 2020) to generate absolute root joints depth (see Supp.).

## Implementation Details

**Data Processing.** We implement the data pre-processing and augmentation similar to Yang et al. (Yang et al. 2019). Specifically, we first crop original RGB images using the bounding box calculated by the ground truth masks and resize the cropped image to 256 × 256. Then, we apply an online data augmentation with a random scaling between $[1, 1.2]$, a random rotation between $[-\pi, \pi]$, a random translation between $[-20, 20]$, and a color jittering with a random hue between $[-0.1, 0.1]$. As for the global hand scale and root joints, we follow the same way as Yang et al. (Yang



(a) RHD (b) STB (c) Dexter + Object

Figure 3: Comparison with SOTA methods on the RHD, the STB, and the Dexter + Object datasets. For the RHD dataset, the results of some methods (*e.g.*, Kulon et al. (Kulon et al. 2020), Iqbal et al. (Iqbal et al. 2018)) are not shown here because their AUC curves of this dataset are not available yet. (Best view in colors.)

et al. 2019) using the MCP of the middle finger as the root joint and selecting the euclidean distance between the MCP and the PIP of the middle finger as the global scale for fair comparisons. We also align the annotations of the RHD and the STB datasets so that the joints with the same annotation index have the same semantic meaning.

**Optimization.** We use Adam optimizer (Kingma and Ba 2014) to train the network. We train the *shared feature extraction module* (hourglass network) to predict 2D joints location using $\ell_1$ loss for initialization, with a learning rate of 1e-3 and a mini-batch size of 64. Then, we use the loss function defined in Eq. (10) with $\beta = 20$ to optimize the overall network. The learning rates for the newly introduced *grouped feature learning module* and *feature fusing module* are 1e-1 and 1e-2, respectively. For the remaining network parameters, the learning rate is set to 1e-4 with a mini-batch size of 32.

For training, we initialize every $\theta_{i,j}$ of Eq. (8) to be $1/K$ as we do not impose priors for the group categorization ($K$ is the number of groups). $\tau$ of Eq. (8) is initialized to be 5 and decrease 0.1 for every 1,000 steps until it reaches around 0. We use the number of groups as 3 (i.e., $K = 3$) in all of our experiments as we find that further increasing the number of groups produces comparable results (as shown in our ablation analysis in the main paper), which coincides with the conclusion from (Tang and Wu 2019).

## Benchmark Results

We compare our proposed method with the SOTA methods (Sridhar et al. 2016; Zimmermann and Brox 2017; Cai et al. 2018; Mueller et al. 2018; Panteleris, Oikonomidis, and Argyros 2018; Spurr et al. 2018; Iqbal et al. 2018; Yang and Yao 2019; Yang et al. 2019), on various benchmark datasets to illustrate the effectiveness of our proposed method. The performances of the SOTA methods on each dataset are obtained from their original paper.

As shown in Fig. 3(a), our method outperforms the SOTA methods (Zimmermann and Brox 2017; Spurr et al. 2018; Yang and Yao 2019; Cai et al. 2018; Yang et al. 2019) by a large margin on the RHD dataset, which demonstrates the promising performance of our proposed method. Comparing with (Iqbal et al. 2018) (AUC [20-50] is 0.94), and (Kulon et al. 2020) (AUC [20-50] is 0.956), our proposed method (AUC [20-50] is 0.96) also has better performance.

Our method gets comparable results on the STB dataset with the SOTA methods (Panteleris, Oikonomidis, and Argyros 2018; Zimmermann and Brox 2017; Mueller et al. 2018; Spurr et al. 2018; Iqbal et al. 2018; Yang et al. 2019; Cai et al. 2019) without using additional training data. Note that several works such that (Cai et al. 2019; Baek, Kim, and Kim 2020; Theodoridis et al. 2020) (AUC [20-50] are 0.995, 0.995, 0.997, respectively) are not included in Fig. 3(b) because their AUC curves are not available yet. Apart from less training data used, such comparable results are also partly because the STB dataset is a much easier dataset, where many algorithms perform considerably well.

For the Dexter + Object dataset, we follow the same training setup as that used in (Zimmermann and Brox 2017; Mueller et al. 2018; Yang and Yao 2019; Yang et al. 2019), which makes use of both the RHD and the STB datasets for training. But note that our method does not include extra augmentation of objects. Besides, our evaluation on this dataset is the same as (Yang et al. 2019) which adopts the best root and scale. Following previous methods, the AUC [0-50] is reported, and the comparison is shown in Fig. 3(c).

To further demonstrate the effectiveness of our method, we also show the detailed results on the FreiHAND dataset in Table 1. For the FreiHAND competition, our method ranks Top-1 over the submissions without exploiting the dense 3D shape label at the submission date (see Supp.).

## Ablation Analysis

We perform detailed ablation analysis mainly on the RHD dataset to investigate each proposed component of the grouped feature learning module.

Firstly, we are especially interested in investigating that i) *whether categorizing the hand joints into groups improves the performance*? If so, ii) *how about grouping the joints manually*? Moreover, iii) *whether further learning a unique feature embedding for each group, by exploiting the features from all the groups, is effective*? We validate those hypotheses in Table 2, where the manual grouping of joints is sim-

| Methods | Use Shape | Mean EPE [mm] (↓) | AUC [0-100] (↑) |
|---|---|---|---|
| Boukh. et al. (Boukhayma, Bem, and Torr 2019) | ✓ | 3.5 | 0.351 |
| Hasson et al. (Hasson et al. 2019) | ✓ | 1.33 | 0.737 |
| MANO Fit (Zimmermann et al. 2019) | ✓ | 1.37 | 0.73 |
| MANO CNN (Zimmermann et al. 2019) | ✓ | 1.1 | 0.783 |
| Kulon et al. (Kulon et al. 2020) | ✓ | 0.84 | 0.834 |
| Baseline (**ours**) | - | 0.932 | 0.816 |
| + Group Learned. | - | 0.862 | 0.830 |
| + Fea. Fuse. | - | 0.856 | 0.831 |
| + **Ensemble** | - | **0.796** | **0.842** |

Table 1: Results (aligned (Schönemann 1966)) on the FreiHAND dataset, where baseline means not to learn joint groups (*i.e.*, the number of groups is 1). Fea. Fuse means whether to learn the (unique) feature embedding for each group exploiting the features from all the groups. Our checkpoints ensemble (Liu and Liu 2018) result reaches the best performance. ↑/↓ represents the higher/lower the better.

| Group | | Fea. Fuse | Mean EPE [mm] (↓) | Median EPE [mm] (↓) | AUC [20-50] (↑) |
|---|---|---|---|---|---|
| Manual | Learned | | | | |
| - | - | - | 11.534 | 8.885 | 0.951 |
| ✓ | - | - | 11.205 | 8.622 | 0.954 |
| ✓ | - | ✓ | 11.021 | 8.543 | 0.957 |
| - | ✓ | - | 10.771 | 8.283 | 0.958 |
| - | ✓ | ✓ | **10.653** | **8.200** | **0.960** |

Table 2: Ablation analysis of the proposed method on the RHD dataset, where Fea. Fuse means whether to learn the feature embedding for each group exploiting the features from all groups. The Mean/Median EPE are in mm.

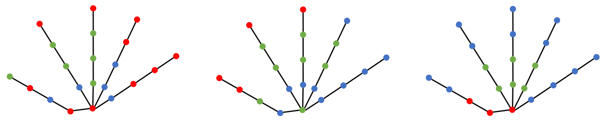

(a) RHD (b) STB (c) FreiHAND

Figure 4: The automatically learned joints grouping results on the RHD, the STB, and FreiHAND datasets (with the number of groups equal to 3). The red, green, and blue represent the three groups, respectively. (Best view in colors.)

| #Group | Mean EPE (↓) | Median EPE (↓) | AUC [20-50] (↑) |
|---|---|---|---|
| Baseline | 11.534 | 8.885 | 0.951 |
| 2 | 11.042 | 8.478 | 0.956 |
| **3** | **10.771** | **8.283** | **0.958** |
| 4 | 10.808 | 8.340 | 0.958 |
| 5 | 10.923 | 8.418 | 0.957 |
| 21 | 10.964 | 8.422 | 0.956 |

Table 3: Ablation analysis of the number of groups (#Group) on the RHD dataset, where baseline means not to learn joint groups (*i.e.*, #Group $= 1$).

ilar to the idea of (Tang and Wu 2019) and we separate the hand joints into three groups (*i.e.*, thumb, index, and other fingers) following Zhou et al. (Zhou et al. 2018). Table 2 illustrates that the proposed method, with learnable joints grouping and feature fusing among all the groups, achieves the best performance. The improvements shown in Table 1 further illustrate the effectiveness of the proposed modules.

After validating the effectiveness of the proposed method, the next immediate question is *how many groups should we categorize the joints into*? In Table 3, we show the performance when learning 2-5 groups, which demonstrates that learning larger than 3 groups produces comparable results, coinciding with the conclusion from (Tang and Wu 2019).

The grouping results (shown in Fig.4) validate that the joints cluster is not only related to constraints due to the skeletons and nerves, but also related to the distribution of poses in the datasets (Jahangiri and Yuille 2017; Tang and Wu 2019). For example, one dataset may contain more hand images from a kitchen scenario, while another is collected from sporting scenarios, intuitively they should have distinctive grouping results, and our method can well characterize those differences. We have verified the good transferring-ability of the learned groups to a different dataset, please see Fig. 3(c), where the model is trained on the RHD and the STB datasets and evaluated on the Dexter + Object dataset.

## Conclusions

We propose a novel method for 3D hand pose estimation from a single RGB image, which automatically categorizes the hand joints into groups. By treating groups as different tasks that learn different features, our method efficiently avoids negative transfer across groups. Moreover, we further exploit features from different groups to learn a more discriminative feature embedding for each group. We carried out extensive experiments and detailed ablation analysis to illustrate the effectiveness of our method and the proposed network can be optimized end-to-end in deep neural networks. Our results on the RHD, the STB, the Dexter + Object, and the FreiHAND datasets significantly outperform the SOTA methods.

## Broader Impact

Our work improves hand pose estimation performance which can facilitate much easier animation fabrication, more accurate sign language recognition, and many other human-computer interaction applications. All of those contribute a lot to us especially carton filmmakers and physically challenged people. What's more, our work has some positive impacts on the academic research community. Our proposed method can be easily applied to many other tasks such as human pose estimation, hand objects estimation, and other related tasks.

Since our method does not use the identity information of each individual, the authors believe there is no offensive to ethical or personal privacy.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China under grant 61871435 and the Fundamental Research Funds for the Central Universities no. 2019kfyXKJC024.

## References

Baek, S.; Kim, K. I.; and Kim, T.-K. 2019. Pushing the Envelope for RGB-based Dense 3D Hand Pose Estimation via Neural Rendering. In *CVPR*.

Baek, S.; Kim, K. I.; and Kim, T.-K. 2020. Weakly-Supervised Domain Adaptation via GAN and Mesh Model for Estimating 3D Hand Poses Interacting Objects. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Boukhayma, A.; Bem, R. d.; and Torr, P. H. 2019. 3d hand shape and pose from images in the wild. In *CVPR*.

Cai, Y.; Ge, L.; Cai, J.; and Yuan, J. 2018. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *ECCV*.

Cai, Y.; Ge, L.; Liu, J.; Cai, J.; Cham, T.-J.; Yuan, J.; and Thalmann, N. M. 2019. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2272–2281.

Chen, X.; Wang, G.; Guo, H.; and Zhang, C. 2019. Pose guided structured region ensemble network for cascaded hand pose estimation. *Neurocomputing* .

Chen, Z.; Badrinarayanan, V.; Lee, C.-Y.; and Rabinovich, A. 2017. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. *arXiv preprint arXiv:1711.02257* .

Doosti, B.; Naha, S.; Mirbagheri, M.; and Crandall, D. J. 2020. HOPE-Net: A Graph-Based Model for Hand-Object Pose Estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Du, K.; Lin, X.; Sun, Y.; and Ma, X. 2019. CrossInfoNet: Multi-Task Information Sharing Based Hand Pose Estimation. In *CVPR*.

Gao, Y.; Bai, H.; Jie, Z.; Ma, J.; Jia, K.; and Liu, W. 2020. MTL-NAS: Task-Agnostic Neural Architecture Search towards General-Purpose Multi-Task Learning. In *CVPR*.

Gao, Y.; Ma, J.; Zhao, M.; Liu, W.; and Yuille, A. L. 2019. NDDR-CNN: Layerwise Feature Fusing in Multi-Task CNNs by Neural Discriminative Dimensionality Reduction. In *CVPR*.

Ge, L.; Cai, Y.; Weng, J.; and Yuan, J. 2018. Hand PointNet: 3d hand pose estimation using point sets. In *CVPR*.

Ge, L.; Ren, Z.; Li, Y.; Xue, Z.; Wang, Y.; Cai, J.; and Yuan, J. 2019. 3D Hand Shape and Pose Estimation from a Single RGB Image. In *CVPR*.

Ge, L.; Ren, Z.; and Yuan, J. 2018. Point-to-point regression pointnet for 3d hand pose estimation. In *ECCV*.

Hasson, Y.; Varol, G.; Tzionas, D.; Kalevatykh, I.; Black, M. J.; Laptev, I.; and Schmid, C. 2019. Learning joint reconstruction of hands and manipulated objects. In *CVPR*.

Iqbal, U.; Molchanov, P.; Breuel Juergen Gall, T.; and Kautz, J. 2018. Hand pose estimation via latent 2.5 d heatmap regression. In *ECCV*.

Jahangiri, E.; and Yuille, A. L. 2017. Generating multiple diverse hypotheses for human 3d pose consistent with 2d joint detections. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 805–814.

Jang, E.; Gu, S.; and Poole, B. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144* .

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Kulon, D.; Guler, R. A.; Kokkinos, I.; Bronstein, M. M.; and Zafeiriou, S. 2020. Weakly-Supervised Mesh-Convolutional Hand Reconstruction in the Wild. In *CVPR*.

Lee, W.; Na, J.; and Kim, G. 2019. Multi-Task Self-Supervised Object Detection via Recycling of Bounding Box Annotations. In *CVPR*.

Liang, M.; Yang, B.; Chen, Y.; Hu, R.; and Urtasun, R. 2019. Multi-Task Multi-Sensor Fusion for 3D Object Detection. In *CVPR*.

Liu, S.; Johns, E.; and Davison, A. J. 2019. End-To-End Multi-Task Learning With Attention. In *CVPR*.

Liu, T.; and Liu, B. 2018. Constrained-size Tensorflow Models for YouTube-8M Video Understanding Challenge. In *ECCV*.

Madadi, M.; Escalera, S.; Baró, X.; and Gonzalez, J. 2017. End-to-end global to local cnn learning for hand pose recovery in depth data. *arXiv preprint arXiv:1705.09606* .

Maddison, C. J.; Mnih, A.; and Teh, Y. W. 2016. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712* .

Malik, J.; Elhayek, A.; Nunnari, F.; Varanasi, K.; Tamaddon, K.; Heloir, A.; and Stricker, D. 2018. Deephps: End-to-end estimation of 3d hand pose and shape by learning from synthetic depth. In *2018 International Conference on 3D Vision (3DV)*, 110–119. IEEE.

Misra, I.; Shrivastava, A.; Gupta, A.; and Hebert, M. 2016. Cross-stitch networks for multi-task learning. In *CVPR*.

Moon, G.; Yong Chang, J.; and Mu Lee, K. 2018. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *CVPR*.

Mueller, F.; Bernard, F.; Sotnychenko, O.; Mehta, D.; Sridhar, S.; Casas, D.; and Theobalt, C. 2018. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *CVPR*.

Newell, A.; Yang, K.; and Deng, J. 2016. Stacked hourglass networks for human pose estimation. In *ECCV*.

Panteleris, P.; Oikonomidis, I.; and Argyros, A. 2018. Using a single RGB frame for real time 3D hand pose estimation in the wild. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 436–445. IEEE.

Pham, Q.-H.; Nguyen, T.; Hua, B.-S.; Roig, G.; and Yeung, S.-K. 2019. JSIS3D: Joint Semantic-Instance Segmentation of 3D Point Clouds With Multi-Task Pointwise Networks and Multi-Value Conditional Random Fields. In *CVPR*.

Romero, J.; Tzionas, D.; and Black, M. J. 2017. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (ToG)* 36(6): 245.

Schönemann, P. H. 1966. A generalized solution of the orthogonal procrustes problem. *Psychometrika* 31(1): 1–10.

Spurr, A.; Iqbal, U.; Molchanov, P.; Hilliges, O.; and Kautz, J. 2020. Weakly Supervised 3D Hand Pose Estimation via Biomechanical Constraints. *arXiv preprint arXiv:2003.09282* .

Spurr, A.; Song, J.; Park, S.; and Hilliges, O. 2018. Cross-modal deep variational hand pose estimation. In *CVPR*.

Sridhar, S.; Mueller, F.; Zollhöfer, M.; Casas, D.; Oulasvirta, A.; and Theobalt, C. 2016. Real-time joint tracking of a hand manipulating an object from rgb-d input. In *ECCV*.

Sun, X.; Xiao, B.; Wei, F.; Liang, S.; and Wei, Y. 2018. Integral human pose regression. In *ECCV*.

Tang, W.; and Wu, Y. 2019. Does Learning Specific Features for Related Parts Help Human Pose Estimation? In *CVPR*.

Theodoridis, T.; Chatzis, T.; Solachidis, V.; Dimitropoulos, K.; and Daras, P. 2020. Cross-Modal Variational Alignment of Latent Spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Wan, C.; Probst, T.; Van Gool, L.; and Yao, A. 2018. Dense 3D Regression for Hand Pose Estimation. In *CVPR*.

Xiong, F.; Zhang, B.; Xiao, Y.; Cao, Z.; Yu, T.; Zhou, J. T.; and Yuan, J. 2019. A2J: Anchor-to-Joint Regression Network for 3D Articulated Pose Estimation from a Single Depth Image. *arXiv preprint arXiv:1908.09999* .

Yang, L.; Li, S.; Lee, D.; and Yao, A. 2019. Aligning Latent Spaces for 3D Hand Pose Estimation. In *ICCV*.

Yang, L.; and Yao, A. 2019. Disentangling Latent Hands for Image Synthesis and Pose Estimation. In *CVPR*.

Zhang, J.; Jiao, J.; Chen, M.; Qu, L.; Xu, X.; and Yang, Q. 2017. A hand pose tracking benchmark from stereo matching. In *ICIP*. IEEE.

Zhang, X.; Li, Q.; Mo, H.; Zhang, W.; and Zheng, W. 2019. End-to-end hand mesh recovery from a monocular RGB image. In *ICCV*.

Zhou, Y.; Lu, J.; Du, K.; Lin, X.; Sun, Y.; and Ma, X. 2018. Hbe: Hand branch ensemble network for real-time 3d hand pose estimation. In *ECCV*.

Zimmermann, C.; and Brox, T. 2017. Learning to estimate 3d hand pose from single rgb images. In *ICCV*.

Zimmermann, C.; Ceylan, D.; Yang, J.; Russell, B.; Argus, M.; and Brox, T. 2019. FreiHAND: A Dataset for Markerless Capture of Hand Pose and Shape from Single RGB Images. In *ICCV*.