Accurate-PGNet: Learning to Assemble Perceptual Body Parts for Accurate Human Skeleton Establishment

Renjie Zhang, Di Lin, *Member, IEEE*, Xin Wang, George Baciu, *Senior Member, IEEE*, C. L. Philip Chen, *Life Fellow, IEEE*, and Ping Li, *Member, IEEE*

***Abstract*—The human skeleton establishment aims to provide accurate localization information of the human body from RGB images and establish a complete human skeleton for many applications, such as action recognition, video surveillance, and human-computer interaction. Considering the inherent human body structure, many recent methods group the relevant body parts and utilize the deep convolutional network to learn the visual context from the part groups. However, the grouping approaches used in these methods heavily rely on prior knowledge of the human body shape but lose important relationships between parts. In this paper, we introduce the Accurate Part Grouping Network (Accurate-PGNet), a novel network for hierarchically grouping body parts in a data-driven manner. In contrast to the previous methods, we use neural architecture search (NAS) to optimize the architecture of Accurate-PGNet and properly group the body parts. The part grouping respects the diverse visual patterns of parts, producing groups containing different body parts. From each group, we learn the visual feature map. It helps to capture the correlation between parts and predict their locations. The feature maps of the part groups are merged hierarchically to capture the higher-order context of parts in larger groups. We extensively evaluated our method on the challenging benchmarks, demonstrating that Accurate-PGNet effectively helps to achieve state-of-the-art results.**

***Index Terms*—Hierarchical body structure, human skeleton establishment, neural architecture search, perceptual body parts.**

I. Introduction

HUMAN skeleton establishment is a long-standing problem in multimedia and vision. It requires the localization and classification of human body parts, then builds a human skeleton by connecting them. Many applications (e.g., autonomous driving, video surveillance, human-computer interaction) benefit from such tasks, which provide detailed information about the human body. As a critical task, the accuracy and robustness of human skeleton establishment methods must be guaranteed. However, in real-world scenarios, we face variations of occlusion, truncation, scales, and human appearances, which may cause severe performance deterioration. To this end, accurately extracting human body part features from images is crucial to developing the human skeleton establishment and its relevant tasks. Thanks to the deep convolutional networks (DCNs) that learn discriminative features from the image data, recent deep-learning-based approaches [2], [3], [4], [5], [6], [7] can directly learn the visual features for the recognition of all parts of one person. However, these methods ignored the inherent knowledge of human body structure, only capturing the global visual features of the overall human body. Intuitively, the parts of an individual form a graph, where we denote a part as a node. Different groups of the parts capture specific spatial articulation and semantic correlation. Thus, state-of-the-art methods [1], [8], [9], [10] proposed to assemble single body parts into different part groups. From each group, DCN learns the spatial and semantic dependency between parts to enhance the visual features of parts.

Previous methods mainly adopt two different dependency types (connections) between parts, regarded as the edges in the graphical structure. The physical connection is illustrated in Fig. 1(a). It is based on a basic human skeleton and has been broadly used by many human pose estimation (HPE) methods [2], [11], [12], [13], [14] for part assembling. A couple of parts are assembled with a physical connection in-between (see the red edges at the bottom of Fig. 1(a)). A group of parts is regarded as a sub-graph of the entire graph. Note that the physical connections are deterministic, which means the long-range parts (see the red dots at the top of Fig. 1(a)) have to be connected via

Received 24 October 2023; revised 8 July 2024; accepted 4 August 2024. Date of publication 23 December 2024; date of current version 4 April 2025. This work was supported in part by The Hong Kong Polytechnic University under Grant P0048387, Grant P0042740, Grant P0044520, Grant P0043906, Grant P0049586, and Grant P0050657, and in part by the PolyU Research Institute for Sports Science and Technology under Grant P0044571. The associate editor coordinating the review of this article and approving it for publication was Prof. Sanghoon Lee. *(Renjie Zhang and Di Lin contributed equally to this work.) (Corresponding author: Ping Li.)*

Renjie Zhang, Xin Wang, and George Baciu are with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong (e-mail: renjie.zhang@connect.polyu.hk; xin1025.wang@connect.polyu.hk; csgeorge@polyu.edu.hk).

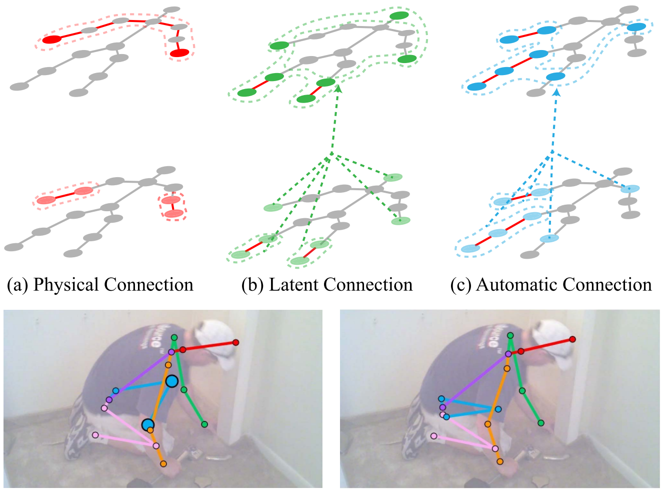
Di Lin is with the College of Intelligence and Computing, Tianjin University, Tianjin 300072, China (e-mail: ande.lin1988@gmail.com).

C. L. Philip Chen is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China, and also with the Pazhou Laboratory, Guangzhou 510335, China (e-mail: philip.chen@ieee.org).

Ping Li is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, also with the School of Design, The Hong Kong Polytechnic University, Hong Kong, and also with the Research Institute for Sports Science and Technology, The Hong Kong Polytechnic University, Hong Kong (e-mail: p.li@polyu.edu.hk).

This article has supplementary downloadable material available at https://doi.org/10.1109/TMM.2024.3521763, provided by the authors.

Digital Object Identifier 10.1109/TMM.2024.3521763


(a) Physical Connection (b) Latent Connection (c) Automatic Connection

(d) PBN (e) Ours

Fig. 1. The illustration of different connections and examples of skeleton establishment. The body parts (the gray dots) are naturally connected by (a) the physical connections (the red edges). The pictorial and mutual-correlation models rely on the intuition of the human body and employ (b) the latent connections (the multi-branch arrow in green) to yield the part group. We learn (c) the automatic connections (the multi-branch arrow in blue) for part grouping. (d) and (e) are the establishment results of PBN [1] and ours, respectively. Larger dots are the parts with problematic locations.

other parts, forming an overlarge group (see the dash-line region) that captures a less focused correlation of parts. This brings in much irrelevant information about correlation, which inevitably deteriorates the accuracy of the establishment. To overcome this problem, more recent methods [1], [9], [12], [15], [16], [17], [18] constructed the latent connections of parts. In contrast to the physical connection that normally connects a pair of parts, the latent connection connects two or more groups[1] (see the green dash-line arrow in Fig. 1(b)). The latent connections allow the long-range and highly correlated parts to be assembled into a group (see the dash-line region at the top of Fig. 1(b)) without requiring extra parts for connection. The part can join more groups in various ranges, achieving richer information from other parts for feature enhancement. The pictorial models [10], [12], [15], [18] and the mutual-correlation models [1], [17] established latent connections for groups. However, these models are straightforwardly based on the intuition of the human body (e.g., the adjacency of parts and the symmetric shape of the body) or simple clustering according to spatial information. They may miss the underlying connections essential for learning the higher-order context of parts. Moreover, the most effective part of the hierarchical structure may have yet to be found among various potential grouping patterns. For example, in Fig. 1(d), the occluded left knee and left ankle were detected wrongly by PBN [1], which utilized physical and latent connection. Inspired by the skeleton of the right leg, PBN [1] considered that the human in the image was on one knee, causing its mistake of skeleton establishment.

In response to these challenges, we propose a novel approach that treats the parts as perceptual components of the human body. These parts can be automatically combined by establishing group connections, which form larger groups. The novelty of our approach lies in its goal of recognizing parts accurately without relying on human experience or other prior knowledge. This approach has the potential to significantly improve the accuracy of human pose estimation.

Specifically, we propose an Accurate Part Grouping Network (Accurate-PGNet) to learn the connections between groups for providing accurate human skeleton information. Accurate-PGNet learns feature maps of groups. We also model the part grouping as the merging of feature maps. As illustrated in Fig. 2(a), we merge the feature maps at several stages, yielding the feature maps of new groups. In Fig. 2(b), we illustrate a single stage of feature merging. At each stage, we use *Group Recognition Blocks* (GRB) to predict the heat map for the visual feature map, computing specific feature maps with part group information for each group. The heat map indicates the locations of the parts in the same group. Next, we input the specific feature maps of different groups into a searchable *Part Grouping Block* (PGB) between the adjacent groups of feature maps. In PGB, we merge the feature maps using network connections. The feature merging uses a novel neural architecture search (NAS) strategy to determine the network connections between the specific and visual feature maps. Furthermore, the merging result can be regarded as feedback, which guides the optimization of the connection weights. We prune the unnecessary connections based on the optimized connection weight, yielding the optimized network architecture. Note that the architecture search of the PGB is guided by the task of recognizing the separate joints and by the searched result of the NAS strategy. It helps to optimize the network architecture better, allowing the features to be sensitive to the natural property of the part groups. The searched architecture captures the crucial effects of single parts and high-order semantic relationships, forming an effective hierarchical part grouping pattern. As shown in Fig. 1(e), without the solid misleading of traditional human structure knowledge, our method extracts contexts of human parts more freely, providing more accurate establishment results.

Our work makes the following three main contributions:

- We advocate using automatic connections of perceptual body parts without prior knowledge of the human body. We propose a framework that models the part grouping as the merging of visual feature maps and automatically determines the merging patterns, providing a richer context of parts for human skeleton establishment.
- We propose a novel NAS strategy by leveraging the intermediate result of part grouping to optimize the weights of network connections in every searching epoch. Considering leveraging the specific visual information of various human body parts to strengthen the learning of higher-level human structures, we consider the part grouping progressive and assemble the parts by combining the feature maps. Also, we use the searched structure in the previous epoch to initialize the next epoch, guiding the whole architecture search.

[1] We refer to a single part as a group, reducing the redundant differentiation of the part and the group.

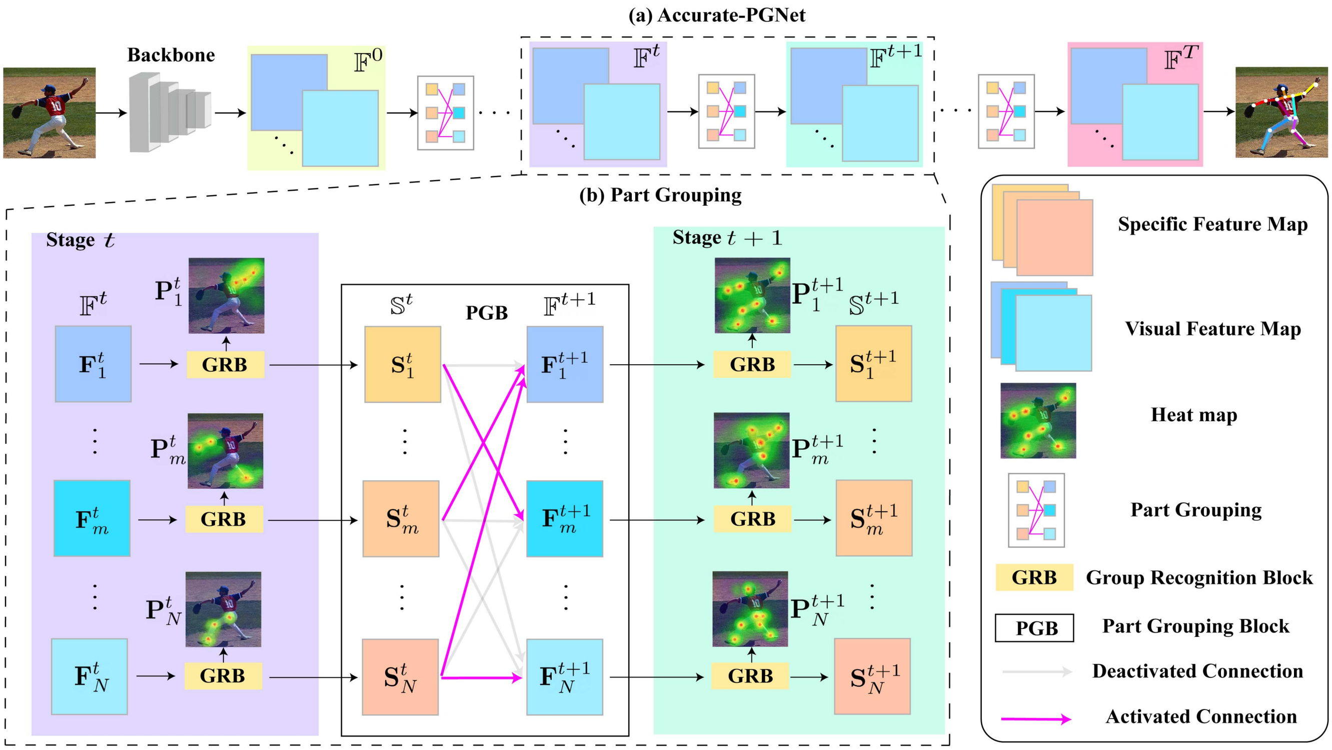


Fig. 2. The illustration of Accurate-PGNet. We input the image of the human pose into the backbone CNN, producing the feature maps for different body parts. Accurate-PGNet uses several stages to merge the feature maps. It computes the heat map for localizing the parts in the groups, as illustrated in (a), it merges the body parts at several stages. (b) From the $t \sim (t+1)$th stage, the *Group Recognition Blocks* (GRBs) take input as the visual feature maps and yield the heat map of the parts in different groups. Each GRB also computes the specific feature map for a group. The *Part Grouping Block* (PGB) is equipped with NAS. It determines the network connections for merging the specific feature maps and forming the visual feature maps of the new groups.

- Extensive experiments are conducted based on the MPII [19] and COCO [20] datasets. Our proposed NAS framework, Accurate-PGNet, achieves state-of-the-art accuracy on these two public datasets. Our searched network component can help achieve higher accuracy than most NAS-based networks. Compared with most traditional methods, Accurate-PGNet costs less computation resources with competitive performance. These verify that our proposed framework can obtain search results that contain the effective potential human skeleton knowledge.

II. Related Work

A. CNN-Based Human Skeleton Establishment

With the significant development of DCNs, many human skeleton establishment methods [2], [3], [4], [11] used DCNs to learn the visual features of parts and estimated the part locations. Typically, they included a backbone to capture high-level representations and a hand-crafted detection head to detect part locations. Xiao et al. [11] used the ResNet [21] as the backbone to extract global visual features and employed deconvolutional operations to obtain part locations. Sun et al. [3] designed multi-resolution fusion sub-networks as the backbone to exchange visual information between multiple resolutions. In the feature extraction of these CNN-based methods, multiple resolutions of convolutional feature maps were combined to model the relationships between parts from the perspective of global image space. The shared feature maps extracted by the backbone were used to localize all body parts. However, the shared feature maps obtained by traditional CNN-based methods may need to be more effective for capturing different parts' specific appearances and locations. In our work, we introduce a framework that leverages a hierarchical network to learn specific features for parts. With the shared representations learnt by the backbone and the part group representations by the hierarchical network, our framework simultaneously learns the part features globally and locally.

B. Body Part Grouping for Human Skeleton Establishment

The latest methods group the body parts and learn the visual context of body parts. The visual context enriches the part features. Tian et al. [14] proposed the articulation tree to represent all body parts. The tree structure connected the parts, forming the visual context. Park and Zhu [22] employed the annotations of bounding boxes to divide the body parts into various groups. According to the degree of freedom from the kinematics of human pose, Nie et al. [13] used different layers to group the parts with diverse appearances. Yang and Ramanan [23] used the relative part locations to indicate co-occurrence between parts and formed the groups. Tang and Wu [1] employed the normalized mutual information between each pair of body parts to measure their correlations. With the mutual information, they resorted to spectral clustering to group body parts. Qiu et al. [17] constructed dynamic graphs to tolerate the variations of human pose when grouping body parts. However, the above methods heavily rely on prescribed strategies for grouping body parts. They may miss the critical part groups, thus achieving sub-optimal performance. In contrast, we employ NAS to optimize the network architecture, merging the visual features of body parts to form the groups. The accurate recognition of body parts drives our grouping. We hierarchically merge the parts without requiring extra annotations or other prior knowledge. It captures higher-order relationships between the body parts.

C. Neural Architecture Search

NAS has been broadly used to learn network architecture for various visual recognition tasks. Some NAS methods depended on reinforcement learning [24], [25], [26] and evolutionary algorithms [27], [28] to optimize network architectures. Recently, Liu et al. [29] proposed the gradient-based method for NAS at lower computation cost. The current works [30], [31], [32] have demonstrated NAS's effectiveness in merging feature maps for feature augmentation. There have been HPE methods that borrow the success of NAS and achieve significant progress in HPE. Yang et al. [33] utilized NAS to search the cell-based neural fabrics to connect body parts that belong to already-determined groups. Bao et al. [34] searched the architecture of pose encoder to exchange visual context between various scales of parts, whose features were augmented by multi-scale information. Zhang et al. [35] restricted the search space of the network architecture for computing efficient backbone network for HPE. Gong et al. [36] used parallel branches to connect the feature maps of parts in various scales. In gradient-based NAS methods, the final step is network pruning. Early NAS pruning method [37] tried to learn the weights and connections simultaneously. However, Liu et al. [38] proved that the learnt weights are insignificant in the following learning process. Then, Mei et al. [39] proposed a dynamic network shrinkage method to cut down network architecture by removing those "dead" NAS blocks. However, these methods employed a general NAS strategy without considering the special structure knowledge of human body. In this paper, we propose a NAS scheme that respects the diverse patterns of the part groups for merging the feature maps. This approach dynamically prunes inactive connections between parts based on the architecture parameters of candidate operations and important factors. By doing so, we create more opportunities for discovering potential part groups that contain valuable information for human skeleton establishment. Moreover, we merge highly correlated body parts to produce more consistent representations of the part groups. This research has the potential to inspire new approaches in the fields of human pose estimation.

III. Overview

This paper proposes a novel framework called Accurate-PGNet to deal with the perceptual part ensemble issue for human skeleton establishment. As illustrated in Fig. 2, Accurate-PGNet learns the context information of the part groups via *Group Recognition block* (GRB). The context information enhances the part features. Accurate-PGNet starts from the feature maps of the independent parts. It hierarchically merges the feature maps corresponding to different part groups at multiple stages. This work introduces the NAS-based block called *Part Grouping Block* (PGB) for feature merging between two adjacent stages. We regard the activation of network connections in PGB as the feature merging and the part grouping. In the PGB, we propose a novel NAS strategy called *Part Grouping* to activate the connections associated with part-specific feature maps. The hierarchical framework learns the local and global semantic correlations between parts in this fashion. Fig. 3 illustrates the training process of the PGB between two adjacent stages. The training is divided into two phases: searching and fine-tuning. In the searching, we learn the network weights and group the feature maps of part groups iteratively. The part grouping has been updated, along with the connection weights. The *Part Grouping* merges part-specific feature maps to achieve this goal. In this strategy, we initialize the activation status with no connections activated. With updated connection weights, we employ 4 operations to determine useful groups, activate corresponding connections, and compute each group's new specific feature maps. Following traditional NAS methods, we set a convergence threshold for our optimization to indicate the end of searching. If the threshold is unmet, we continue updating the network connections with the newly obtained activation way and repeatedly employ the *Part Grouping*. Otherwise, we end the search and prune inactive connections. In the fine-tuning, we fix the pruned network architecture and only optimize the network weights. Finally, the semantic contexts of different levels are extracted via the learning of a fixed grouping network, bringing richer visual information about the human body.

IV. Automatic Part Grouping Network

In this section, we first elaborate on the overall architecture of Accurate-PGNet. Then, we give more details about the proposed GRB and PGB. Further, we describe the *Part Grouping*. Finally, we present the optimization details of our framework.

A. Network Architecture

We illustrate Accurate-PGNet in Fig. 2. We employ the backbone DCN to extract a set of visual feature maps, i.e., $\mathbb{F}^0 = \{\mathbf{F}_j^0 \in \mathbb{R}^{H\times W\times C} | j = 1, \ldots, J\}$, where $\mathbf{F}_j^0$ is the feature map of the jth body parts. $H \times W$ and C indicate the resolution and the number of channels of the feature map. J is the total number of body parts. Accurate-PGNet learns the semantic and spatial context from the groups of body parts to enhance the features of the individual parts. Accurate-PGNet consists of T stages of part grouping. At each stage, we rely on the visual feature map of each part group, using GRB to predict the part locations. Based on the visual feature map and part locations, GRB produces the specific feature map of each group. Next, we use PGB to merge the specific feature maps of different part groups. PGB produces the visual feature maps to represent the groups at the next stage.

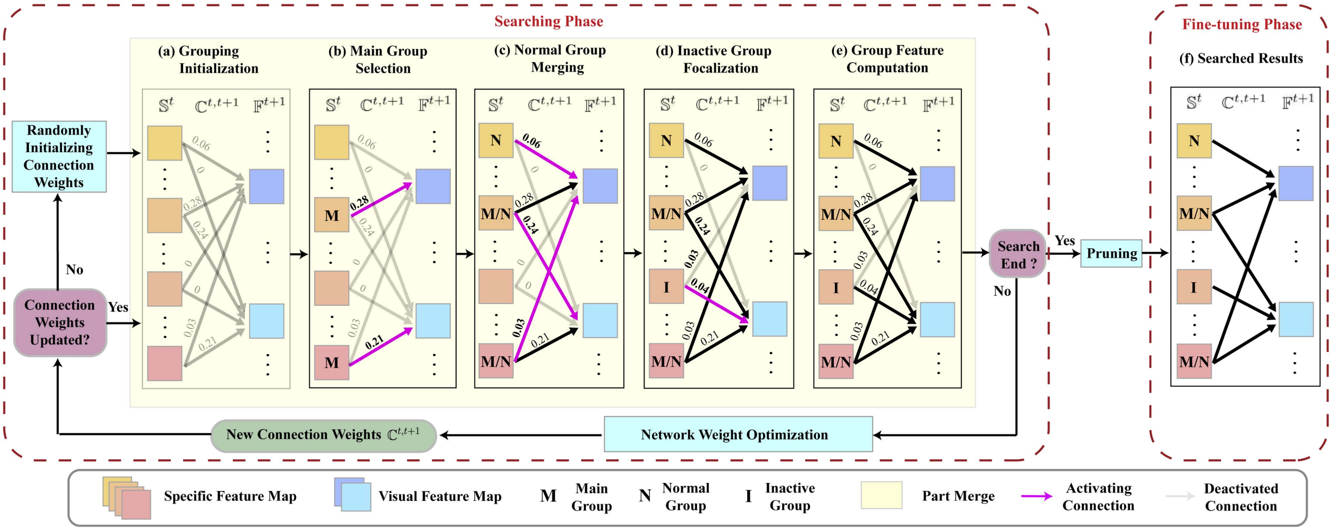


Fig. 3. Illustration of the training of the PGB between the tth and $(t+1)$th stages. In the searching phase, we use *Part Grouping* to merge feature maps. Given connections weights, It use (a) the grouping initialization to initialize the part feature maps' grouping. And it adopts (b) the main group selection, (c) the normal group merging and (d) the inactive group activation to progressively activate the connections. The hidden feature maps associated with activated connections are merged to compute (e) the visual feature maps of the new groups. If the searching continue, the obtained visual feature maps will be used to optimize the connection weights, which can be used in the next iteration of grouping. Otherwise, we prune all inactive connections to form a fixed architecture for the fine-tuning of the Accurate-PGNet. Here, the yellow box represent the *Part Grouping*. The multi-branch arrow represents the feature merging. The pink/black/transparent arrow indicates the activating/activated/inactive connection.

There are N groups[2] at the tth stage. We let $\mathbb{G}^t_m \in \{\mathbb{G}^t_1, \ldots, \mathbb{G}^t_N\}$ contain the indices of the human body parts in the mth group. The visual feature maps in the set $\mathbb{F}^t = \{\mathbf{F}^t_m \in \mathbb{R}^{H\times W\times C} | m = 1, \ldots, N\}$ represent all of the N groups. We feed the visual feature maps into GRB to localize the parts in each group. Given the specific feature map $\mathbf{F}^t_m$, GRB computes the heat maps $\mathbf{P}^t_m \in \mathbb{R}^{H\times W\times |\mathbb{G}^t_m|}$. Here, $\mathbf{P}^t_m$ is a combination of heat maps, and each heat map indicates the location of one part in the part group $\mathbb{G}^t_m$. The group-truth part locations supervise these heat maps. Then we use the heat map $\mathbf{P}^t_m$ and visual feature map $\mathbf{F}^t_m$ to compute the specific feature map $\mathbf{S}^t_m$ which contains the more specific part visual context for each group. For all groups, GRB produces the specific feature maps in set $\mathbb{S}^t = \{\mathbf{S}^t_m \in \mathbb{R}^{H\times W\times C} | m = 1, \ldots, N\}$. Moreover, we input the set of specific feature maps $\mathbb{S}^t$ into PGB, which merges the specific feature maps and yields the visual feature maps $\mathbb{F}^{t+1} = \{\mathbf{F}^{t+1}_m \in \mathbb{R}^{H\times W\times C} | m = 1, \ldots, N\}$ at the $(t+1)$th stage. The body parts in the group $\mathbb{G}^{t+1}_o$ are determined by architecture search of PGB, which learns the weighted connections $\mathbb{C}^{t,t+1} = \{\mathbf{c}^{t,t+1}_{m,o} \in \mathbb{R} | m, o = 1, \ldots, N\}$ between the specific feature maps in $\mathbb{S}^t$ and the visual feature maps in $\mathbb{F}^{t+1}$. They are computed based on activation, deactivation, and impact weights $(\mathbf{a}^{t,t+1}_{m,o}, \mathbf{d}^{t,t+1}_{m,o}, \mathbf{i}^{t,t+1}_{m,o})$. Between the feature maps $\mathbf{S}^t_m$ and $\mathbf{F}^{t+1}_o$, a high connection weight indicates that the group $\mathbb{G}^t_m$ is likely merged into the new group $\mathbb{G}^{t+1}_o$. We use *Part Grouping* to activate/deactivate the connections $\mathbb{C}^{t,t+1}$ in PGB. Eventually, after removing inactive connections, our framework constructs an effective part grouping pattern, and the PGB of the last stage outputs the visual feature map of all parts.

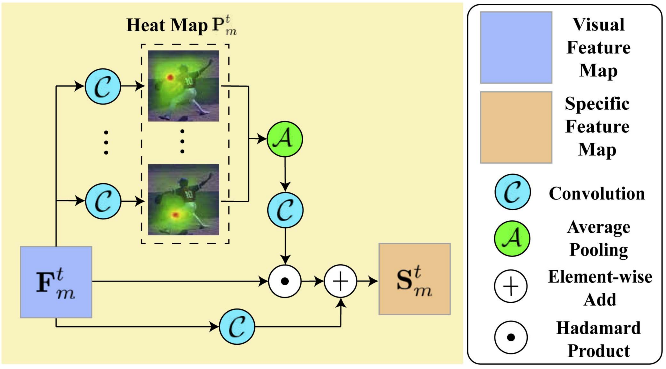


Fig. 4. In the group recognition block, we use the visual feature map and the heat map to compute the specific feature map.

B. Group Recognition Block

We illustrate GRB at the tth stage in Fig. 4. For the group $\mathbb{G}^t_m$, we use the separate convolutions to process the visual feature map $\mathbf{F}^t_m$, yielding the heat maps $\mathbf{P}^t_m$. Each convolution is associated with a single part detection in the part group $\mathbb{G}^t_m$. It produces a channel of $\mathbf{P}^t_m$, and each channel is a separate heat map indicating the localization of a single part. Given the heat maps $\mathbf{P}^t_m$ and the visual feature map $\mathbf{F}^t_m$, we compute the specific feature map $\mathbf{S}^t_m$ with the specific part information of $\mathbb{G}^t_m$ as:

$$\mathbf{S}^t_m = \mathcal{C}[\mathbf{F}^t_m] + \mathcal{C}[\mathcal{A}[\mathbf{P}^t_m]] \odot \mathbf{F}^t_m, \quad (1)$$

where $\mathcal{C}$, $\mathcal{A}$, and $\odot$ represent the convolution, the channel-wise average pooling, and the Hadamard product. The heat map $\mathbf{P}^t_m$

[2] To simplify the notations, we use unified group number at different stages.

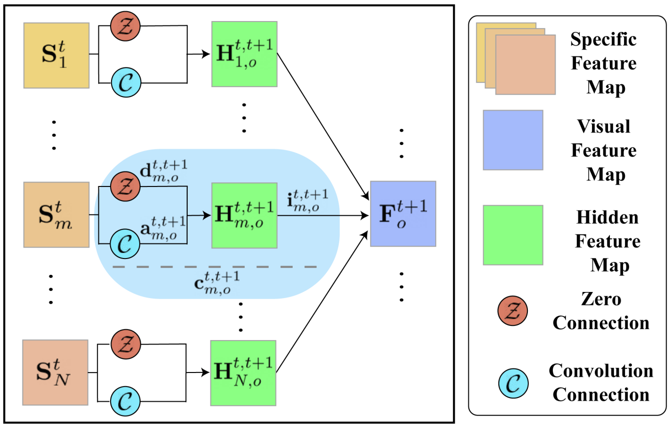


Fig. 5. The connection between the feature maps, which represent the part groups at adjacent stages. As highlighted by the cyan region, each connection consists of a hidden map and three weights (i.e., the activation, deactivation and impact weights).

plays a role in weighting the visual feature map $\mathbf{F}^t_m$. The locations, where the parts likely appear, have high heat values. Thus, the heat map can be regarded as a dynamic filter. It respects the prediction result and adaptively augments the information of the parts propagated from the visual feature map to the specific feature map.

C. Part Grouping Block

We illustrate the connections in the PGB between the tth and $(t+1)$th stages in Fig. 5, where we merge the specific feature maps and compute the visual feature maps of the new groups via *Part Grouping*. The cyan region of Fig. 5 indicates the connection between the feature maps $\mathbf{F}^t_m \in \mathbb{F}^t$ and $\mathbf{S}^{t+1}_o \in \mathbb{S}^{t+1}$. The connection is associated with the weight $\mathbf{c}^{t,t+1}_{m,o}$, which is computed as:

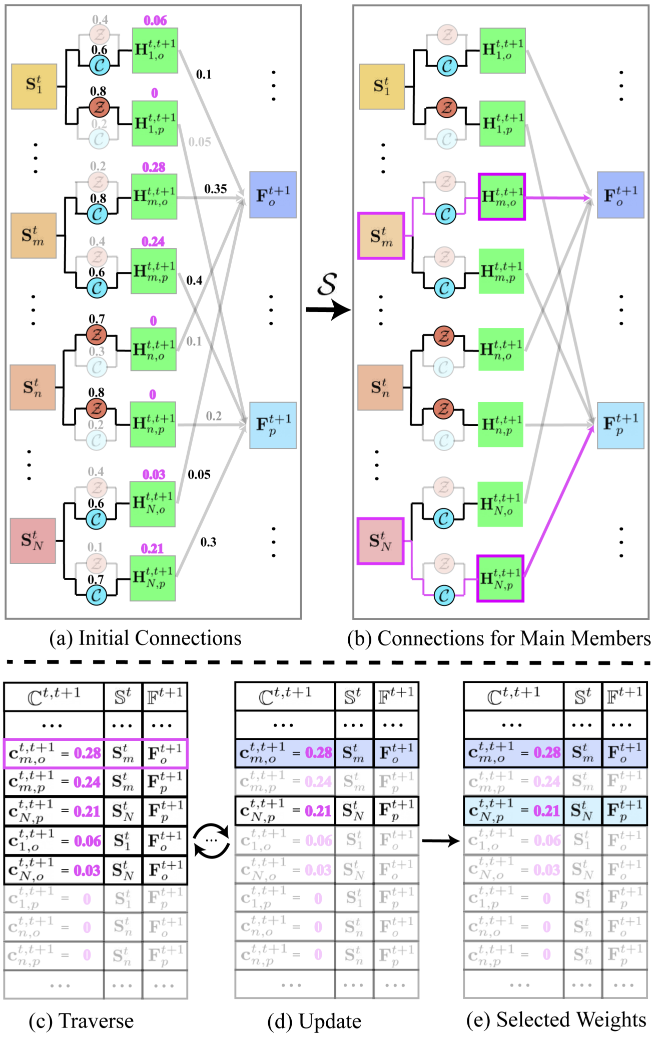
$$\mathbf{c}^{t,t+1}_{m,o} = \begin{cases} 0 & \mathbf{a}^{t,t+1}_{m,o} < \mathbf{d}^{t,t+1}_{m,o}, \\ \mathbf{a}^{t,t+1}_{m,o} \cdot \mathbf{i}^{t,t+1}_{m,o} & \text{otherwise}. \end{cases} \tag{2}$$

Given $\mathbf{a}^{t,t+1}_{m,o} < \mathbf{d}^{t,t+1}_{m,o}$, we deactivate $\mathbf{F}^t_m$ when we compute $\mathbf{F}^{t+1}_o$. We remark that all connection weights have been optimized before merging features. The connection is also associated with the hidden feature map $\mathbf{H}^{t,t+1}_{m,o} \in \mathbb{H}^{t,t+1}$. Based on the specific feature map $\mathbf{S}^t_m$ and the connection weight $\mathbf{c}^{t,t+1}_{m,o}$, we compute the hidden feature map $\mathbf{H}^{t,t+1}_{m,o}$ as:

$$\mathbf{H}^{t,t+1}_{m,o} = \mathbf{c}^{t,t+1}_{m,o} \cdot \mathcal{C}(\mathbf{S}^t_m). \tag{3}$$

Here, we let $\mathbb{H}^{t,t+1} = \{\mathbf{H}^{t,t+1}_{m,o} \in \mathbb{R}^{H\times W\times C} \,|\, m, o = 1, \ldots, N\}$ contain all of the hidden feature maps at the tth stage of *Part Grouping*.

D. Part Grouping



(a) Initial Connections (b) Connections for Main Members

$\mathbb{C}^{t,t+1}$	$\mathbb{S}^t$	$\mathbb{F}^{t+1}$
…	…	…
$\mathbf{c}^{t,t+1}_{m,o}$ = 0.28	$\mathbf{S}^t_m$	$\mathbf{F}^{t+1}_o$
$\mathbf{c}^{t,t+1}_{m,p}$ = 0.24	$\mathbf{S}^t_m$	$\mathbf{F}^{t+1}_p$
$\mathbf{c}^{t,t+1}_{N,p}$ = 0.21	$\mathbf{S}^t_N$	$\mathbf{F}^{t+1}_p$
$\mathbf{c}^{t,t+1}_{1,o}$ = 0.06	$\mathbf{S}^t_1$	$\mathbf{F}^{t+1}_o$
$\mathbf{c}^{t,t+1}_{N,o}$ = 0.03	$\mathbf{S}^t_N$	$\mathbf{F}^{t+1}_o$
$\mathbf{c}^{t,t+1}_{1,p}$ = 0	$\mathbf{S}^t_1$	$\mathbf{F}^{t+1}_p$
$\mathbf{c}^{t,t+1}_{n,o}$ = 0	$\mathbf{S}^t_n$	$\mathbf{F}^{t+1}_o$
$\mathbf{c}^{t,t+1}_{n,p}$ = 0	$\mathbf{S}^t_n$	$\mathbf{F}^{t+1}_p$
…	…	…

(c) Traverse

$\mathbb{C}^{t,t+1}$	$\mathbb{S}^t$	$\mathbb{F}^{t+1}$
…	…	…
$\mathbf{c}^{t,t+1}_{m,o}$ = 0.28	$\mathbf{S}^t_m$	$\mathbf{F}^{t+1}_o$
$\mathbf{c}^{t,t+1}_{m,p}$ = 0.24	$\mathbf{S}^t_m$	$\mathbf{F}^{t+1}_p$
$\mathbf{c}^{t,t+1}_{N,p}$ = 0.21	$\mathbf{S}^t_N$	$\mathbf{F}^{t+1}_p$
$\mathbf{c}^{t,t+1}_{1,o}$ = 0.06	$\mathbf{S}^t_1$	$\mathbf{F}^{t+1}_o$
$\mathbf{c}^{t,t+1}_{N,o}$ = 0.03	$\mathbf{S}^t_N$	$\mathbf{F}^{t+1}_o$
$\mathbf{c}^{t,t+1}_{1,p}$ = 0	$\mathbf{S}^t_1$	$\mathbf{F}^{t+1}_p$
$\mathbf{c}^{t,t+1}_{n,o}$ = 0	$\mathbf{S}^t_n$	$\mathbf{F}^{t+1}_o$
$\mathbf{c}^{t,t+1}_{n,p}$ = 0	$\mathbf{S}^t_n$	$\mathbf{F}^{t+1}_p$
…	…	…

(d) Update

$\mathbb{C}^{t,t+1}$	$\mathbb{S}^t$	$\mathbb{F}^{t+1}$
…	…	…
$\mathbf{c}^{t,t+1}_{m,o}$ = 0.28	$\mathbf{S}^t_m$	$\mathbf{F}^{t+1}_o$
$\mathbf{c}^{t,t+1}_{m,p}$ = 0.24	$\mathbf{S}^t_m$	$\mathbf{F}^{t+1}_p$
$\mathbf{c}^{t,t+1}_{N,p}$ = 0.21	$\mathbf{S}^t_N$	$\mathbf{F}^{t+1}_p$
$\mathbf{c}^{t,t+1}_{1,o}$ = 0.06	$\mathbf{S}^t_1$	$\mathbf{F}^{t+1}_o$
$\mathbf{c}^{t,t+1}_{N,o}$ = 0.03	$\mathbf{S}^t_N$	$\mathbf{F}^{t+1}_o$
$\mathbf{c}^{t,t+1}_{1,p}$ = 0	$\mathbf{S}^t_1$	$\mathbf{F}^{t+1}_p$
$\mathbf{c}^{t,t+1}_{n,o}$ = 0	$\mathbf{S}^t_n$	$\mathbf{F}^{t+1}_o$
$\mathbf{c}^{t,t+1}_{n,p}$ = 0	$\mathbf{S}^t_n$	$\mathbf{F}^{t+1}_p$
…	…	…

(e) Selected Weights

Fig. 6. Illustration of main group selection between the tth and $(t+1)$th stages. In (a), the operation $\mathcal{S}$ respects the non-zeros connection weights (see the weights in pink), activating the connections (see the arrows in pink) between the *main groups* and new groups in (b). In (c)–(d), we schematically illustrate the traverse and the update, which are iterated by the operation $\mathcal{S}$. In (e), the selected weights correspond to the activating connections in (b).

In the PGB, the gradient-based NAS [29] is employed to search the network architecture with the first order approximation. Instead of simply picking up the connections with highest values like traditional NAS methods, in *Part Grouping*, we use the main group selection $\mathcal{S}$, the normal group merging $\mathcal{M}$, the inactive group activation $\mathcal{I}$ to choose effective connection activation way and merge the features of different groups with the group computation $\mathcal{G}$ at each stage.

1) Main Group Selection: For each $\mathbb{G}^{t,t+1}$ in the $(t+1)$th stage, the main group selection $\mathcal{S}$ tries to find the part group from tth stage which has the largest impact (see Fig. 3(b)). Given the set $\mathbb{C}^{t,t+1}$ of the connection weights at the tth stage, $\mathcal{S}$ yields the weight set $\mathbb{M}^{t,t+1}$ as:

$$\mathbb{M}^{t,t+1} = \mathcal{S}(\mathbb{C}^{t,t+1}), \tag{4}$$

where $\mathbb{M}^{t,t+1}$ is a subset of $\mathbb{C}^{t,t+1}$ (i.e., $\mathbb{M}^{t,t+1} \subset \mathbb{C}^{t,t+1}$). It contains the weights of the network connections. Here, each connection (see the pink arrows illustrated in "Main Group Selection") uniquely indicates a main group of a new group.

We illustrate $\mathcal{S}$ in Fig. 6. In Fig. 6(a), we prepare the full connections between the feature maps in $\mathbb{S}^t$ and $\mathbb{F}^{t+1}$, respectively. We use the operation $\mathcal{S}$ to select the connections, which are highlighted by the pink arrows in Fig. 6(b). Here, Fig. 6(b) shows an example, where the mth and Nth groups at the tth stage play as the *main groups* of the oth and pth groups at the $(t+1)$th stage. To select the *main groups*, the operation $\mathcal{S}$ respects the connection weights in the set $\mathbb{C}^{t,t+1} = \{\mathbf{c}_{m,o}^{t,t+1} \in \mathbb{R} \,|\, m, o = 1, \ldots, N\}$. We provide more details of $\mathcal{S}$ in Fig. 6(c)–(e). First, we sort the connection weights in descending order, yielding the list in Fig. 6(c). The first column contains the sorted connection weights. The second and third columns represent the sets $\mathbb{S}^t$ and $\mathbb{F}^{t+1}$, where the feature maps are sorted along with the connection weights. Some of the weights are zeros, e.g., $\mathbf{c}_{1,p}^{t,t+1}$, $\mathbf{c}_{n,o}^{t,t+1}$, and $\mathbf{c}_{n,p}^{t,t+1}$ in last rows. This is because the corresponding connections, e.g., the connections between the feature maps $\mathbf{S}_1^t$ and $\mathbf{F}_p^{t+1}$, $\mathbf{S}_n^t$ and $\mathbf{F}_o^{t+1}$, $\mathbf{S}_n^t$ and $\mathbf{F}_p^{t+1}$ in Fig. 6(a), are deactivated by the zero connections. The zero weights are neglected by $\mathcal{S}$.

The sorted list is input to the iteration between traverse and update. The traverse is top-down, allowing the connections with relatively large weights to be selected. The selected weight belongs to the *main group*, whose connection to the new group is activated. In Fig. 6(c), we show an example, where $\mathbf{c}_{m,o}^{t,t+1}$ is selected. It means that the mth group is the *main group* of the oth new group. We let $\mathbf{c}_{m,o}^{t,t+1} \in \mathbb{M}^{t,t+1}$. After selecting a weight, we input the list to the update process in Fig. 6(d). Some of the weights, e.g., $\mathbf{c}_{m,p}^{t,t+1}$, $\mathbf{c}_{1,o}^{t,t+1}$, and $\mathbf{c}_{N,o}^{t,t+1}$, are associated with the traversed *main groups* and the corresponding new groups. These weights are neglected, when the list is fed backward to the traverse process in Fig. 6(d). Given that all of the non-zero weights are traversed, we achieve the list in Fig. 6(e), where the selected weights belong to the set $\mathbb{M}^{t,t+1}$.

Generally, for the feature map $\mathbf{F}_o^{t+1}$, $\mathcal{S}$ select the feature map $\mathbf{S}_\star^t$ whose associated connection weights $\mathbf{c}_{\star,o}^{t,t+1}$ is the highest in the set $\{\mathbf{c}_{m,o}^{t,t+1} \,|\, m \in \{1, \ldots, N\}\}$. We can formulate operation $\mathcal{S}$ as:

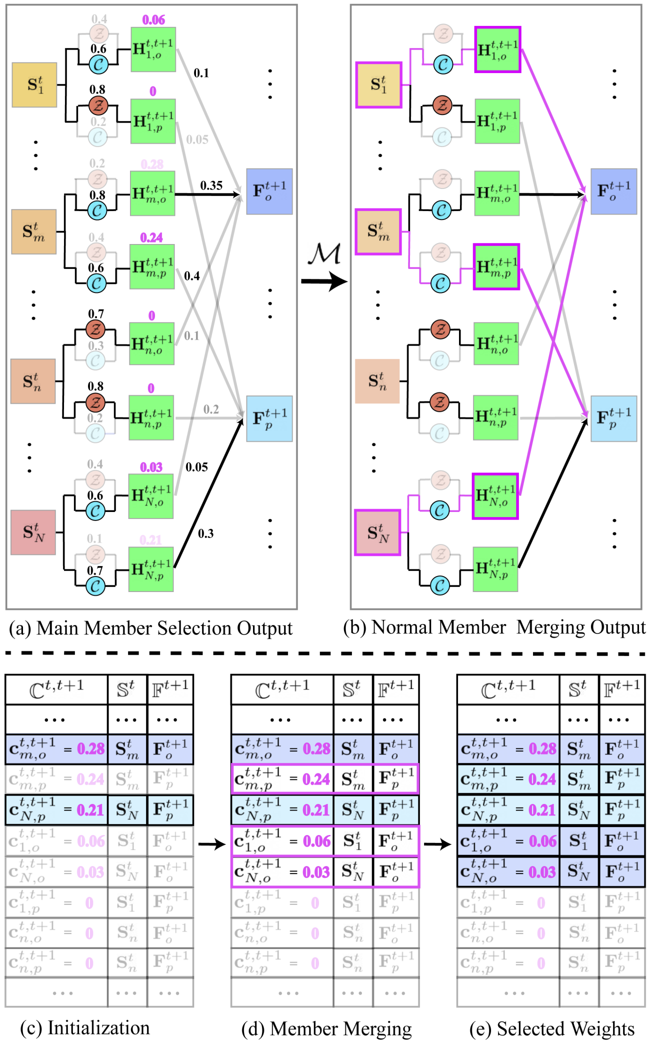
$$\mathbf{c}_{\star,o}^{t,t+1} = \underset{m\in\{1,\ldots,N\}}{\operatorname{argmax}} \; \mathbf{c}_{m,o}^{t,t+1}, \;\; \mathbb{G}_\star^t \subset \mathbb{G}_o^{t+1}. \tag{5}$$

The feature map $\mathbf{S}_\star^t$ represents the *main group* included into the new group $\mathbb{G}_o^{t+1}$ and the weight set $\mathbb{M}^{t,t+1} = \{\mathbf{c}_{\star,o}^{t,t+1} \,|\, o \in \{1, \ldots, N\}\}$. We remark that each new group contains a *main group*, which has larger connection weights than other groups.

2) Normal Group Merging: As formulated in (6), we subtract the set $\mathbb{M}^{t,t+1}$ from the entire set $\mathbb{C}^{t,t+1}$ of the connection weights. The rest of the connection weights are input to the operation $\mathcal{M}$ of the normal group merging (see Fig. 3(c)). $\mathcal{M}$ produces the weight set $\mathbb{N}^{t,t+1}$ as:

$$\mathbb{N}^{t,t+1} = \mathcal{M}(\mathbb{C}^{t,t+1} - \mathbb{M}^{t,t+1}), \tag{6}$$

where $\mathbb{N}^{t,t+1} \subset \mathbb{C}^{t,t+1} - \mathbb{M}^{t,t+1}$. The set $\mathbb{N}^{t,t+1}$ also contains the connection weights (see the pink arrows). We use these weights to determine the normal groups in the new groups. It should be noted that the normal groups have smaller connection weights than the main groups. We illustrate M in Fig. 7. In Fig. 7(a), we show the activated connections between the feature maps of the *main groups* and the new groups (see the connections highlighted by the black arrows). Except these connections, we employ the operation $\mathcal{M}$ to activate the connections of the *normal groups* in Fig. 7(b). In Fig. 7(c)–(e), we detail how the operation $\mathcal{M}$ works. In Fig. 7(c), we illustrate the weights in the set $\mathbb{M}^{t,t+1}$, where the connections of these weights have been activated in $\mathcal{S}$. We exclude these weights from the set $\mathbb{C}^{t,t+1}$. The operation $\mathcal{M}$ selects the non-zero weights in Fig. 7(d), where these weights are associated with the connections (see the connections highlighted by the pink arrows) between the *normal groups* and the new groups. Then, we can obtain the list in Fig. 7(e).



(a) Main Member Selection Output

(b) Normal Member Merging Output

$\mathbb{C}^{t,t+1}$	$\mathbb{S}^t$	$\mathbb{F}^{t+1}$
⋯	⋯	⋯
$\mathbf{c}_{m,o}^{t,t+1}$ = 0.28	$\mathbf{S}_m^t$	$\mathbf{F}_o^{t+1}$
$\mathbf{c}_{m,p}^{t,t+1}$ = 0.24	$\mathbf{S}_m^t$	$\mathbf{F}_p^{t+1}$
$\mathbf{c}_{N,p}^{t,t+1}$ = 0.21	$\mathbf{S}_N^t$	$\mathbf{F}_p^{t+1}$
$\mathbf{c}_{1,o}^{t,t+1}$ = 0.06	$\mathbf{S}_1^t$	$\mathbf{F}_o^{t+1}$
$\mathbf{c}_{N,o}^{t,t+1}$ = 0.03	$\mathbf{S}_N^t$	$\mathbf{F}_o^{t+1}$
$\mathbf{c}_{1,p}^{t,t+1}$ = 0	$\mathbf{S}_1^t$	$\mathbf{F}_p^{t+1}$
$\mathbf{c}_{n,o}^{t,t+1}$ = 0	$\mathbf{S}_n^t$	$\mathbf{F}_o^{t+1}$
$\mathbf{c}_{n,p}^{t,t+1}$ = 0	$\mathbf{S}_n^t$	$\mathbf{F}_p^{t+1}$
⋯	⋯	⋯

(c) Initialization

$\mathbb{C}^{t,t+1}$	$\mathbb{S}^t$	$\mathbb{F}^{t+1}$
⋯	⋯	⋯
$\mathbf{c}_{m,o}^{t,t+1}$ = 0.28	$\mathbf{S}_m^t$	$\mathbf{F}_o^{t+1}$
$\mathbf{c}_{m,p}^{t,t+1}$ = 0.24	$\mathbf{S}_m^t$	$\mathbf{F}_p^{t+1}$
$\mathbf{c}_{N,p}^{t,t+1}$ = 0.21	$\mathbf{S}_N^t$	$\mathbf{F}_p^{t+1}$
$\mathbf{c}_{1,o}^{t,t+1}$ = 0.06	$\mathbf{S}_1^t$	$\mathbf{F}_o^{t+1}$
$\mathbf{c}_{N,o}^{t,t+1}$ = 0.03	$\mathbf{S}_N^t$	$\mathbf{F}_o^{t+1}$
$\mathbf{c}_{1,p}^{t,t+1}$ = 0	$\mathbf{S}_1^t$	$\mathbf{F}_p^{t+1}$
$\mathbf{c}_{n,o}^{t,t+1}$ = 0	$\mathbf{S}_n^t$	$\mathbf{F}_o^{t+1}$
$\mathbf{c}_{n,p}^{t,t+1}$ = 0	$\mathbf{S}_n^t$	$\mathbf{F}_p^{t+1}$
⋯	⋯	⋯

(d) Member Merging

$\mathbb{C}^{t,t+1}$	$\mathbb{S}^t$	$\mathbb{F}^{t+1}$
⋯	⋯	⋯
$\mathbf{c}_{m,o}^{t,t+1}$ = 0.28	$\mathbf{S}_m^t$	$\mathbf{F}_o^{t+1}$
$\mathbf{c}_{m,p}^{t,t+1}$ = 0.24	$\mathbf{S}_m^t$	$\mathbf{F}_p^{t+1}$
$\mathbf{c}_{N,p}^{t,t+1}$ = 0.21	$\mathbf{S}_N^t$	$\mathbf{F}_p^{t+1}$
$\mathbf{c}_{1,o}^{t,t+1}$ = 0.06	$\mathbf{S}_1^t$	$\mathbf{F}_o^{t+1}$
$\mathbf{c}_{N,o}^{t,t+1}$ = 0.03	$\mathbf{S}_N^t$	$\mathbf{F}_o^{t+1}$
$\mathbf{c}_{1,p}^{t,t+1}$ = 0	$\mathbf{S}_1^t$	$\mathbf{F}_p^{t+1}$
$\mathbf{c}_{n,o}^{t,t+1}$ = 0	$\mathbf{S}_n^t$	$\mathbf{F}_o^{t+1}$
$\mathbf{c}_{n,p}^{t,t+1}$ = 0	$\mathbf{S}_n^t$	$\mathbf{F}_p^{t+1}$
⋯	⋯	⋯

(e) Selected Weights

Fig. 7. Illustration of normal group merging between the tth and $(t+1)$th stages. In (a), the operation $\mathcal{M}$ selects the retaining non-zeros connection weights (see the weights in pink), activating the connections (see the arrows in pink) between the *normal groups* and the new groups in (b). In (c)–(d), we illustrate the process of merging, which are decided by the operation $\mathcal{M}$. In (e), the selected weights correspond to the activating connections in (b).

For the feature map $\mathbf{F}_o^{t+1}$, $\mathcal{M}$ select the feature maps $\mathbf{S}_\dagger^t$ whose associated connection weights $\mathbf{c}_{\dagger,o}^{t,t+1}$ are not 0, and we can formulate $\mathcal{M}$ as:

$$\mathbb{G}_\dagger^t \subset \mathbb{G}_o^{t+1}, \;\; s.t. \; \mathbf{c}_{\dagger,o}^{t,t+1} \neq 0. \tag{7}$$

The feature map $\mathbf{S}^t_\dagger$ represents the *main group* included into the new group $\mathbb{G}^{t+1}_o$ and the weight set $\mathbb{N}^{t,t+1} = \{\mathbf{c}^{t,t+1}_{\dagger,o} \,|\, o \in \{1,\ldots,N\}\}$.

3) Inactive Group Activation: Possibly, any new group in the set $\{\mathbb{G}^{t+1}_1,\ldots,\mathbb{G}^{t+1}_N\}$ contains little or even no parts. It means in the subsequent stages, the feature learning of these parts will be missed. These parts let their groups in the set $\{\mathbb{G}^t_1,\ldots,\mathbb{G}^t_N\}$ be the *inactive groups*. For the *inactive groups*, the weights of the zero connections are larger than the convolution connections, where the zero connections may eliminate the information of the *inactive groups*. To avoid the information loss of these parts, we leverage the inactive group activation $\mathcal{I}$ to re-activate those *inactive groups* and pass them to the next stage. The proposed inactive group activation $\mathcal{I}$ excludes the connection weights of the main and normal groups from the set $\mathbb{C}^{t,t+1}$ (see Fig. 3(d)). We use $\mathcal{I}$ to identify the inactive groups. $\mathcal{I}$ chooses some of the inactive groups and activates the associated connections (see the activating arrow in pink). It produces the weight set $\mathbb{I}^{t,t+1}$ as:

$$\mathbb{I}^{t,t+1} = \mathcal{I}\left(\mathbb{C}^{t,t+1} - \mathbb{M}^{t,t+1} - \mathbb{N}^{t,t+1}\right), \qquad (8)$$

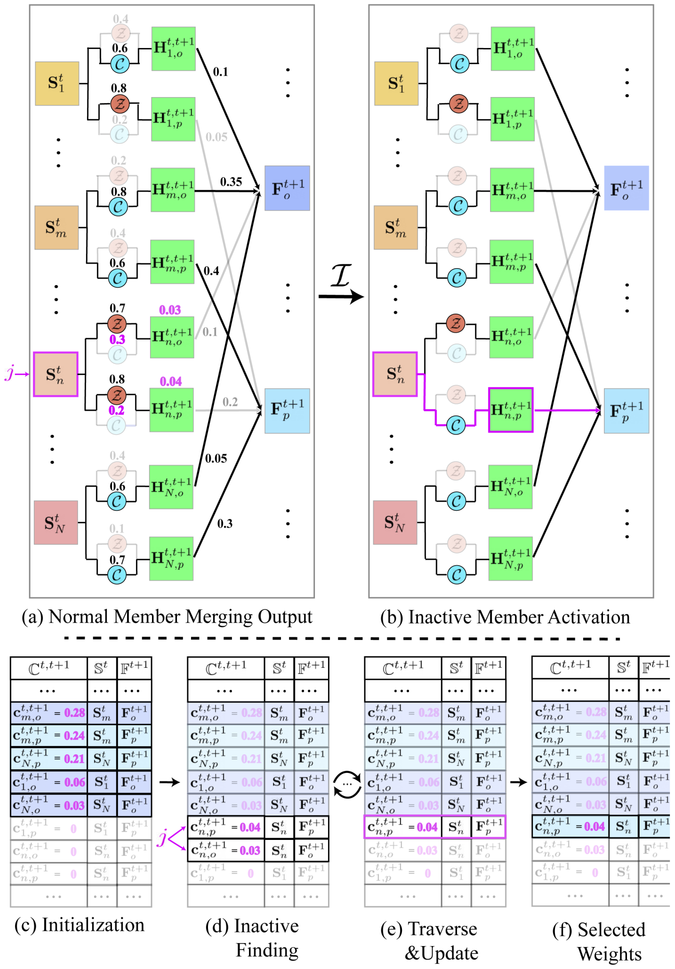
where $\mathbb{I}^{t,t+1} \subset \mathbb{C}^{t,t+1} - \mathbb{M}^{t,t+1} - \mathbb{N}^{t,t+1}$.

We illustrate the operation $\mathcal{I}$ in Fig. 8. Assuming that the jth body part is missed in $(t+1)$th stage and G^t_n is an *inactive group* containing j, in Fig. 8(a), we show the connection weights of the *inactive group*. And we use operation $\mathcal{I}$ to select the connections, which are highlighted by the pink arrows in Fig. 8(b). In Fig. 8(c)–(f), we provide more details of operation $\mathcal{I}$. Taking the result of $\mathcal{M}$ as initialization in Fig. 8(c), first, we exclude those activated connections in $\mathcal{S}$ and $\mathcal{M}$. Then we search the set $\{\mathbb{G}^t_1,\ldots,\mathbb{G}^t_N\}$ to find the *inactive groups* which contains part j. Here we find G^t_n. Including the connections of these *inactive groups* and sorting the connection weights in descending order, we obtain the result in Fig. 8(d). Then we traverse these connections of *inactive groups* and select connections with the largest weights. Fig. 8(e) gives an example, where $c^{t,t+1}_{n,p}$ is selected. It means that connection $c^{t,t+1}_{n,p}$ is re-activated and the G^t_n is merged to G^{t+1}_p. After traversing all missed parts and update the connections of them, we achieve the result in Fig. 8(f), where the selected weights belong to the set $\mathbb{I}^{t,t+1}$.

Generally, in operation $\mathcal{I}$, we select the *inactive group* with the largest impact to the new group in the set $\{\mathbb{G}^{t+1}_1,\ldots,\mathbb{G}^{t+1}_N\}$. For the missed jth part, $\mathcal{I}$ firstly search the subset $\mathbb{J} = \{\mathbb{G}^t_n \,|\, j \in \mathbb{G}^t_n, n \in \{1,\ldots,N\}\}$ whose each element $\mathbb{G}^t_n$ contains jth part, then $\mathcal{A}$ select the feature map $S^t_\ddagger$ whose associated connection weights $\mathbf{c}^{t,t+1}_{\ddagger,*}$ is the highest in the set $\{\mathbf{c}^{t,t+1}_{x,y} \,|\, \mathbb{G}^t_x \in \mathbb{J}, y \in \{1,\ldots,N\}\}$. Hence, $\mathcal{I}$ can be formulated as:

$$\mathbf{c}^{t,t+1}_{\ddagger,*} = \underset{\mathbb{G}_x \in \mathbb{J}, y \in \{1,\ldots,N\}}{\operatorname{argmax}} \mathbf{c}^{t,t+1}_{x,y},$$
$$\mathbb{G}^t_\ddagger \subset \mathbb{G}^{t+1}_*, \ \ s.t.\ j \in \mathbb{G}^t_\ddagger. \qquad (9)$$

Equation (9) enables the activation on the inactive group $\mathbb{G}^t_\ddagger$, where a relative large weight of the convolution connection helps to propagate more information of the part j to the oth new group. The weight set $\mathbb{I}^{t,t+1} = \{\mathbf{c}^{t,t+1}_{\ddagger,*}\}$. For the simplification, we will regard the $*$ as o in the following description.


0.4
0.6
0.8
0.2
0.7
0.3
0.03
0.04
0.1
0.05
0.35
0.4
0.1
0.2
0.05
0.3
$\mathbf{S}^t_1$
$\mathbf{S}^t_m$
$\mathbf{S}^t_n$
$\mathbf{S}^t_N$
j
$\mathcal{Z}$
$\mathcal{C}$
$\mathbf{H}^{t,t+1}_{1,o}$
$\mathbf{H}^{t,t+1}_{1,p}$
$\mathbf{H}^{t,t+1}_{m,o}$
$\mathbf{H}^{t,t+1}_{m,p}$
$\mathbf{H}^{t,t+1}_{n,o}$
$\mathbf{H}^{t,t+1}_{n,p}$
$\mathbf{H}^{t,t+1}_{N,o}$
$\mathbf{H}^{t,t+1}_{N,p}$
$\mathbf{F}^{t+1}_o$
$\mathbf{F}^{t+1}_p$
$\mathcal{I}$

(a) Normal Member Merging Output

(b) Inactive Member Activation

$\mathbb{C}^{t,t+1}$	$\mathbb{S}^t$	$\mathbb{F}^{t+1}$
…	…	…
$\mathbf{c}^{t,t+1}_{m,o}$ = 0.28	$\mathbf{S}^t_m$	$\mathbf{F}^{t+1}_o$
$\mathbf{c}^{t,t+1}_{m,p}$ = 0.24	$\mathbf{S}^t_m$	$\mathbf{F}^{t+1}_p$
$\mathbf{c}^{t,t+1}_{N,p}$ = 0.21	$\mathbf{S}^t_N$	$\mathbf{F}^{t+1}_p$
$\mathbf{c}^{t,t+1}_{1,o}$ = 0.06	$\mathbf{S}^t_1$	$\mathbf{F}^{t+1}_o$
$\mathbf{c}^{t,t+1}_{N,o}$ = 0.03	$\mathbf{S}^t_N$	$\mathbf{F}^{t+1}_o$
$\mathbf{c}^{t,t+1}_{1,p}$ = 0	$\mathbf{S}^t_1$	$\mathbf{F}^{t+1}_p$
$\mathbf{c}^{t,t+1}_{n,o}$ = 0	$\mathbf{S}^t_n$	$\mathbf{F}^{t+1}_o$
$\mathbf{c}^{t,t+1}_{n,p}$ = 0	$\mathbf{S}^t_n$	$\mathbf{F}^{t+1}_p$
…	…	…

(c) Initialization

$\mathbb{C}^{t,t+1}$	$\mathbb{S}^t$	$\mathbb{F}^{t+1}$
…	…	…
$\mathbf{c}^{t,t+1}_{m,o}$ = 0.28	$\mathbf{S}^t_m$	$\mathbf{F}^{t+1}_o$
$\mathbf{c}^{t,t+1}_{m,p}$ = 0.24	$\mathbf{S}^t_m$	$\mathbf{F}^{t+1}_p$
$\mathbf{c}^{t,t+1}_{N,p}$ = 0.21	$\mathbf{S}^t_N$	$\mathbf{F}^{t+1}_p$
$\mathbf{c}^{t,t+1}_{1,o}$ = 0.06	$\mathbf{S}^t_1$	$\mathbf{F}^{t+1}_o$
$\mathbf{c}^{t,t+1}_{N,o}$ = 0.03	$\mathbf{S}^t_N$	$\mathbf{F}^{t+1}_o$
$\mathbf{c}^{t,t+1}_{n,p}$ = 0.04	$\mathbf{S}^t_n$	$\mathbf{F}^{t+1}_p$
$\mathbf{c}^{t,t+1}_{n,o}$ = 0.03	$\mathbf{S}^t_n$	$\mathbf{F}^{t+1}_o$
$\mathbf{c}^{t,t+1}_{1,p}$ = 0	$\mathbf{S}^t_1$	$\mathbf{F}^{t+1}_p$
…	…	…

(d) Inactive Finding

$\mathbb{C}^{t,t+1}$	$\mathbb{S}^t$	$\mathbb{F}^{t+1}$
…	…	…
$\mathbf{c}^{t,t+1}_{m,o}$ = 0.28	$\mathbf{S}^t_m$	$\mathbf{F}^{t+1}_o$
$\mathbf{c}^{t,t+1}_{m,p}$ = 0.24	$\mathbf{S}^t_m$	$\mathbf{F}^{t+1}_p$
$\mathbf{c}^{t,t+1}_{N,p}$ = 0.21	$\mathbf{S}^t_N$	$\mathbf{F}^{t+1}_p$
$\mathbf{c}^{t,t+1}_{1,o}$ = 0.06	$\mathbf{S}^t_1$	$\mathbf{F}^{t+1}_o$
$\mathbf{c}^{t,t+1}_{N,o}$ = 0.03	$\mathbf{S}^t_N$	$\mathbf{F}^{t+1}_o$
$\mathbf{c}^{t,t+1}_{n,p}$ = 0.04	$\mathbf{S}^t_n$	$\mathbf{F}^{t+1}_p$
$\mathbf{c}^{t,t+1}_{n,o}$ = 0.03	$\mathbf{S}^t_n$	$\mathbf{F}^{t+1}_o$
$\mathbf{c}^{t,t+1}_{1,p}$ = 0	$\mathbf{S}^t_1$	$\mathbf{F}^{t+1}_p$
…	…	…

(e) Traverse &Update

$\mathbb{C}^{t,t+1}$	$\mathbb{S}^t$	$\mathbb{F}^{t+1}$
…	…	…
$\mathbf{c}^{t,t+1}_{m,o}$ = 0.28	$\mathbf{S}^t_m$	$\mathbf{F}^{t+1}_o$
$\mathbf{c}^{t,t+1}_{m,p}$ = 0.24	$\mathbf{S}^t_m$	$\mathbf{F}^{t+1}_p$
$\mathbf{c}^{t,t+1}_{N,p}$ = 0.21	$\mathbf{S}^t_N$	$\mathbf{F}^{t+1}_p$
$\mathbf{c}^{t,t+1}_{1,o}$ = 0.06	$\mathbf{S}^t_1$	$\mathbf{F}^{t+1}_o$
$\mathbf{c}^{t,t+1}_{N,o}$ = 0.03	$\mathbf{S}^t_N$	$\mathbf{F}^{t+1}_o$
$\mathbf{c}^{t,t+1}_{n,p}$ = 0.04	$\mathbf{S}^t_n$	$\mathbf{F}^{t+1}_p$
$\mathbf{c}^{t,t+1}_{n,o}$ = 0.03	$\mathbf{S}^t_n$	$\mathbf{F}^{t+1}_o$
$\mathbf{c}^{t,t+1}_{1,p}$ = 0	$\mathbf{S}^t_1$	$\mathbf{F}^{t+1}_p$
…	…	…

(f) Selected Weights

Fig. 8. Illustration of inactive group activation between the tth and $(t+1)$th stages. In (a), the missed part j is in F^t_n, and operation $\mathcal{I}$ respects the non-zeros connection weights(see the weights in pink), activating the connections (see the arrows in pink) between the *inactive group* and the new groups in (b). In (c)–(e), we illustrate the process of the inactive finding, the traverse and the update which are decided by the operation $\mathcal{I}$. In (f), the selected weights correspond to the activating connections in (b).

4) Group Feature Computation: The group feature computation $\mathcal{G}$ use all the active connections. As shown in Fig. 3(e), we use the connections (see the activated arrows in black) that are associated with the weights in the sets $\mathbb{M}^{t,t+1}$, $\mathbb{N}^{t,t+1}$, and $\mathbb{I}^{t,t+1}$, along with the feature maps in $\mathbb{S}^t$ and the heat maps in $\mathbb{H}^t$, to compute the feature maps in the set $\mathbb{F}^{t+1}$ and the heat maps in the set $\mathbb{H}^{t+1}$ as:

$$\mathbb{F}^{t+1} = \mathcal{G}\left(\mathbb{M}^{t,t+1}, \mathbb{N}^{t,t+1}, \mathbb{I}^{t,t+1}, \mathbb{S}^t, \mathbb{H}^t\right). \qquad (10)$$

Specifically, we merge the hidden feature maps of the *main*, *normal*, and *inactive groups* to compute the visual feature map of each new group. For the feature map $\mathbf{F}^{t+1}_o$, we compute it as:

$$\mathbf{F}^{t+1}_o = \mathbf{H}^{t,t+1}_{\star,o} + \sum_{\mathbb{G}^t_\dagger \subset \mathbb{G}^{t+1}_o} \mathbf{H}^{t,t+1}_{\dagger,o} + \sum_{\mathbb{G}^t_\ddagger \subset \mathbb{G}^{t+1}_o} \mathbf{H}^{t,t+1}_{\ddagger,o}. \qquad (11)$$

In (11), the hidden feature maps $\mathbf{H}^{t,t+1}_{\star,o}$, $\mathbf{H}^{t,t+1}_{\dagger,o}$, and $\mathbf{H}^{t,t+1}_{\ddagger,o}$ are associated with the *main*, *normal* and *inactive groups* of the new group $\mathbb{G}_o^{t+1}$. We use the set of visual feature maps $\{\mathbf{F}_o^{t+1} | o = 1, \ldots, N\}$ as the specific feature maps $\{\mathbf{S}_o^{t+1} | o = 1, \ldots, N\}$ at the $(t+1)$th stage of part grouping.

E. Network Optimization

Similar to previous NAS methods [29], our network optimization also consists of searching and fine-tuning. We employ the *Part Grouping* to obtain new connection activation in each iteration of network optimization. And we use the searched results (activation) and to optimize all weights (including searchable connection weights $\mathbb{C}$ and non-searchable network weights) in the next optimization iteration. To supervise the part detection and specific information extraction of part group, we propose a novel objective consisting of several parts.

First, the heat map representation loss is used to supervise the accuracy of part detection. We input the visual feature map $\mathbf{F}_o^{t+1}$ into GRB, predicting the heat map $\mathbf{P}_o^{t+1}$, whose difference with the ground-truth heat map $\widehat{\mathbf{P}}_o^{t+1}$ is computed as:

$$\mathcal{L}_h = \sum_{t=0}^{T-1} \sum_{m=1}^{N} ||\mathbf{P}_o^{t+1} - \widehat{\mathbf{P}}_o^{t+1}||_2^2, \tag{12}$$

where $\mathcal{L}_h$ measures the localization error of parts. We minimize $\mathcal{L}_h$ during the network searching.

Besides the traditional heat map representation supervision, we explore the supervision at feature level to promote specific feature learning of body parts. We encourage different *main groups* to capture the diverse visual patterns of the part groups. It helps to reduce the redundant information of the feature maps that represent the *main groups*. For this purpose, we employ the cosine similarity between the feature maps of the *main groups*. The similarity is accumulated as:

$$\mathcal{L}_s = \sum_{\mathbf{c}_{m,o}^{t,t+1}, \mathbf{c}_{n,p}^{t,t+1} \in \mathbb{M}^{t,t+1}} \cos\left[\mathbf{H}_{m,o}^{t,t+1}, \mathbf{H}_{n,p}^{t,t+1}\right], \tag{13}$$

where we use (3) to compute the hidden feature maps (e.g., $\mathbf{H}_{m,o}^{t,t+1}$ and $\mathbf{H}_{n,p}^{t,t+1}$). We minimize $\mathcal{L}_s$ during the network searching.

Moreover, we encourage the *normal groups* and the *inactive groups* to have a strong correlation with the *main group*. It allows the merged groups to form relevant information. For the tth stage, we use the dot-product between the hidden feature maps to measure the group correlation as:

$$\mathcal{L}_c = \sum_{o=1}^{N} \left(\sum_{\mathbb{G}_\star^t, \mathbb{G}_\dagger^t \subset \mathbb{G}_o^{t+1}} \mathbf{H}_{\star,o}^{t,t+1} \cdot \mathbf{H}_{\dagger,o}^{t,t+1} \right.$$
$$\left. + \sum_{\mathbb{G}_\star^t, \mathbb{G}_\ddagger^t \subset \mathbb{G}_o^{t+1}} \mathbf{H}_{\star,o}^{t,t+1} \cdot \mathbf{H}_{\ddagger,o}^{t,t+1} \right), \tag{14}$$

where the correlation is maximized during searching.

We allow the result of part grouping to guide the network weight optimization. Thus, we use (12), (13), and (14) to formulate the overall searching objective as:

$$\mathcal{L} = \mathcal{L}_h + \alpha \cdot \mathcal{L}_s - \beta \cdot \mathcal{L}_c, \tag{15}$$

where α and β serve as hyper-parameters in our network to balance the ratio of different terms. Empirically, we set $\alpha = 1$ and $\beta = 1$. And, the further explanation for the settings of hyperparameters are in Section V-B5. During network searching, we minimize $\mathcal{L}$ to optimize connections.

When the end condition of searching is met, we prune all the inactivated connections with the zero hidden feature map in-between and fix the network architecture. Previous methods mostly set the convergence for optimization as the searching end condition [29]. While unaware of this, in practice, we set a maximum iteration number of optimization for each stage, like [40]. Once the optimization iteration number reaches the preset maximum, the weight optimization and part grouping in the corresponding stage are done. We set the end iteration number for each stage, increasing along with the stage number. Then, the search for the previous stages can stop early, and we obtain this stage's fixed part grouping pattern. This can make the *Part Grouping* in the following stages more specific and easy to extract the high-level correlation information between learnt part groups. Given the fixed network architecture, we follow the convention to use the localization error $\mathcal{L}_h$ (see (12)) for further fine-tuning the learnable parameters, which are used for human skeleton establishment eventually. Fig. 3 describes the brief process of the optimization.

V. Experimental Results

A. Experimental Settings

1) Implementation Details: Accurate-PGNet is implemented based on PyTorch [41]. We use an Nvidia GeForce RTX™ 3090 GPU with 24GB memory to train and test it. We set the maximum training epochs to 270. The explanation of the setting of the training iterations is shown in Section V-B6. We use the Adam solver [42] with the initial learning rate of 1e-4 for optimizing the network parameters. The learning rates are decayed linearly. After the pruning, we set the learning rate to 3e-3 for fine-tuning. Each mini-batch contains 12 images. The resolution of feature maps in PGB and GRB is set according to the resolutions of input images for different backbone networks. The maximum iteration number of searching for the first stage is 30 epochs. Moreover, the maximum number of the following stages will increase by 15 epochs for each stage, with the number increasing.

2) Dataset: We use the **MPII** dataset [19] for the major evaluation. This dataset contains 25K images extracted from YouTube videos. About 40K instances of humans are cropped from the images to produce 28K, 3K, and 11K images in the training, validation, and test sets. 16 parts annotate the human body. The split of the *train/validation/test* datasets is the same as in [19]. We also report the results on the **MSCOCO** Keypoint dataset [20]. The *train*, *val*, and *test-dev* sets contain 57K, 5K, and 20K images. 17 parts annotate each person's instance. We use the OCHuman [55] dataset for evaluation to measure the performance of different models in dealing with occluded persons. OCHuman contains human instances with heavy occlusions and is only used for evaluation. It consists of 4K images and 8K instances. We conduct architecture searches on **MPII**, **MSCOCO**,

TABLE I
SENSITIVITY TO THE NUMBER OF GROUPS

Groups M	Params	FLOPs	PCKh@0.5
0	**28.9M**	**10.2G**	90.4
2	29.3M	11.3G	90.5
4	29.7M	14.0G	**91.7**
8	30.8M	16.1G	91.4
16	34.8M	33.1G	91.3

We list the results on the MPII val set.

TABLE II
SENSITIVITY TO THE NUMBER OF STAGES

Stages T	Params	FLOPs	PCKh@0.5
0	**28.9M**	**10.2G**	90.4
1	29.3M	11.5G	90.6
2	29.7M	12.7G	91.5
3	29.8M	13.1G	**91.7**
4	29.9M	13.4G	91.3
5	30.0M	13.6G	91.0

We list the results on the MPII val set.

and **OCHuman**, respectively. Besides, to evaluate robustness, we construct **COCO-C** dataset based on the validation set of **MSCOCO** by applying different corruptions. In this paper, we focus on the influence of noise and lighting. For the noise, we apply three common types of corruption: Motion Blur, Gaussian Noise, and Impulse Noise. As for lighting, we simulate various lighting conditions using three types of corruptions: Brightness, Darkness, and Contrast.

3) Evaluation Metrics: We use two metrics for the evaluation of HPE models, called percentage of correct keypoints (PCK) and object keypoint similarity (OKS). PCK is the proportion of correct keypoints estimated. It is to calculate the ratio of the normalized distance between the real part locations and corresponding predicted ones less than the given threshold. Normally, the length of human head is set as the normalized reference. And this kind of metric is called PCKh. We report the performance of our approach on the MPII validation and test sets, in terms of PCKh. A lot of datasets adopt OKS as the evaluation index. It calculates the similarities of groundtruth and predicted part positions of human body. For a person instance p, the OKS can be defined as:

$$OKS_p = \frac{\sum_i \exp\{-d_{p^i}^2/2S_p^2\sigma_i^2\}\delta(v_{p^{i=1}})}{\sum_i \delta(v_{p^{i=1}})}, \qquad (16)$$

where d_{p^i} is the Euclidean distance between the predicted and ground-truth part positions, s_p represents the scale of the human instance, v_{p^i} is the visibility label of the human body part, and σ_i defines the offset of the artificially labeled location. Based on the OKS between the predicted and the ground-truth parts, given OKS threshold, we compute the average precision (AP) and the average recall (AR) on the COCO *val* and *test-dev* sets. For the evaluation of model robustness, we use mean average precision (mAP) and mean average recall (mAR). We report mAP as the average of AP values at thresholds ranging from 0.5 to 0.95, with a step size of 0.05. mAR is similar. Following [43], we introduce the robustness metric, mean Relative Robustness (mRR), to evaluate how much a model's performance drops under certain corruptions compared to clean images. In the following, in Tables I to XIII, the best results are highlighted in bold.

B. Ablation Study and Discussion

We conduct ablation experiments on the **MPII** dataset, and the models are basically searched based on HRNet.

TABLE III
SENSITIVITY TO THE BACKBONE DCNS

Backbone	Grouping	Params	FLOPs	PCKh@0.5
ResNet-50 [21]	×	**24.5M**	**5.2G**	87.0
	✓	26.1M	10.4G	**88.6**
MobileNet [44]	×	**1.3M**	**0.4G**	85.4
	✓	2.2M	3.5G	**86.2**
HRNet-W32-S3 [3]	×	**8.1M**	**5.8G**	89.8
	✓	8.9M	8.6G	**90.6**
HRNet-W32-S4 [3]	×	**28.5M**	**9.5G**	90.3
	✓	29.8M	13.1G	**91.7**

We report the results on the MPII val set.

TABLE IV
RESULTS OF DIFFERENT INITIAL ARCHITECTURES ON THE MPII VAL SET

Architecture	Groups	Params	FLOPs	PCKh@0.5
Pyramid	{2,4,8}	29.7M	12.7G	91.2
Inv-Pyramid	{8,4,2}	30.1M	14.0G	**91.7**
Bottleneck	{8,2,8}	30.2M	14.2G	91.3
Spindle	{2,8,2}	**29.6M**	**12.3G**	91.4
Plain	{4,4,4}	29.8M	13.1G	**91.7**

1) Sensitivity to the Groups and Stages: In Table I, we change the number of the part groups. We choose the number of groups from the set $\{0, 2, 4, 8, 16\}$, evaluating the effect on the number of network parameters ("Params"), the floating point of operations ("FLOPs") and the accuracy ("PCKh@0.5"). By setting the group number to 0, we turn off the part grouping and degrade the accuracy. Too many groups (8 and 16 groups) increase the complexity of the architecture search. It requires more training time and also degrades performance. Below, we use four groups.

In Table II, we experiment with using a different number of stages ($\{0, 1, 2, 3, 4, 5\}$). Again, 0 means that the part grouping is disabled. The single stage ($T = 1$) loses the context of parts in various ranges, yielding an unsatisfactory accuracy. More stages enrich the context but require more computation. Considering the trade-off between the performance and computation, we use three stages as default.

2) Sensitivity to the Backbone DCNs: Accurate-PGNet can be built on different backbone DCNs. We use the popular backbone DCNs [3], [21], [44] to construct Accurate-PGNet. The accuracies and the computational costs are listed in Table III. Accurate-PGNet effectively learns richer context from the groups. Compared to the independent DCN without grouping, Accurate-PGNet successfully improves the accuracy. Moreover, the optimization of Accurate-PGNet removes the redundant network connections between the feature maps, allowing the relevant groups to be merged. On average, Accurate-PGNet additionally requires 1.2 M network parameters and 3.7 G FLOPs with different backbone DCNs.

3) Sensitivity to Initial Architectures: In Table IV, we present a comprehensive comparison of the HPE performance of various

TABLE V
RESULTS OF DIFFERENT GROUPING STRATEGIES ON THE MPII VAL SET

Method	Group	Params	FLOPs	PCKh@0.5
Fully-connected	{4, 4}	29.8M	13.4G	90.5
PBN [1]	{5, 1}	**29.2M**	11.2G	90.4
PNFS-8 [33]	{8, 1}	29.4M	11.6G	90.3
PNFS-3 [33]	{3, 1}	**29.2M**	**11.0G**	90.6
DLCM [9]	{12, 6}	29.7M	12.2G	90.7
Accurate-PGNet-S2	{4,4}	29.7M	12.7G	91.5
Accurate-PGNet-S4	{4,4,4}	29.8M	13.1G	**91.7**

Accurate-PGNet-S2 and -S4 have 2 and 4 stages of part grouping.

initial network architectures. Notably, we found that the number of stages set to 3 for different initial architectures ensures a fair comparison. The column "Groups" provides a clear breakdown of the number of groups at each stage, which is a key factor in our evaluation.

The initial Accurate-PGNet is plain, which means the group numbers at all stages are the same. We evaluate the pyramid architecture, where the group numbers increase at later stages. The pyramid enforces the network to produce larger part groups at earlier stages, losing the relationship between parts in local ranges. With an inverse pyramid ("Inv-Pyramid"), we achieve accuracy on par with plain architecture. It demonstrates the importance of using larger groups at earlier stages. Furthermore, we use the bottleneck and spindle architectures to initialize the network, where the middle stage has fewer and more groups, respectively. Again, the spindle architecture degrades the accuracy due to the loss of local information. Though the beginning stage of the bottleneck architecture has more groups, fewer groups at the middle stage disallow the multiple ranges of context to be formed, thus leading to a lower accuracy than Accurate-PGNet.

4) Comparison With Prescribed Grouping Strategies: To evaluate effectiveness of Accurate-PGNet, we compare Accurate-PGNet with the prescribed strategies for part groups. We report the performance in Table V. First, we evaluate the fully-connected strategy for part grouping (see the first row). The fully-connected strategy merges all of the groups into each new group. It misses the correlation between parts, inevitably injecting redundant information into the visual feature maps of the new groups. Thus, the performance of the fully-connected strategy lags far behind Accurate-PGNet. Next, we compare Accurate-PGNet with the alternative architectures, which follow the existing HPE methods [1], [9], [33] to determine the part groups. Generally, these methods conduct the part grouping at 2 stages. They are in the inverse pyramid shape, for capturing the local and global context in a bottom-up manner. For a fair comparison, we report Accurate-PGNet with 2 stages (see "Accurate-PGNet-S2") of part grouping in Table V. Compared to the prescribed grouping strategies, the flexible Accurate-PGNet yields a better accuracy.

5) Different Ways of Supervising Part Grouping: With (13) and (14), we measure the similarities between the groups in different/identical larger group(s). In Table VI, we initialize the hyperparameters of the similarities with different values and compare the accuracies. It is evident that without any similarity constraints ($\alpha = 0, \beta = 0$), we rely solely on the localization error of separate parts to supervise part grouping. This approach still yields better results compared to the original HRNet. By incorporating these similarities, we further enhance accuracy. However, the results from the first row indicate that $\mathcal{L}_c$ may not be effective in the absence of $\mathcal{L}_s$. It is because that $\mathcal{L}_c$ aims to maximize the correlations between *normal groups* and the *main groups*. But without $\mathcal{L}_s$, *main groups* likely exhibit identical features. Under this condition, $\mathcal{L}_s$ will not affect the learning. The superior performance in the first column validates the effectiveness of $\mathcal{L}_s$. The results from the second and third rows demonstrate that $\mathcal{L}_c$ aids in the specific part feature learning of our model, with the feature diversity of *main groups* is ensured by $\mathcal{L}_s$. Experiments of the last row show that inappropriate values of hyperparameter of $\mathcal{L}_c$ can also deteriorate the model's performance, underscoring the importance of proper hyperparameter initialization. In this paper, we set values $\alpha = 1$ and $\beta = 1$.

TABLE VI
RESULTS OF VARIOUS WAYS OF SUPERVISING PART GROUPING ON THE MPII VAL SET

PCKh@0.5 β α	0	0.1	1	5
0	90.9	91.0	91.0	90.9
0.1	91.1	91.2	91.1	91.0
1	91.4	91.6	**91.7**	91.5
5	91.3	91.4	91.5	91.3

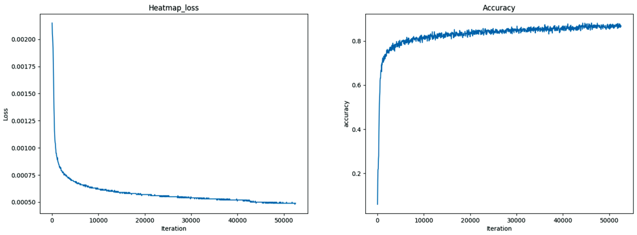


Fig. 9. Training losses and accuracies of Accurate-PGNet on the COCO *train* datasets along the training process.

6) Setting of Iteration Number: As is widely recognized, the quantity of iterations plays a pivotal role in influencing both the training process and the overall performance of a model. To strike an optimal balance between training efficiency and model efficacy, we train the model by varying the number of iterations, capping it at 55,000. The outcomes of this approach are depicted in Fig. 9. A sharp decline in loss is observed within the initial 10,000 iterations, followed by a more gradual reduction from 10,000 to 40,000 iterations. This indicates that the first 40,000 iterations are crucial for enhancing model performance. Around the 45,000 iteration mark, the model is expected to converge, with the loss stabilizing. A similar pattern is observed in the prediction accuracy, mirroring the changes in loss. These findings suggest that the optimal point for balancing the number of training iterations lies between 45,000 and 55,000. Consequently, in this study, we have empirically determined the iteration count to be fixed within this range.

TABLE VII
COMPARISONS WITH STATE-OF-THE-ART METHODS

Method	Params	FLOPs	PCKh@0.5
Hourglass [4]	25.1M	19.1G	89.2
SimpleBaseline [11]	68.6M	20.9G	89.6
DLCM [9]	15.5M	33.6G	89.8
HRNet-W32-S4 [3]	28.5M	9.5G	90.3
DARK [45]	28.5M	9.5G	90.6
TokenPose [46]	28.1M	12.8G	90.2
TransPose [47]	17.5M	13.0G	90.3
HRFormer-B [57]	43M	12.2G	90.4
ViTPose [48]	632M	121.05G	93.0
PNFS [33]	16.3M	9.4G	90.1
PoseNAS [34]	33.6M	14.8G	90.4
Accurate-PGNet-S3	**8.9M**	**8.6G**	90.6
Accurate-PGNet-S4	29.8M	13.1G	91.7
Accurate-PGNet-V	633.9M	126.9G	**94.2**

The total accuracies are reported on the MPII val set.

TABLE VIII
COMPARISONS WITH STATE-OF-THE-ART METHODS

Method	Params	FLOPs	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total
CPHR [49]	-	-	97.9	95.1	89.9	85.3	89.4	85.7	81.7	89.7
Recurrent HPE [50]	-	-	97.7	95.0	88.2	83.0	87.9	82.6	78.4	88.1
CPM [2]	-	-	97.8	95.0	88.7	84.0	88.4	82.8	79.4	88.5
DeepCut [51]	42.6M	41.2G	96.8	95.2	89.3	84.4	88.4	83.4	78.0	88.5
Hourglass [4]	25.1M	19.1G	98.2	96.3	91.2	87.1	90.1	87.4	83.6	90.9
SimpleBaseline [11]	68.6M	20.9G	98.5	96.6	91.9	87.6	91.1	88.1	84.1	91.5
HRNet-W32 [3]	28.5M	9.5G	98.3	96.3	91.9	88.0	90.1	87.7	84.0	91.3
DARK [45]	28.5M	9.5G	98.3	96.5	91.9	87.8	90.7	88.1	84.3	91.5
TokenPose [46]	28.1M	12.8G	98.2	96.4	91.7	87.1	90.3	87.5	83.7	91.1
TransPose [47]	17.5M	13.0G	98.2	96.5	92.0	87.8	90.3	88.3	84.6	91.4
ViTPose [48]	632M	121.05	96.0	91.2	82.4	76.1	81.6	74.5	71.6	82.5
PNFS [33]	16.4M	9.4G	98.2	95.9	91.5	87.6	90.1	87.3	83.2	91.0
PoseNAS [34]	33.6M	14.8G	98.3	96.7	92.5	88.4	90.3	87.4	83.9	91.5
Accurate-PGNet-S3	**8.9M**	**8.6G**	98.3	96.5	92.4	88.2	91.7	88.9	84.8	91.9
Accurate-PGNet-S4	29.8M	13.1G	98.6	97.3	93.3	89.3	91.8	90.2	86.5	92.7
Accurate-PGNet-V	633.9M	126.9G	**98.9**	**98.0**	**95.4**	**92.5**	**94.3**	**94.1**	**91.2**	**95.1**

We report the recognition accuracies of different parts on MPII test set, in term of PCKh@0.5. Each accuracy accounts for the symmetric parts.

TABLE IX
COMPARISONS WITH STATE-OF-THE-ART METHODS

Method	Backbone	Input Size	Params	FLOPs	AP
CPN [54]	ResNet-Inception	384 × 288	42.0M	-	72.9
SimpleBaseline [11]	ResNet-152	256× 192	68.6M	35.6G	74.3
HRNet [3]	HRNet-W32-S4	384 × 288	28.5M	16.0G	75.8
HRNet [3]	HRNet-W48-S4	384 × 288	63.6M	32.9G	76.3
DARK [45]	HRNet-W48-S4	384 × 288	63.6M	32.9G	76.8
UDP [55]	HRNet-W48-S4	384 × 288	63.6M	32.9G	77.2
FastPose [56]	ResNet-152	256× 192	60M	35.6G	74.3
TokenPose [46]	HRNet-W48-S4	256 × 192	27.5M	11.0G	75.8
TransPose [47]	HRNet-Small-W48	256 × 192	17.5M	21.8G	75.8
HRFormer [57]	HRFormer-B	256 × 192	43M	12.2G	75.6
HRFormer [57]	HRFormer-B	384 × 288	43M	26.8G	77.2
ViTPose [48]	ViTPose-H	256 × 192	632M	121.05	79.1
PNFS [33]	ResNet-50	384 × 288	27.5M	11.4G	73.0
PoseNAS [34]	L18-C64	384 × 288	33.6M	14.8G	76.7
ViPNAS [52]	HRNet-W32-S4	256 × 192	16.3M	**5.6G**	74.7
Accurate-PGNet-S3	HRNet-W32-S3	384 × 288	**8.9M**	15.0G	74.1
Accurate-PGNet-S4	HRNet-W32-S4	384 × 288	29.8M	23.1G	77.4
Accurate-PGNet-V	ViTPose-H	256 × 192	633.9M	126.9G	**79.8**

We report the accuracies in term of AP on COCO validation set. We also compare the parameter numbers, FLOPs and speed of different methods.

C. Comparison With State-of-the-art Methods

1) Results on MPII Dataset: In Table VII, we compare Accurate-PGNet with state-of-the-art methods. Many methods [3], [4], [9], [11], [45] aggregated the convolutional feature maps at multiple layers, also enabling the context exchange between parts. But they globally account for the correlation of all parts, inevitably losing the local context. Comparably, Accurate-PGNet respects the specific content of the groups to determine aggregation. It achieves a better accuracy with fewer parameters and FLOPs, providing an efficient solution for human pose estimation. Some latest methods [33], [34] leveraged NAS to match the appropriate convolution kernels for learning the visual patterns of parts. Rather than using the single-part information alone, Accurate-PGNet builds more effective latent connections between parts and allows convolution kernels to capture the diverse patterns of the part groups. It harvests richer context to improve the recognition of parts. In Table VIII, we compare the results of different methods on the MPII test set. Compared to state-of-the-art methods [9], [33], [34], [45], Accurate-PGNet-S3 based on HRNet-W32-S3 requires less computation. Accurate-PGNet-S4 based on HRNet-W32-S4 achieves a better accuracy. We search the network architecture Accurate-PGNet-V based on a large model ViTPose-H [48], also obtaining better performance. It is noteworthy that our method achieves significantly higher results compared to other NAS methods such as PNFS [33], PoseNAS [34], and ViPNAS [52]. Because previous methods primarily focus on the efficiency of the searched network, neglecting the application of NAS for identifying the optimal part grouping pattern. In contrast, our proposed human-specific NAS for data-driven part grouping effectively enhances the understanding of body part correlations, thereby improving the performance of human skeleton establishment.

2) Results on COCO Dataset: In Tables IX and X, we compare the results of different methods on the COCO validation and test sets. We use Faster-RCNN without bells and whistles to detect the human bodies, which are input to Accurate-PGNet. Here, Faster-RCNN achieves 60.9 AP on the human detection on the COCO test-dev set. Based on the backbone HRNet-W32-S4, Accurate-PGNet surpasses the latest methods [34], [36], [45] on the COCO validation dataset. On the COCO test set, Accurate-PGNet-S4 again achieves a very competitive accuracy compared to DARK [45]. Compared to Accurate-PGNet, DARK requires 1.5∼2 times the parameters and FLOPs. Our method can also be employed on any backbone network with the same little computational cost. Considering the speed, our Accurate-PGNet-S3 can achieve an acceptable performance with a very high speed. The Accurate-PGNet-S4 can also achieve higher results compared to the backbone HRNet with little speed degradation. As for the huge-size model, we conducted a network search based on ViTPose-H and obtained the best performance with little computational cost increase. Thus, Accurate-PGNet is a good choice for HPE with limited computational budgets. Besides, similar to the MPII dataset, we compare our methods with other NAS methods. Our model also outperforms all of them, demonstrating the generalization ability of our NAS strategy.

3) Results on OCHuman Dataset: To evaluate the performance of different methods on human instances with heavy occlusions, we compare our method against other state-of-the-art methods on the OCHuman dataset. To focus solely on the performance of pose estimation and eliminate the variable of person detection accuracy, we utilized ground truth bounding boxes rather than those generated by a person detector. This approach was necessary because not all human instances are labeled in

TABLE X
COMPARISONS WITH STATE-OF-THE-ART METHODS

Method	Backbone	Input Size	Params	FLOPs	Speed(fps)	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
Mask-RCNN [58]	ResNet-50-FPN	-	-	-	-	63.1	87.3	68.7	57.8	71.4	-
G-RMI [59]	ResNet-101	256 × 256	42.6M	57.0G	-	68.5	87.1	75.5	65.8	73.3	73.3
CPN [54]	ResNet-Inception	384 × 288	-	-	-	72.1	91.4	80.0	68.7	77.2	78.5
RMPE [60]	Stacked Hourglass	384 × 288	28.1M	26.7G	-	72.3	89.2	79.1	68.0	78.6	-
SimpleBaseline [11]	ResNet-152	256 × 192	68.6M	35.6G	76.3	73.7	91.9	81.1	70.3	80.0	79.0
HRNet [3]	HRNet-W32-S4	384 × 288	28.5M	16.0G	87.1	74.9	92.5	82.8	71.3	80.9	80.1
HRNet [3]	HRNet-W48-S4	384 × 288	63.6M	32.9G	75.5	75.5	92.5	83.3	71.9	81.5	80.5
DARK [45]	HRNet-W48-S4	384 × 288	63.6M	32.9G	62.1	76.2	92.5	83.6	72.5	82.4	81.1
UDP [55]	HRNet-W48-S4	384 × 288	63.8M	33.0G	67.9	76.5	92.7	84.0	73.0	82.4	81.6
TokenPose [46]	HRNet-W48-S4	256 × 192	27.5M	11.0G	52.9	75.9	92.3	83.4	72.2	82.1	80.8
TransPose [47]	HRNet-Small-W48	256 × 192	17.5M	21.8G	56.7	75.0	92.2	82.3	71.3	81.1	80.8
TokenPose with EMA [61]	HRNet-W48-S4	256 × 192	27.5M	11.0G	52.9	75.4	92.5	82.5	72.5	79.9	78.3
FasterPose [62]	ResNet-50	256 × 192	25.7M	**3.8G**	166.4	70.8	91.3	78.8	67.2	76.8	76.4
HRFormer [57]	HRFormer-B	384 × 288	43M	26.8G	25.2	76.2	92.7	83.8	72.5	82.3	81.2
ViTPose [48]	ViTPose-H	256 × 192	632M	121.05	21.8	78.1	93.3	85.7	74.9	83.8	83.1
PNFS [33]	HRNet-W32-S3	384 × 288	15.8M	14.8G	84.2	72.3	90.9	79.5	68.4	79.2	77.9
PoseNAS [34]	L18-C64	384 × 288	33.6M	14.8G	-	75.9	93.0	83.8	72.2	81.4	80.7
ViPNAS [52]	HRNet-W32-S4	256 × 192	16.3M	5.6G	151.1	73.9	91.7	82.0	70.5	79.5	80.4
LitePose [63]	LitePose	448 × 448	**5.7M**	9.17G	133.0	62.4	82.5	67.9	54.8	73.7	67.1
Accurate-PGNet-S3	HRNet-W32-S3	384 × 288	8.9M	15.0G	**167.5**	72.4	91.1	79.5	68.3	79.4	78.0
Accurate-PGNet-S4	HRNet-W32-S4	384 × 288	29.8M	23.1G	61.3	76.3	92.7	83.9	72.7	82.4	81.3
Accurate-PGNet-V	ViTPose-H	256 × 192	633.9M	126.9G	21.7	**79.1**	**94.1**	**86.9**	**76.0**	**83.9**	**84.0**

We report the accuracies in terms of AP and AR on COCO test-dev set. We also compare the parameter numbers, FLOPs and speed of different methods.

TABLE XI
COMPARISONS WITH STATE-OF-THE-ART METHODS

Method	Backbone	Resolution	Params	AP	AP^{50}	AR
SimpleBaseline [11]	ResNet-152	384 × 288	68.6M	58.8	72.7	63.1
HRNet [3]	HRNet-W32-S4	384 × 288	28.5M	60.9	76.0	65.1
HRNet [3]	HRNet-W48-S4	384 × 288	63.6M	62.1	76.1	65.9
MIPNet [53]	HRNet-W48-S4	384 × 288	63.7M	74.1	89.7	81.0
HRFormer [57]	HRFormer-S	384 × 288	**7.8M**	53.1	73.1	59.6
HRFormer [57]	HRFormer-B	384 × 288	43.2M	50.4	71.5	58.8
ViTPose-S [48]	ViTPose-S	256 × 192	22M	57.6	75.2	61.8
ViTPose-H [48]	ViTPose-H	256 × 192	632M	67.5	79.6	70.7
Accurate-PGNet-S3	HRNet-W32-S3	384 × 288	8.9M	50.3	70.2	57.1
Accurate-PGNet-S4	HRNet-W32-S4	384 × 288	29.8M	63.2	76.9	66.4
Accurate-PGNet-H	ViTPose-H	256 × 192	633.9M	**87.3**	**93.5**	**88.6**

The total accuracies are reported on the OCHuman test set with ground truth bounding boxes. We also compare the parameter numbers of different methods.

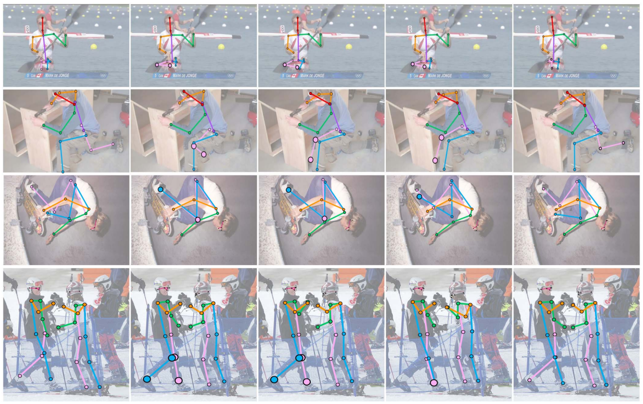
TABLE XII
THE RESULTS OF THE STATE-OF-THE-ART METHODS ON THE OCHUMAN DATASET

Method	Backbone	Input Size	Speed(fps)	OC-val	OC-test
SimpleBaseline [11]	Swin-B	256× 192	16.6	40.1	39.8
HRNet [3]	HRNet-W48-S4	384 × 288	35.5	38.1	38.1
DARK [45]	HRNet-W48-S4	384 × 288	35.5	38.6	39.2
UDP [55]	HRNet-W48-S4	384 × 288	67.9	38.6	38.8
HRFormer [57]	HRFormer-B	384 × 288	25.2	40.5	40.3
Poseur [64]	HRFormer-B	384 × 288	25.8	44.4	45.6
ViTPose [48]	ViTPose-H	256 × 192	21.8	46.7	45.8
Accurate-PGNet-S3	HRNet-W32-S3	384 × 288	**121.5**	36.8	36.9
Accurate-PGNet-S4	HRNet-W32-S4	384 × 288	61.3	39.5	39.7
Accurate-PGNet-V	ViTPose-H	256 × 192	21.7	**47.6**	**47.1**

The metrics are computed only on the occluded joints that overlap with the COCO annotated joints. The GT bounding box is used. “OC” denotes the OCHuman dataset. We also compare the speed of different methods.

the OCHuman dataset, and reliance on a person detector could introduce false positives or omissions, potentially distorting the true capabilities of the pose estimation models. The comparative results are presented in Table XI. Our findings reveal that Accurate-PGNet significantly outperforms established general SOTA methods, such as HRFormer and ViTPose, by approximately 20 AP points. Moreover, compared to methods with intricate architectures, for instance, MIPNet [53], Accurate-PGNet improves over 10 AP points on the OCHuman validation set. This is noteworthy as Accurate-PGNet does not explicitly incorporate specialized structural designs to address occlusions. Instead, the enhancement in performance can be attributed to the effective use of part relationships and context understanding, which are learnt from the structures identified through our NAS strategy. Besides, we specifically compare the prediction results of occluded joints, which are shown in Table XII. It is easy to tell that our method can obtain the best performance by learning the specific part relationships and corresponding visual contexts.

4) Comparison of Robustness: To evaluate the robustness of our method, we conduct experiments with different noise and lighting settings, which can degrade the model’s performance. We conduct experiments via clean images from **MSCOCO** *val* and processed images from **COCO-C** dataset. The results are shown in Table XIII. Compared with previous SOTA methods like HRFormer and ViTPose, our method exhibits the best robustness under different corrupted conditions.



(a) Ground Truth (b) HRNet (c) Dark (d) TransPose (e) Accurate-PGNet

Fig. 10. Part recognition results of HRNet-based methods. The top and the bottom rows contain the examples taken from MPII and COCO validation sets, respectively. Larger dots are the parts with problematic locations. For a clear visualization, we only provide the predicted parts on the salient persons.

TABLE XIII
ROBUSTNESS BENCHMARK RESULTS ON THE COCO-C DATASET, WITH mRR SCORES PRESENTED AS PERCENTAGES (%)

Method	**Backbone**	**Input Size**	**Clean**		**Blur & Noise**			**Lightning**		
			mAP	mAR	mAP	mAR	mRR	mAP	mAR	mRR
SimpleBaseline [11]	ResNet-152	256× 192	73.59	79.09	47.23	53.21	64.18	56.76	62.62	77.13
HRNet [3]	HRNet-W32-S4	384 × 288	74.90	80.38	48.75	54.58	65.09	59.42	65.00	79.33
HRNet [3]	HRNet-W48-S4	384 × 288	75.58	80.85	50.29	56.09	66.53	60.12	65.60	79.54
MSPN [65]	MSPN-50	384 × 288	72.28	78.80	43.57	50.48	60.28	55.15	62.06	76.30
MSPN [65]	4 × MSPN-50	384 × 288	76.49	82.62	51.42	58.39	67.22	60.63	67.42	79.26
HRFormer [57]	HRFormer-S	384 × 288	73.84	79.27	47.30	53.38	64.06	57.95	63.77	78.48
HRFormer [57]	HRFormer-B	384 × 288	75.37	80.67	49.34	55.28	65.46	59.34	64.88	78.73
LiteHRNet [66]	LiteHRNet-18	384 × 288	64.16	70.45	37.15	43.58	57.91	47.57	54.01	74.15
LiteHRNet [66]	LiteHRNet-30	384 × 288	67.54	73.61	40.30	46.75	59.66	51.57	57.92	76.35
TransPose [47]	HRNet-W32-S4	256 × 192	74.17	79.45	46.43	52.11	62.60	58.29	63.82	78.59
TransPose [47]	HRNet-W48-S4	256 × 192	75.28	80.33	47.94	53.61	63.68	60.23	65.58	80.01
Poseur [64]	HRNet-W48-S4	384 × 288	77.62	82.33	52.31	57.60	67.39	64.22	69.11	82.75
Poseur [64]	HRFormer-B	384 × 288	77.97	82.95	54.48	60.18	69.88	65.54	70.73	84.07
Poseur [64]	ViTPose-H	256 × 192	76.72	81.92	54.61	60.24	71.19	65.11	70.44	84.87
ViTPose [48]	ViTPose-S	256 × 192	73.92	79.24	49.58	55.72	67.07	59.01	64.78	79.83
ViTPose [48]	ViTPose-H	256 × 192	78.84	83.92	58.89	64.56	74.7	66.96	72.29	84.93
SimCC [67]	ViPNAS-MobileNetV3	256 × 192	69.48	75.52	42.06	48.45	60.54	53.0	59.23	76.28
Accurate-PGNet-V	ViTPose-H	256 × 192	**79.22**	**84.73**	**59.01**	**64.99**	**75.3**	**67.11**	**72.65**	**85.47**

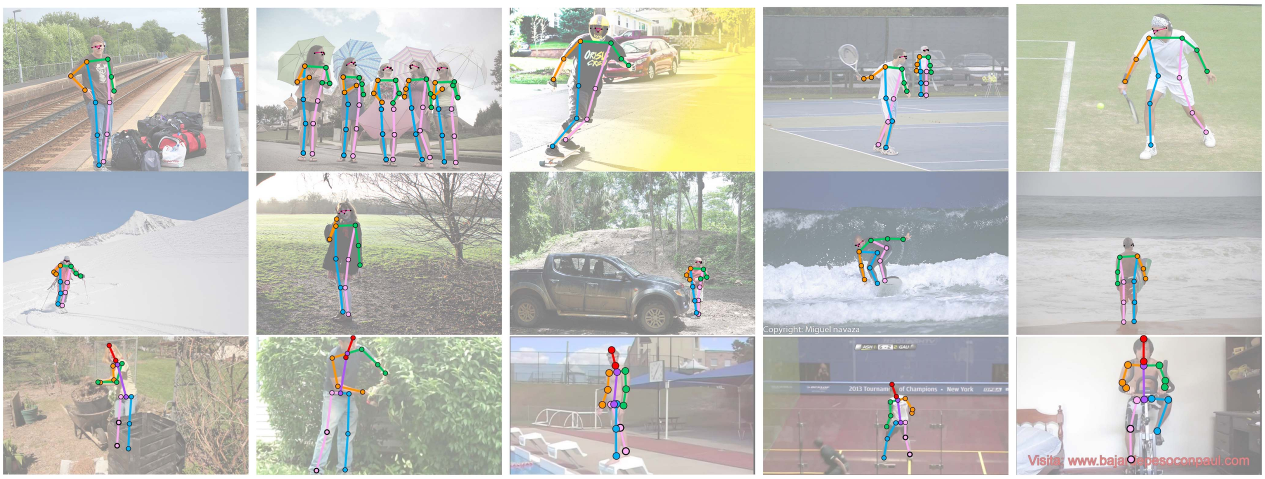

Copyright: Miguel navaza
2013 Tournament of Champions - New York
Visit: www.baja...pesoconpaul.com

Fig. 11. The visual results of our method on MSCOCO and MPII datasets. The dots represent the detected human parts and the lines are the skeletons.

5) Comparison of Visual Results: In Fig. 10, we compare the visual results of different HRNet-based methods on both MPII and COCO datasets. It is easy to see that in some challenging scenarios, such as occlusions and crowded backgrounds, our results are better than those state-of-the-art methods, namely, (b) HRNet [3], (c) DARK [45], and (d) TransPose [47]. Accurate-PGNet shows a better ability to handle visual degradation by building specific connections between parts. In Fig. 11, we present more visual results.

VI. Conclusion

Human body part correlation is a significant factor in the establishment of the human skeleton. In this study, we introduce a novel framework, Accurate-PGNet, which features a flexible and automatic body grouping strategy. Accurate-PGNet has the unique ability to merge related body parts, forming part groups that effectively model the visual and spatial relationships between parts. Our model also learns specific information from the part groups, thereby respecting the diversity of different groups. To our knowledge, this is the first study to employ the NAS strategy to identify an effective part grouping pattern for human skeleton establishment. The experimental results confirm that our model is not only efficient but also a highly practical method. In the future, we will investigate a more innovative scheme for learning the stage and group numbers. Rather than the current method, which requires progressively grouping the parts, we will investigate a faster grouping strategy to further reduce the computational overheads. We will also investigate a more general architecture to provide effective results appropriate for different datasets. This topic is more challenging due to the various datasets' different scenarios and annotations.

References

[1] W. Tang and Y. Wu, "Does learning specific features for related parts help human pose estimation?," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1107–1116.

[2] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4724–4732.

[3] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5686–5696.

[4] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 483–499.

[5] H. Liu, T. Liu, Y. Chen, Z. Zhang, and Y.-F. Li, "EHPE: Skeleton cues-based Gaussian coordinate encoding for efficient human pose estimation," *IEEE Trans. Multimedia*, vol. 26, pp. 8464–8475, 2024.

[6] Q. Bao, W. Liu, Y. Cheng, B. Zhou, and T. Mei, "Pose-guided tracking-by-detection: Robust multi-person pose tracking," *IEEE Trans. Multimedia*, vol. 23, pp. 161–175, 2021.

[7] H. Liu et al., "Fast human pose estimation in compressed videos," *IEEE Trans. Multimedia*, vol. 25, pp. 1390–1400, 2023.

[8] M. Andriluka, S. Roth, and B. Schiele, "Pictorial structures revisited: People detection and articulated pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1014–1021.

[9] W. Tang, P. Yu, and Y. Wu, "Deeply learned compositional models for human pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 197–214.

[10] L. Jin et al., "Grouping by Center: Predicting centripetal offsets for the bottom-up human pose estimation," *IEEE Trans. Multimedia*, vol. 25, pp. 3364–3374, 2023.

[11] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 472–487.

[12] S. Park, B. X. Nie, and S.-C. Zhu, "Attribute and-or grammar for joint parsing of human pose, parts and attributes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 7, pp. 1555–1569, Jul. 2018.

[13] X. Nie, J. Feng, J. Xing, S. Xiao, and S. Yan, "Hierarchical contextual refinement networks for human pose estimation," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 924–936, Feb. 2019.

[14] Y. Tian, C. L. Zitnick, and S. G. Narasimhan, "Exploring the spatial hierarchy of mixture models for human pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 256–269.

[15] Y. Bin et al., "Structure-aware human pose estimation with graph convolutional networks," *Pattern Recognit.*, vol. 106, 2020, Art. no. 107410.

[16] W. Wang et al., "Learning compositional neural information fusion for human parsing," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 5702–5712.

[17] Z. Qiu, K. Qiu, J. Fu, and D. Fu, "DGCN: Dynamic graph convolutional network for efficient multi-person pose estimation," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 11924–11931.

[18] F. Wang and Y. Li, "Beyond physical connections: Tree models in human pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 596–603.

[19] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2D human pose estimation: New benchmark and state of the art analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3686–3693.

[20] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[22] S. Park and S.-C. Zhu, "Attributed grammars for joint estimation of human attributes, part and pose," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2372–2380.

[23] Y. Yang and D. Ramanan, "Articulated pose estimation with flexible mixtures-of-parts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 1385–1392.

[24] B. Baker, O. Gupta, N. Naik, and R. Raskar, "Designing neural network architectures using reinforcement learning," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–18.

[25] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–16.

[26] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8697–8710.

[27] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 4780–4789.

[28] E. Real et al., "Large-scale evolution of image classifiers," in *Proc. Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 2902–2911.

[29] H. Liu, K. Simonyan, and Y. Yang, "DARTS: Differentiable architecture search," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–13.

[30] C. Liu et al., "Auto-DeepLab: Hierarchical neural architecture search for semantic image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 82–92.

[31] H. Xu, L. Yao, W. Zhang, X. Liang, and Z. Li, "Auto-FPN: Automatic network architecture adaptation for object detection beyond classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6648–6657.

[32] J. Guo et al., "Hit-detector: Hierarchical trinity architecture search for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11402–11411.

[33] S. Yang, W. Yang, and Z. Cui, "Searching part-specific neural fabrics for human pose estimation," *Pattern Recognit.*, vol. 128, 2022, Art. no. 108652.

[34] Q. Bao, W. Liu, J. Hong, L. Duan, and T. Mei, "Pose-native network architecture search for multi-person human pose estimation," in *Proc. ACM Int. Conf. Multimedia*, 2020, pp. 592–600.

[35] W. Zhang, J. Fang, X. Wang, and W. Liu, "EfficientPose: Efficient human pose estimation with neural architecture search," *Comput. Vis. Media*, vol. 7, no. 3, pp. 335–347, 2021.

[36] X. Gong et al., "AutoPose: Searching multi-scale branch aggregation for pose estimation," 2020, *arXiv:2008.07018*.

[37] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," in *Proc. Int. Conf. Learn. Representations*, 2016, pp. 1–14.

[38] Z. Liu, M. Sun, T. Zhou, G. Huang, and T. Darrell, "Rethinking the value of network pruning," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–21.

[39] J. Mei et al., "Atomnas: Fine-grained end-to-end neural architecture search," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–14.

[40] H. Liang et al., "DARTS+: Improved differentiable architecture search with early stopping," 2019, *arXiv:1909.06035*.

[41] A. Paszke et al., "Automatic differentiation in pytorch," in *Proc. Annu. Conf. Neural Inf. Process. Syst. Workshops*, 2017, pp. 1–4.

[42] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–15.

[43] R. Zeng et al., "Benchmarking the robustness of temporal action detection models against temporal corruptions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 18263–18274.

[44] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.

[45] F. Zhang, X. Zhu, H. Dai, M. Ye, and C. Zhu, "Distribution-aware coordinate representation for human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7091–7100.

[46] Y. Li et al., "TokenPose: Learning keypoint tokens for human pose estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 11293–11302.

[47] S. Yang, Z. Quan, M. Nie, and W. Yang, "TransPose: Keypoint localization via transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 11782–11792.

[48] Y. Xu, J. Zhang, Q. Zhang, and D. Tao, "ViTPose: Simple vision transformer baselines for human pose estimation," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2022, vol. 35, pp. 38571–38584.

[49] A. Bulat and G. Tzimiropoulos, "Human pose estimation via convolutional part heatmap regression," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 717–732.

[50] V. Belagiannis and A. Zisserman, "Recurrent human pose estimation," in *Proc. IEEE Int. Conf. Automat. Face Gesture Recognit.*, 2017, pp. 468–475.

[51] L. Pishchulin et al., "DeepCut: Joint subset partition and labeling for multi person pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4929–4937.

[52] L. Xu et al., "ViPNAS: Efficient video pose estimation via neural architecture search," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 16067–16076.

[53] R. Khirodkar, V. Chari, A. Agrawal, and A. Tyagi, "Multi-instance pose networks: Rethinking top-down pose estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 3102–3111.

[54] Y. Chen et al., "Cascaded pyramid network for multi-person pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7103–7112.

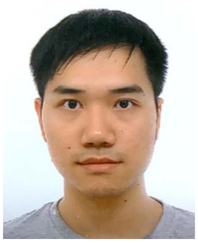
[55] J. Huang, Z. Zhu, F. Guo, and G. Huang, "The devil is in the details: Delving into unbiased data processing for human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5699–5708.

[56] H.-S. Fang et al., "AlphaPose: Whole-body regional multi-person pose estimation and tracking in real-time," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 6, pp. 7157–7173, Jun. 2023.

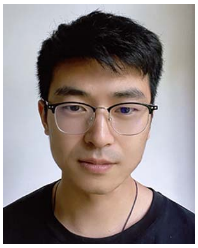
[57] Y. Yuan et al., "HRFormer: High-resolution vision transformer for dense predict," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2021, pp. 7281–7293.

[58] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.
[59] G. Papandreou et al., "Towards accurate multi-person pose estimation in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3711–3719.
[60] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "RMPE: Regional multi-person pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2353–2362.
[61] K. Ludwig, P. Harzig, and R. Lienhart, "Detecting arbitrary intermediate keypoints for human pose estimation with vision transformers," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. Workshops*, 2022, pp. 663–671.
[62] H. Dai et al., "FasterPose: A faster simple baseline for human pose estimation," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 18, no. 4, pp. 103:1–103:16, 2022.
[63] Y. Wang, M. Li, H. Cai, W. Chen, and S. Han, "Lite pose: Efficient architecture design for 2D human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 13116–13126.
[64] W. Mao et al., "Poseur: Direct human pose regression with transformers," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 72–88.
[65] W. Li et al., "Rethinking on multi-stage networks for human pose estimation," 2019, *arXiv:1901.00148*.
[66] C. Yu et al., "Lite-HRNet: A lightweight high-resolution network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 10440–10450.
[67] Y. Li et al., "SimCC: A simple coordinate classification perspective for human pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 89–106.

Renjie Zhang received the B.Eng. degree in software engineering from Sun Yat-sen University, Guangzhou, China, in 2019. He is currently working toward the Ph.D. degree in computing with The Hong Kong Polytechnic University, Hong Kong. His research interests include human skeleton establishment, neural architecture search, pose estimation, deep learning, and 3D human reconstruction.



Di Lin (Member, IEEE) received the B.Eng. degree in software engineering from Sun Yat-sen University, Guangzhou, China, in 2012, and the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong, Hong Kong, in 2016. He is currently an Associate Professor with the College of Intelligence and Computing, Tianjin University, Tianjin, China. His research interests include computer vision and machine learning.



Xin Wang received the B.Eng. degree in computer science and technology from the Dalian University of Technology, Dalian, China, in 2017. He is currently working toward the Ph.D. degree in computing with The Hong Kong Polytechnic University, Hong Kong. His research interests include image synthesis, deep learning, and diffusion models.

George Baciu (Senior Member, IEEE) received the Ph.D. degree in systems design engineering from the University of Waterloo, Waterloo, ON, Canada, in 1992. He was a Professor with the Department of Computing (COMP), The Hong Kong Polytechnic University (PolyU), Hong Kong. He was also a Member of the Waterloo Computer Graphics Laboratory and the Pattern Analysis and Machine Intelligenc Laboratory. He is the founding Director of the GAME Laboratory, The Hong Kong University of Science and Technology, Hong Kong, in 1993, and the Graphics and Multimedia Applications Laboratory, COMP, PolyU, in 2000. He has authored or coauthored extensively in computer graphics, image processing, and VR journals and conferences, and was as Chair of many international conferences. His research interests include information visualization, cognitive computing, virtual reality and computer graphics, with applications to cognitive digital agents, digital twins, motion synthesis, animation, collision detection, geometric modeling, and image analysis.

C. L. Philip Chen (Life Fellow, IEEE) received the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1988. He is currently a Chair Professor and the Dean of the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. He is also a Program Evaluator of the Accreditation Board of Engineering and Technology Education (ABET) in the U.S., for computer engineering, electrical engineering, and software engineering programs, he successfully architects the University of Macau's Engineering and Computer Science programs receiving accreditations from Washington or Seoul Accord through Hong Kong Institute of Engineers (HKIE), of which is considered as his utmost contribution in engineering or computer science education for Macau as the former Dean of the Faculty of Science and Technology. His research interests include systems, cybernetics, and computational intelligence.. He was the recipient of the IEEE Norbert Wiener Award in 2018, for his contribution in systems and cybernetics, and machine learnings. From 2018 to 2019, he was a Highly Cited Researcher by Clarivate Analytics. He was also the recipient of the 2016 Outstanding Electrical and Computer Engineers Award from his alma mater, Purdue University, in 1988, after he graduated from the University of Michigan, Ann Arbor, MI, USA, in 1985. He was the IEEE Systems, Man, and Cybernetics Society President from 2012 to 2013, Editor-in-Chief of IEEE TRANSACTIONS ON CYBERNETICS from 2020 to 2021, and the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS from 2014 to 2019, and currently, an Associate Editor for IEEE TRANSACTIONS ON FUZZY SYSTEMS. From 2015 to 2017, he was the Chair of Technical Committee 9.1 Economic and Business Systems of International Federation of Automatic Control and currently the Vice President of Chinese Association of Automation. He is a Fellow of IEEE, AAAS, IAPR, CAA, and HKIE; a member of Academia Europaea (AE), European Academy of Sciences and Arts (EASA), and International Academy of Systems and Cybernetics Science (IASCYS)

Ping Li (Member, IEEE) received the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong, Hong Kong, in 2013. He is currently an Assistant Professor with the Department of Computing, and an Assistant Professor with the School of Design, The Hong Kong Polytechnic University, Hong Kong. He has authored or coauthored more than 250 top-tier scholarly research articles, pioneered many new research directions, and made a series of landmark contributions in his areas. He has an excellent research project reported by the *ACM TechNews*, which only reports the top breakthrough news in computer science worldwide. His many research outcomes have strong impacts to research fields, addressing societal needs and contributed tremendously to the people concerned. His research interests include image or video stylization, colorization, artistic rendering and synthesis, computational art, and creative media.