Pose Neural Fabrics Search

Sen Yang, Wankou Yang, *Member, IEEE* and Zhen Cui, *Member, IEEE*

***Abstract*—Neural Architecture Search (NAS) technologies have been successfully performed for efficient neural architectures for tasks such as image classification and semantic segmentation. However, existing works implement NAS for target tasks independently of domain knowledge and focus only on searching for an architecture to replace the human-designed network in a common pipeline. *Can we exploit human prior knowledge to guide NAS?* To address it, we propose a framework, named Pose Neural Fabrics Search (PNFS), introducing prior knowledge of body structure into NAS for human pose estimation. We lead a new neural architecture search space, by parameterizing cell-based neural fabric, to learn micro as well as macro neural architecture using a differentiable search strategy. To take advantage of part-based structural knowledge of the human body and learning capability of NAS, global pose constraint relationships are modeled as multiple part representations, each of which is predicted by a personalized cell-based neural fabric. In part representation, we view human skeleton keypoints as entities by representing them as vectors at image locations, expecting it to capture keypoints feature in a relaxed vector space. The experiments on MPII and MS-COCO datasets demonstrate that PNFS[1] can achieve comparable performance to state-of-the-art methods, with fewer parameters and lower computational complexity.**

***Index Terms*—Human pose estimation, neural architecture search, prior knowledge, neural fabrics, vector representation**

I. Introduction

NEURAL Architecture Search (NAS), the process of learning the structure of neural network [8], [11], [15], [27], [28], [39], [56], [62], [63], can play a potential role at automatically designing network architectures. Current methods mainly take image classification as a basic task and only search for a micro cell to build a chain-like structure, thus the neural search space is still at the limit of a micro search space. However, when applying NAS to dense prediction tasks such as semantic segmentation and human pose estimation, the micro search space is no longer able to generate more complex architectures. Therefore, it become a necessity to artificially design the macro search space allowing identifying a hierarchical structure upon cells for these tasks. In addition, existing works focus on discovering an alternative to the human-designed module in a common pipeline. Such practice actually decouples the automating architecture engineering from tasks, and is thus unable to take advantage of the domain knowledge of a specific task.

In this work, we study how to search neural architectures with the guide of prior knowledge for human pose estimation task and propose a framework named Pose Neural Fabrics Search. We notice that modern methods [7], [9], [12], [29]–[33], [47], [54], [55], [59], [59] conducting human pose estimation based on deep CNNs, regardless of top-down or bottom-up pipeline, convert it into pixel-wise prediction problem; they usually focus on two aspects: *neural architecture design* [9], [31], [32], [47], [55], [59] and *pose representation* [3], [30], [33]–[35], [41], [61]. Next, we will discuss our motivations from these two aspects.

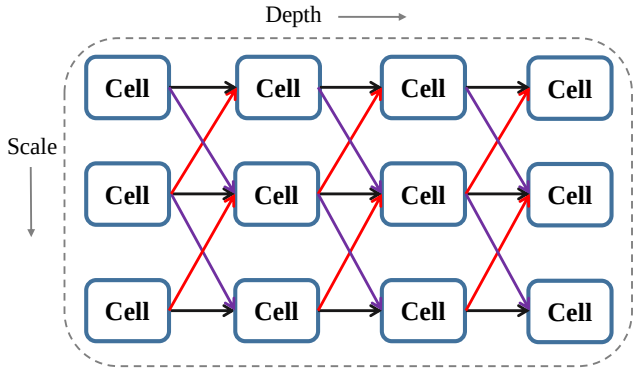

Fig. 1. A schematic map of the structure of cell-based neural fabric. The neural fabric is woven by cells in different scales and layer depths. Black arrow represents identity transformation; purple arrow represents reducing spatial size and doubling the channels of feature maps; red arrow represents increasing spatial size and halving the channels of feature maps.

Naturally, the first step to introduce NAS into pose estimation is to construct an architecture search space that can identify multi-scale, stacked or cascaded neural network. To this end, we propose a general parameterized cell-based neural fabric to encode micro and macro architecture parameters into cells, where the discrete search space is relaxed into continuous search space to make it searchable by gradient descent. This design is motivated by Convolutional Neural Fabrics [46] and DARTS [28], it can be described as a neural fabric architecture woven by cells, as shown in Figure 1. In addition, we observe that DARTS [28] potentially introduces a bias: the final trained operations with relatively smaller weight will be regarded as trivial operations and shall be pruned by *argmax*. But this will cause the derived child network inconsistent with converged parent network, SNAS [57] attempts to eliminate this problem by introducing Gumbel-Softmax. To avoid this bias, we do not re-discretize the continuous architecture, which means that all operations are densely preserved with mixed weights at macro and micro level. In order to verify such a setting, we test the performances of architectures randomly sampled from the continuous parameter space. Furthermore, we explore a simple yet effective synchronous optimization method as major search strategy to reduce the cost of time

Sen Yang and Wankou Yang are with the School of Automation of Southeast University, Nanjing 210096, China. E-mail: yangsenius@seu.edu.cn, wkyang@seu.edu.cn. (Corresponding author: Wankou Yang.)

Zhen Cui is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. E-mail: zhen.cui@njust.edu.cn.

[1]Code is available at https://github.com/yangsenius/PoseNFS

and computational budgets for the search process.

In terms of the pose representation, common deep learning methods usually utilize heatmaps (a.k.a confidence maps), encoding semantics and locations of body keypoints, instead of numerical coordinates to provide supervision signal. And the relationship of all joints of body is indiscriminately learned by regression heatmaps using shared convolutional blocks. Though the image-dependent relationships between parts can be learned by convolutional neural networks with large capacity, but such implicit discriminative learning method ignores the explicit knowledge that the distributions of all joints are induced by the highly structured human body. For example, assuming that we have observed the location of the left elbow, the distribution of the left wrist's location is independent of the most other joints, even the left shoulder. In other words, each joint is closely related to its neighboring joint and weakly related with those far away from it. Therefore the main drawback of current CNN-based methods is that using the same network to learn shared feature for all keypoints fails to depict the pairwise relationships between parts.

Early part-based models [2], [14], [48], [53] take prior structure knowledge into consideration, but usually use part-detector based on hand-crafted feature or hand-designed convolutional neural network. To further develop the part-based method, we consider exploiting the learning capacity of NAS to learn different structures of CNNs to extract different feature representations for related parts. Intuitively, NAS can learn personalized neural fabrics to adapt to the requirements of different body parts with different micro and macro architecture parameters. Concretely, we disentangle the relationships of all parts by grouping all keypoints into different part representations according to the structure prior. And each part representation is predicted by a structure-searchable neural fabric, as shown in Figure 2. Then each location of keypoint can be predicted by the associated part representation. Recent work [51] also has found that some pairs of body parts are strongly related and grouping them to learn specific feature representation can improve pose estimation, which is similar to our method. But we employ NAS to learn different neural fabrics structure for different part groups, while it uses the same structure for all parts.

Besides, we replace scalar value by a vector in each pixel position of heatmaps, because the scalar value representing the existing probability of keypoints is still inadequate to encode local feature, and ambiguity probably occurs between image feature and groundtruth heatmaps if some invisible keypoint is labeled. Inspired by Capsules [44], we view keypoints as entities in image pixels. We use the length (ℓ_2 norm) of the vector to represent the existing score of the keypoint and vector space to capture more local component information of the body part. This method allows vectors to range in a relaxed space when some occluded or invisible keypoints' locations are under strong supervision. We describe this method as vector in pixel.

In summary, our main contributions are: (1) We propose a general parameterized cell-based neural fabric architecture allowing search continuous architecture parameters at micro and macro architecture search space. (2) Introducing the prior structure of the human body, we search multiple personalized cell-based neural fabrics to predict multiple part representations using a gradient-based search strategy, further developing the part-based method. (3) We empirically demonstrate the effectiveness of vector in pixel method for body keypoint localization. (4) With light-weight models and low computing complexity, our method achieves comparable results with state-of-the-art performance on MPII and MS-COCO datasets.

II. Related Work

a) Neural Architecture Search: Our work is motivated by convolutional neural fabrics [46] and neural architecture search methods [8], [27], [28], [56]. Liu et al. [28] propose the continuous relaxation of the architecture representation to search architecture using gradient descent. Chen et al. [8] use random search to address dense image prediction by Dense Prediction Cell. Ghiasi et al. [15] use Reinforce Learning to explore more connection possibilities in different scales of feature pyramid network. Xie et al. [56] explore randomly wired networks for image classification and proposed the concept of *network generator*. Liu et al. [27] propose Auto-DeepLab, searching for a hierarchical backbone neural network to replace original human-designed network in a common pipeline for semantic segmentation. It aims to search a cell structure and a better path in multiple branches. Different with its two-step construction for architecture search space, our architecture space construction method is one-step, the whole architecture at macro and micro level is totally parameterized as each cell's parameters and not pruned into sparsely connected architecture.

b) Human pose estimation and part-based model: Top-down [9], [12], [16], [47], [54], [55] and bottom-up [7], [20], [21], [23], [30] human pose estimation approaches have been proven extremely successful in learning global or long-range dependencies relationships of body pose. However, parts occlusions, viewpoint variations, crowed scene, and background interference etc. are still tough problems. Compositional structure models or part-based models [2], [3], [5], [13], [14], [32], [36], [48], [49], [51], [52], [60] attempt to overcome aforementioned problems by representing the human body as a hierarchy of parts and subparts. Based on the top-down pipeline, our method also exploits the compositionality of body pose to separately predict keypoints locations and all keypoints are still constrained by global relationship due to the end-to-end learning method.

c) Vector Representation: The vector in pixel method is motivated by embedding and vector representation methods [7], [18], [30], [34], [35], [44]. Newell et al. [30] propose associative embedding to group body keypoints. Papandreou et al. [34] use geometric embedding representation to predict offset vectors of keypoints. Cao et al. [7] use part affinity vector field to supervise the part prediction. In addition, Hinton et al. [18] use matrix with extra scalar to represent an entity. Sabour et al. [44] propose Activity Vector, its length can represent existing probability and its orientation represents the instantiation parameters. In this work, we view each type of keypoint as an entity in the image and use activity vectors to locate keypoints to estimate human pose.

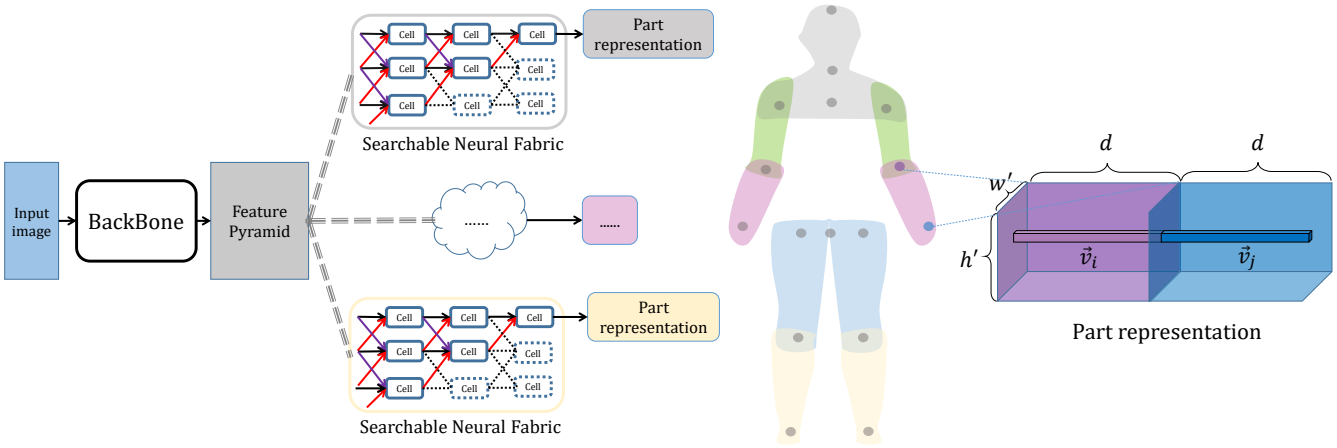

Fig. 2. Pose neural fabrics search framework. **Left:** Given an input image, feature pyramid representing multi-scale feature maps will be produced from backbone network. Then there will be P neural fabrics receiving the same feature pyramid and predicting P part representations, here two neural fabrics shown for simplification. The final cell in the highest scale produces the part representation. Dashed lines mean unused connections and cells. **Mid:** The whole body is divided into multiple parts associated with keypoints. **Right:** For instance, right lower arm part representation is associated with the wrist and elbow keypoints. $\vec{v}_i$ and $\vec{v}_j$ mean two d-dim vectors respectively for wrist and elbow keypoints at location (x, y) of part representation feature map, and it's shape is $w^{'} \times h^{'} \times d \times 2$.

III. Pose Neural Fabrics Search

A. Overview

Based on top-down method, 2D human pose estimation aims to locate all K keypoints coordinates of body joints(e.g., shoulder, wrist, knee, etc) in a single pose. Let $S = \{(x_i, y_i)|x, y \in \mathbb{R}^+, i = 1, 2, ..., K\}$ denote the set consisting of all keypoints coordinates. Considering human body skeleton structures, we convert the whole body pose into P part representations. P subnetworks are constructed from cell-based neural fabrics sharing backbone to separately predict keypoints location subset s ($s \subseteq S$) whose element is associated with the corresponding part. The vector in pixel method is introduced to capture keypoint's feature in a relaxed vector space and the prediction of specified keypoint's location is determined by the location of vector $\vec{v}$ whose length is the largest. The entire framework for pose neural fabrics search is shown in Figure 2.

In section III-B, III-C, III-F, we demonstrate how to design and employ parameterized cell-based neural fabrics as the choice of subnetworks and backbone. Then, we carry out random sampling and synchronous optimization. In section III-E, we describe what prior structures of human body are employed to guide neural architecture search. In section III-D, we demonstrate how to utilize the vector representation to estimate keypoints' locations.

B. Parameterized Cell-based Neural Fabric

a) Micro structure: Cell is a repeatable unit across different layers and scales of the whole architecture. Illustrated in Figure 3, it receives outputs from previous cells as its single input node I and it has H nodes as it hidden states. Each hidden node h_j is connected by a directed edge with each element of candidate nodes set $\{h_0, h_1, h_2, ..., h_{j-1}\}$ $(h_0 = I, j = 1, 2, ..., H)$. Continuous Relaxation [28] method

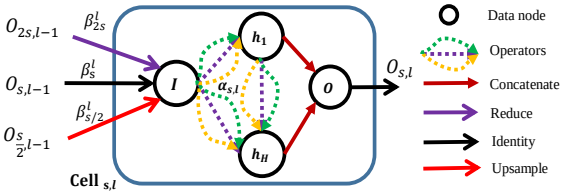

Fig. 3. An overview of inner structure of a cell in scale s and l layer. To simply present the inner structure of cell, we set the number of hidden nodes H as 2. Actually, its number of hidden nodes can be 1 and more than 2, each hidden node h_H is densely connected from its previous nodes $\{h_0, h_1, ..., h_H\}$.

is adopted to represent each directed edge with mixed operations. For each h_j is computed by:

$$h_j = \sum_{i=0}^{j-1} \sum_{o \in \mathcal{O}} \frac{\exp\left(\alpha_o^{(i,j)}\right)}{\sum_{o' \in \mathcal{O}} \exp\left(\alpha_{o'}^{(i,j)}\right)} o(h_i), \tag{1}$$

where $\mathcal{O}$ is the set of candidate operations. We choose 6 types of basic operations o, consisting of: zero, skip connection, 3×3 depthwise separable convolution, 3×3 dilated convolution with 2 rate, 3×3 average pooling, 3×3 max pooling. $\alpha_o^{(i,j)}$ means the associated weight for each operation $o \in \mathcal{O}$ in edge $h_i \to h_j$. For a specified $Cell_{s,l}$ in scale s and layer l in neural fabric, the continuous search space at micro level is: $\alpha_{s,l} = \left\{\alpha_o^{(i,j)}|\forall o \in \mathcal{O}, \forall j \in \{1, 2, ..., H\}, \forall i \in \{0, 1, ..., j-1\}\right\}$, $\alpha_{s,l} \in \mathbb{R}^{\|\mathcal{O}\| \times \frac{H(H+1)}{2}}$. Finally, all hidden nodes $\{h_1, h_2, ..., h_H\}$ are concatenated together and reduced in channels by a 1×1 conv to achieve an independent output node $O_{s,l}$.

b) Macro Structure: For $Cell_{s,l}$, it receives the sum of outputs from cells in previous layer: $O_{2s,l-1}$, $O_{\frac{s}{2},l-1}$ and $O_{s,l-1}$. They are associated with macro architecture parameters $\beta_{s,l} = \left\{\beta^l_{2s}, \beta^l_s, \beta^l_{\frac{s}{2}}\right\} \in \mathbb{R}^3$, which are normalized to

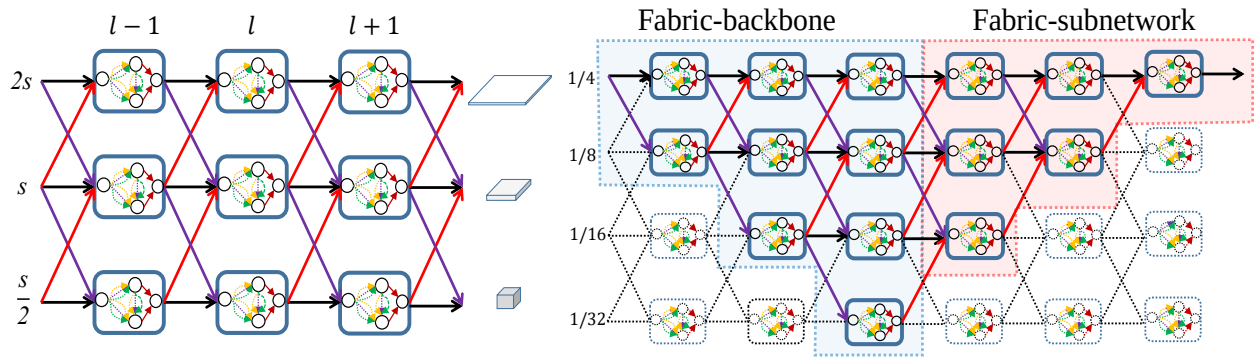

Fig. 4. An overview of cell-based neural fabric. **Left:** The homogeneous local connectivity between cells in a neural fabric. **Right**: Examples of constructing a Fabric-backbone (blue area) or a Fabric-subnetwork (red area) from cell-based neural fabric. Dashed lines mean unused connections and cells.

control different information reception level from previous cells in different scales. All $\beta_{s,l}$ in all cells of the fabric construct the macro continuous search space. For each $Cell_{s,l}$, its I is computed by:

$$I = h_0 = \sum_{(O,\beta)\in Z_{s,l}} \frac{\exp(\beta)}{\sum_{\beta'\in\beta_{s,l}} \exp(\beta')} \mathcal{T}(O), \qquad (2)$$

where $Z_{s,l} = \left\{(O_{2s,l-1}, \beta^l_{2s}), (O_{s,l-1}, \beta^l_s), (O_{\frac{s}{2},l-1}, \beta^l_{\frac{s}{2}})\right\}$ and $\mathcal{T}(\cdot)$ is scale transformation operation. In particular, $\mathcal{T}(O_{2s,l-1})$ is downsampling operation via Conv-BN-ReLU mode with 2 stride (meanwhile doubling the channels of data node), $\mathcal{T}(O_{\frac{s}{2},l-1})$ is upsampling operation via bilinear interpolation (meanwhile halving the channels of data node by 1×1 conv) and $\mathcal{T}(O_{s,l-1})$ means identity transformation.

c) Parametric Form: In summary, we parameterize the form of cell in the l-th layer and s-th scale of neural fabric in such a pattern:

$$O_{s,l} = \text{Cell}_{s,l}\left(O_{2s,l-1}, O_{s,l-1}, O_{\frac{s}{2},l-1}; w_{s,l}, \alpha_{s,l}, \beta_{s,l}\right), \quad (3)$$

where $w_{s,l}$ represents the weights of all operations in each cell, $\alpha_{s,l}$ and $\beta_{s,l}$ encode architecture search space inside the cell. The hyperparameter of each cell is $\theta_{s,l} = (H, C, \mathcal{O})$. H is the number of hidden nodes in each cell. C is the channel factor for each node to control the model capacity, i.e., the number of channels of its data node is $C \times \frac{1}{s}$. $\mathcal{O}$ is the set of candidate operations. This form also can be seen as a network generator [56], which can map from a parameter space Θ to a space of neural network architectures $\mathcal{N}$, formulated as $g : \Theta \longmapsto \mathcal{N}$.

C. Constructing Subnetworks or Backbone

Benefit from its local homogeneous connectivity pattern as shown in the left of Figure 4, cell-based fabric is very flexible and easy to extend into different layers and scales for high and low resource use cases. It is determined by a group of hyperparameters $\Theta = (L, \{1/1, 1/2, ..., 1/2^b\}, H, C, \mathcal{O})$ where L is total layers and $1/2^b$ is the smallest scale. Illustrated in Figure 4, fabric backbone can be constructed by reserving the first m layers and discarding the latter $L - m$ layers to produce feature pyramid in multiple scales. Likewise, fabric subnetwork is constructed by reserving the latter n layers and discarding the first $L - n$ layers to receive feature pyramid from backbone. Note that our backbone is not restricted to the proposed architecture, and we do not use 3×3 average pooling and 3×3 max pooling in candidate operations when constructing subnetworks as they are empirically more suitable for extract low-level visual feature.

Following common practice for pose estimation, the smallest scale is set to 1/32. We use a two-layer convolutional stem structure to firstly reduce the resolution to 1/4 scale, and consecutively weave the whole neural fabric [2]. In order to achieve a higher resolution feature map to locate keypoints' coordinates, we just use the final cell's output in 1/4 scale as the part representation. Finally, we use P subnetworks (multi-head) sharing backbone to produce P part representations. The long-range and short-range constraint relationships of the human pose are enforced by the whole architecture in an end-to-end learning method.

D. Body Part Representation with Vector in Pixel

Based on the top-down method of pose estimation, we estimate human pose with single person proposal. Given a input image $I \in \mathbb{R}^{H\times W\times 3}$ containing person proposal, there will be P part representations $T_1, T_2, ..., T_P$ to be predicted. Let $T_p \in \mathbb{R}^{h^{'}\times w^{'}\times d\times J}$ denote its p-th body part representation, where J is the number of keypoints belonging to this part, $h^{'}$ and $w^{'}$ are the height and width of part representation, in our setting, $h^{'}/H = w^{'}/W = 1/4$. d is the dimension of vector in each pixel position. For i-th keypoints of T_p, vector in pixel (x, y) is denoted as $\vec{v}_{i,x,y} = T_p(i, x, y) \in \mathbb{R}^d$, simplified as $\vec{v}$. Note that the dimension d of vector is set to 8 by default and choice for dimension is discussed in section VI .

We relax scalar value into a latent vector as keypoint entity in each image pixel, expecting it to implicitly capture more local feature information of keypoint. Besides inherent to the characteristics of encoding locations, $\vec{v}$ can represent existing probability of keypoints by using Squashing Function [44] to

[2]For cells in first layer, they only receive the stem's output. And for cells in 1/32 scale or 1/4 scale of its current layer, it may have only have two outputs from previous cells. In this case, we will copy one of candidate inputs.

normalize its ℓ_2 norm to $[0, 1)$. Formally, for i-th keypoint of p part, the squashed vector $\vec{v}_s$ in position (x, y) is computed by:

$$\vec{v}_s = \frac{\|\vec{v}\|^2}{1 + \|\vec{v}\|^2} \frac{\vec{v}}{\|\vec{v}\|}, \tag{4}$$

where $\|\vec{v}_s\|$ exactly represents i-th keypoint's existing score in position (x, y). The position $(\bar{x}, \bar{y})$ of the longest $\vec{v}$ will be regarded as keypoint location in inference. Predicted score maps are achieved by all part presentations. Moreover, groundtruth score maps are generated from groundtruth keypoints positions by applying 2D Gaussian with deviation of σ where the peak value equals 1 and σ controls the spread of the peak. Train loss $\mathcal{L}_{train}$ is computed by Mean Square Error (MSE) between the predicted score maps and groundtruth score maps for all P part representations. In some cases, skeleton joint like elbow maybe fall into multiple parts, and thus its final position $(\bar{x}, \bar{y})$ will be summed from predictions of these parts.

The extra advantage of vector in pixel is that ambiguity between image feature and groundtruth position can be reduced. In supervised learning, the difficulty of fitting label is usually not under consideration, hard or easy samples of the same category receive same level of supervision. This issue occurs in keypoints localization because image appearance varies in some partially occluded areas where the labeled keypoint feature has been disturbed (e.g. the first image in the Figure 8, the man's ankle is occluded by a dog, but his ankle's position is labeled). In such case, our method can handle it as $\|\vec{v}_s\|$ replaces $\|\vec{v}\|$ under supervision (element value of each dimension of vector has no explicit property and is unsupervised) and the expected length for $\vec{v}$ in groundtruth keypoint pixel is not directly supervised by numerical value p from groundtruth score but supervised by $\sqrt{\frac{1}{1-p} - 1} \in [0, +\infty)$ where $p \in [0, 1)$. In a slight abuse of notation, we write $\|\vec{v}_*\|$ as the expected length of $\vec{v}$, which provides a relatively loose range space for $\vec{v}$, even if under strong supervision (more explanation in Appendix A).

E. Prior Knowledge of human body structure

Considering the human body prior structures, we adopt four types of grouping strategy. Specially, $P = 1$ means that we model long-range dependencies relationships of pose and the relationship of all joints is learned globally. $P = 3$ means that body pose is predefined into three parts: *head part, upper limb part and lower limb part*. $P = 8$ means that body pose is predefined into eight parts: *head-shoulder, left upper arm, left lower arm, right upper arm, right lower arm, thigh, left lower leg and right lower leg*. In addition, we also adopt the data-driven grouping strategy of [51] by setting $P = 5$: *head-shoulder, left lower arm, right lower arm, thigh, lower limb part*. In table I, all skeleton keypoints are associated with the corresponding part according to the body structure. For MPII [1] dataset, we set indices of head top, upper neck, thorax, left shoulder,right shoulder, left elbow, right elbow, left wrist, right wrist, left hip, right hip, left knee, right knee, left ankle, right ankle, pelvis keypoints to 0-15 orderly. For COCO [26] dataset, we set indices of nose, left eye, right eye, left ear, right ear, left shoulder, right shoulder, left elbow, right elbow, left wrist, right wrist, left hip, right hip, left knee, right knee, left ankle and right ankle to 0-16 orderly.

TABLE I
ACCORDING TO THE PRIOR KNOWLEDGE OF HUMAN BODY STRUCTURE, THERE ARE DIFFERENT GROUPING TYPES OF BODY PART REPRESENTATIONS.

Representation Mode	Group Name	Index	
		MPII	COCO
$P = 1$	all keypoints	0-15	0-16
$P = 3$	head part	0-2	0-4
	upper limb part	3-8	5-10
	lower limb part	9-15	11-16
$P = 5$	head-shoulder	0-4	0-6
	left lower arm	5,7	7,9
	right lower arm	6,8	8,10
	thigh	9,10,15	11,12
	lower limb part	11-14	13-16
$P = 8$	head-shoulder	0-4	0-6
	left upper arm	3,5	5,7
	left lower arm	5,7	7,9
	right upper arm	4,6	6,8
	right lower arm	6,8	8,10
	thigh	9-12,15	11-14
	left lower leg	11,13	13,15
	right lower leg	12,14	14,16

F. Optimization

a) One-shot Search and Personalized Structures for Different Parts: Given a hyperparameter Θ for a cell-based neural fabric, the weights $w_o = \{w_{s,l}\}$ and architecture $\alpha_o = \{\alpha_{s,l}\}, \beta_o = \{\beta_{s,l}\}$ are optimized. Following the principle of one-shot architecture search [11], we assume that $\alpha_{s,l}$ share same weights across a neural fabric and $\beta_{s,l}$ is cell-wise in a neural fabric. Considering P multiple neural fabrics, their weights of operations are $w = \{w_0, ..., w_P\}$ and the total architecture parameters are $\alpha = \{\alpha_1, ..., \alpha_P\}, \beta = \{\beta_1, ..., \beta_P\}$. We search for personalized neural fabric to adapt each part of body, which means that the architecture parameters $\alpha_1, ..., \alpha_P$ are totally different and so are $\beta_1, ..., \beta_P$. In the section IV-B0c, we make contrastive experiments to study the performances by setting different the body part representation modes.

b) Random Sampling: Random search can be seen as a powerful baseline for neural architecture search or hyperparameter optimization [4], [24], [56]. It is conducted in [28] as well and has a competitive result compared with the gradient-based method. In this work, we randomly initialize values of α, β by standard normal distribution and make them fixed in the whole training process like [56]. From another point of view, this also can be viewed as a *stochastic network generator*. The sampled architecture parameters make no assumption about the structures, and only the weights of neural networks are optimized. Therefore, we make it as our baseline model to validate the design of cell-based neural fabrics, which also represents the performances of cell-based

neural fabrics without any other neural architecture search strategies.

c) Synchronous Optimization: In DARTS [28], the architecture search problem is regarded as a bilevel optimization problem. An extra subset val of original train set is held out serving as performance validation to produce the gradient w.r.t. architecture parameters α, β. However, this method is still time-consuming and restricted by computing resources because the gradient-based method with second-order term consumes much time and GPU memory for pose estimation due to its high resolution representation. Moreover, the training for parent continuous network and the derived net are inconsistent in DARTS, final pruned network needs to be trained with all training samples. Benefit from the parametric form of cell, we explore a more simple way as our major optimization strategy. The α and β are registered to model's parameters, synchronously optimized by the same optimizer of the weights w, i.e. the w, α and β are updated by $\nabla_{w,\alpha,\beta}\mathcal{L}_{train}$ in a single step of gradient descent. Without extra validation performance, the final continuous α, β and the weights w have seen all training samples, thus it do not need to be pruned into discrete architecture by $argmax$ operation and trained again from scratch.

IV. Experiments

A. Implementation Details

As for subnetworks, we set C=10 and total final layers is 3 (discarding first 3 layers of neural fabric architecture with $L = 6$), and the total number of cells is 6 as a basic configuration. Backbone with fabric can be constructed as described in section III-C. To make fair comparison with methods using model pretrained on ImageNet [43], we take Fabric-1, Fabric-2, Fabric-3, pretrained Mobilenet-V2 [45] and ResNet-50 [17] feature blocks (5.8M, 6.8M, 10.5M, 1.3M and 23.5M parameters respectively) as choices for backbone to provide feature pyramid to subnetworks. The trade-off curves between performance and inference complexity (Madds, Multiply-Adds of operations) on MPII validation are shown in Figure 5.

We implement our work by PyTorch [37] and each experiment is conducted on a single NVIDIA Titan Xp GPU. Training epoch is 200 and batchsize is set to 24 (not fixed). We use Adam [22] optimizer to update the weights and architecture parameters with 0.001 initial learning rate, decay at epoch 90, 120, 150 with 0.25 factor. Data augmentation strategies are used with random rotation range in $[-45°, 45°]$, random scale range in $[0.7, 1.3]$ and random flipping with 0.5 probability. Flip test is used in inference. In practice, one quarter offset from the peak to the secondary peak is introduced to reduce the quantization error. Strategies mentioned above are adopted in all experiments.

B. Ablation Study

a) Dataset and Evaluation: We conduct ablation study on MPII Human Pose Dataset [1] which is a benchmark for evaluation of pose estimation. The dataset consist of around 25K images containing over 40K people with annotated body joints. All models in ablation study experiments are trained on a subset MPII training set and evaluate on a held validation set of 2958 images following [55]. The standard PCKh metric (head-normalized probability of correct keypoint) is used for MPII. PCKh@0.5 means that a predicted joint is correct if its position is within 50% of the length groudtruth head box from its groundtruth location. Evaluation procedure reports the PCKh@0.5 of head, shoulder, elbow, wrist, hip, knee, ankle, mean PCKh@0.5 and mean PCKh@0.1.

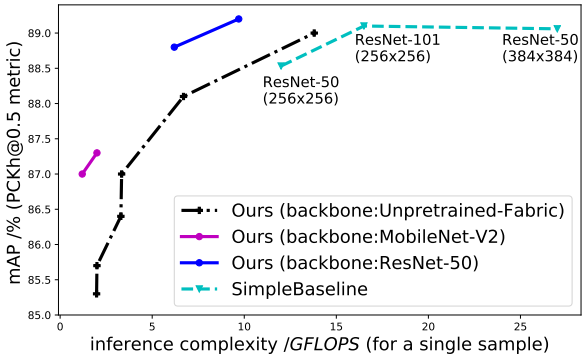

Fig. 5. The mAP of PCKh@0.5 metric vs. model inference complexity (GFLOPs) on MPII.

TABLE II
Optimization Strategies (search method). We choose MobileNet-V2 as the backbone of model. $P = 3, H = 1, C = 10, d = 8$, each subnetwork has six cells and total parameters of model is 3.3M, the Madds of model inference complexity for single input sample is 1.2 GFLOPs.

Search Method (search strategy)	**Search Time** (GPU days)	**Mean** (PCKh@0.5)	**Mean** (PCKh@0.1)
Random	0.8	87.1±0.2	35.2±0.4
First-order gradient-based	0.9	87.1	34.7
Synchronous gradient-based	0.8	87.0	34.6

b) Optimization Strategies: For random search strategy, we conduct 5 experiments with different pseudo random seeds, each experiment costs 0.8 days for a single GPU. Result shows that completely random architecture parameters can perform well. As shown in Table II, synchronous optimization is effective as well as random search under the same configuration and search time. We observe that the best result of random initialization for architecture surpasses the synchronous optimization, this reveals that search space design has more crucial impact on the performance of neural

TABLE III
Body Part Representation Modes. We choose MobileNet-V2 as the backbone of model. $H = 1, C = 10, d = 8$, each subnetwork has six cells.

Representation Mode	**Mean**(PCKh@0.5)	**Mean**(PCKh@0.1)
$P = 1$	86.4	33.4
$P = 3$	87.0	34.6
$P = 5$	87.1	35.1
$P = 8$	87.3	35.6

TABLE IV
COMPARISONS OF PERFORMANCE, MODEL PARAMETERS AND INFERENCE COMPLEXITY ON MPII TEST SET

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total	# Params	# FLOPs
Tompson et al. [53]	95.8	90.3	80.5	74.3	77.6	69.7	62.8	79.6	-	-
Belagiannis & Zisserman [3]	97.7	95.0	88.2	83.0	87.9	82.6	78.4	88.1	-	-
Wei et al. [54]	97.8	95.0	88.7	84.0	88.4	82.8	79.4	88.5	-	-
Insafutdinov et al. [21]	96.8	95.2	89.3	84.4	88.4	83.4	78.0	88.5	42.6M	41.2G
Bulat&Tzimiropoulos [6]	97.9	95.1	89.9	85.3	89.4	85.7	81.7	89.7	-	-
Newell et al. [31]	98.2	96.3	91.2	87.1	90.1	87.4	83.6	90.9	25.1M	19.1G
Xiao et al. [55]	**98.5**	96.6	91.9	87.6	91.1	88.1	84.1	91.5	68.6M	20.9G
Tang et al. [52]	98.4	**96.9**	**92.6**	**88.7**	**91.8**	**89.4**	**86.2**	**92.3**	15.5M	33.6G
Ours	97.9	95.6	90.7	86.5	89.8	86.0	81.5	90.2	**5.2M**	**4.6G**

TABLE V
COMPARISON WITH HRNET BASELINE ON MPII VALIDATION SET WITH INPUT SIZE IS 256×256. WE CHOOSE HRNET-W32-STEM∼STAGE3 AS THE BACKBONE OF MODEL. OUR-A: $P = 3, H = 2, C = 12$. OUR-B: $P = 5, H = 1, C = 16$.

Method	**Backbone**	**Head**	**Mean**(PCKh@0.5)	**Mean**(PCKh@0.1)
SimpleBaseline [55]	ResNet-152 (52.2M)	transpose conv layers (16.4M)	89.6	35.0
HRNet-32 [47]	stem∼stage3 (8.8M)	stage4 (19.7M)	**90.3**	37.7
Ours-a	stem∼stage3 (8.8M)	neural fabrics (**4.8M**)	89.9	**39.5**
Ours-b	stem∼stage3 (8.8M)	neural fabrics (7.6M)	90.1	39.4

network. In addition, we implement the first-order gradient-based optimization method according to the official code[3] of DARTS [28] for comparison. We hold out half of MPII training data as validation for performance estimation of architecture. Another Adam optimizer is used to update α, β with 0.003 learning rate and 0.001 weight decay, discrete architecture is not derived from continuous architecture for full training.

TABLE VI
DIMENSION CHOICES FOR VECTOR IN PIXEL. WE CHOOSE MOBILENET-V2 AS THE BACKBONE OF MODEL. $P = 3, H = 1, C = 10$, EACH SUBNETWORK HAS SIX CELLS AND TOTAL PARAMETERS OF MODEL IS 3.3M, THE MADDS (FLOPS) OF MODEL INFERENCE COMPLEXITY FOR SINGLE INPUT SAMPLE IS 1.2 GFLOPS.

Dimension	**Mean**(PCKh@0.5)	**Mean**(PCKh@0.1)	**#Params**	**#Madds**(FLOPs)
$d = 1(scalar)$	86.8	33.5	3.3M	1.1G
$d = 4$	86.9	34.9	3.3M	1.1G
$d = 8$	87.0	34.6	3.3M	1.2G
$d = 16$	86.8	34.9	3.3M	1.2G
SimpleBaseline [55]	88.5	33.9	34.0M	12.0G
+ vector(8-dim)	88.7	34.2	34.0M	12.1G
HRNet [47]	90.3	37.7	28.5M	9.5G
+ vector(8-dim)	90.2	38.1	28.5M	9.6G

c) Body Part Representation Modes: We study these four modes predefined by the prior knowledge and results are shown in Table III. We choose MobileNet-v2 [45] feature blocks as backbone. We find that multiple part presentations surpasses global whole-body representation. 8 part representations mode achieve 1% accuracy increase than whole-body representation and mode with $P = 3$ is a trade-off between performance and model capacity. Furthermore, we make a combination with the strong baseline HRNet [47]. We use the ImageNet pretrained HRNet-W32 by replacing blocks of the last stage with multiple subnetworks (neural fabrics), as a result the number of backbone's parameters is 8.8M. The configurations for fabrics are: $P = 5, H = 1, d = 8, C = 16$ and

[3]https://github.com/quark0/darts/blob/master/cnn/architect.py

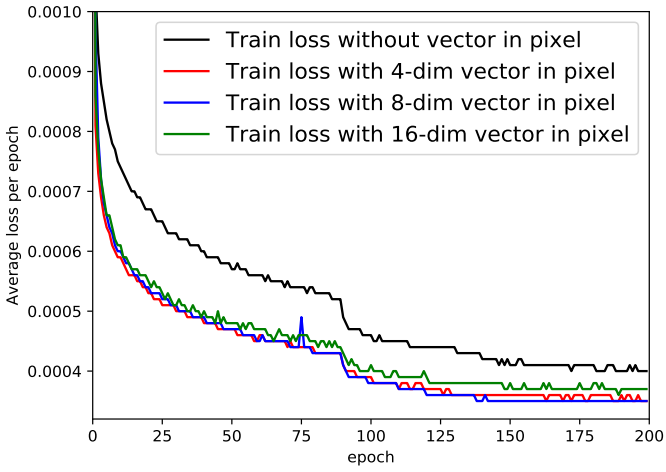

Fig. 6. Train losses of experiments w and w/o vector in pixel method. Detailed configurations are described in Table VI. The sudden drop at epoch 90 is caused by learning rate decay.

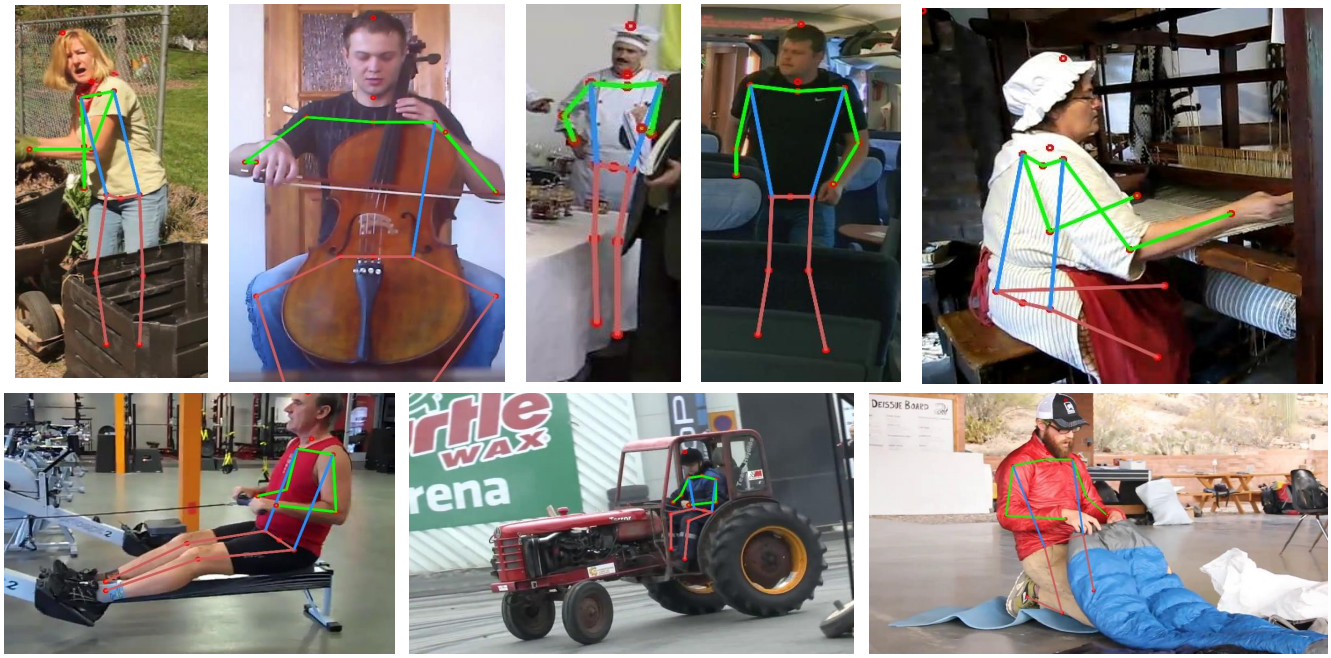

Fig. 7. Qualitative pose estimation results on MPII val set for single person pose estimation. We show the cropped image regions containing human body.

Fig. 8. Qualitative pose estimation results on COCO val2017 set. Estimation is conducted on bounding boxes detected by Faster-RCNN [40]. It is worth noting that our method works well in some heavily partial occluded hard samples (such as left two images in first row and the fourth in second row).

$\mathcal{O}$ includes zero, skip connection, 3×3 separable conv, 3×3 dilated conv with 2 rate. We adopt synchronous optimization and the input size of net is 256×256. As shown in table V, with 16.4M model size and 9.4 GFLOPs, our model achieves 90.1%/39.4% accuracy in PCKh@0.5/PCKh@0.1 metrics, in contrast to 90.3%/37.7% reported from [47] by HRNet-W32 (28.5M, 9.5G FLOPs) in single-scale testing.

d) Dimension Choices for Vector in Pixel: We study the effect of choice for dimension d of the vector on performance by setting d with $4, 8, 16$. We find that 8-dim vector has a better performance shown in Table VI. To validate the generalization of 8-dim vector representation method, we apply it to SimpleBaseline [4] [55] and HRNet [5] [47]. We find that this 8-dim vector representation is effective in these two frameworks. It gains 0.23% increase in PCKh@0.5 and 0.88% increase in PCKh@0.1 than SimpleBaseline official results with little increase of complexity and 1.06% increase in PCKh@0.1 than HRNet official results. We find that there is no obvious boost on PCKh@0.5 metric but a little in PCKh@0.1. In Figure 6, we observe that the losses of training with vector in pixel method converge faster, which implies the fitting between training data and label is more robust.

[4]https://github.com/microsoft/human-pose-estimation.pytorch
[5]https://github.com/leoxiaobin/deep-high-resolution-net.pytorch

TABLE VII
COMPARISONS OF PERFORMANCE, MODEL PARAMETERS AND INFERENCE COMPLEXITY ON COCO TEST-DEV SET. MODEL PARAMETERS AND FLOPS OF DETECTING PERSONS ARE NOT INCLUDED.

	AP	AP^{50}	AP^{75}	AP^M	AP^L	AR	AR^{50}	AR^{75}	AR^M	AR^L	#Params	# FLOPs
CMU-Pose [7]	0.618	0.849	0.675	0.571	0.682	-	-	-	-	-	-	-
Mask-RCNN [16]	0.631	0.873	0.687	0.578	0.714	-	-	-	-	-	-	-
Associative Embedding [30]	0.655	0.868	0.723	0.606	0.726	0.702	0.895	0.760	0.646	0.78	-	-
Integral Pose Regression [50]	0.678	0.882	0.748	0.639	0.74	-	-	-	-	-	45.0M	11.0G
SJTU [12]	0.680	0.867	0.747	0.633	0.750	0.735	0.908	0.795	0.686	0.804	-	-
G-RMI [35]	0.685	0.871	0.755	0.658	0.733	0.733	0.901	0.795	0.681	0.804	42.6M	57.0G
PersonLab [34]	0.687	0.890	0.754	0.641	0.755	0.754	0.927	0.812	0.697	0.830	-	-
MultiPoseNet [23]	0.696	0.863	0.766	0.650	0.763	0.735	0.881	0.795	0.686	0.803	-	-
CPN [9]	0.721	0.914	0.800	0.687	0.772	0.785	0.951	0.853	0.742	0.843	-	-
SimpleBaseline(ResNet-50) [55]	0.702	0.909	0.783	0.671	0.759	0.758	-	-	-	-	34M	8.9G
SimpleBaseline(ResNet-152) [55]	0.737	0.919	0.811	0.703	0.800	0.790	-	-	-	-	68.6M	35.6G
HRNet-W32 [47]	0.749	0.925	0.828	0.713	0.809	0.801	-	-	-	-	28.5M	16.0G
HRNet-W48 [47]	**0.755**	**0.925**	**0.833**	**0.719**	**0.815**	**0.805**	-	-	-	-	63.6M	32.9G
Ours-1	0.674	0.890	0.737	0.633	0.743	0.731	0.928	0.791	0.681	0.800	**6.1M**	**4.0G**
Ours-2	0.709	0.904	0.777	0.667	0.782	0.766	0.941	0.829	0.715	0.836	27.5M	11.4G

C. Comparison with the state-of-the-art

a) Testing on MPII Single Person Pose Estimation: To further evaluate our pose estimation method on test set of MPII [1], we train the model on all samples of MPII train set with early-stopping strategy. All input images are resized to 384×384 pixels, data augmentation is the same as mentioned above. We use pretrained MobileNet-v2 [45] feature blocks as backbone. The hyperparameters are: $P = 3, H = 2, d = 8, C = 10$ and $\mathcal{O}$ includes zero, skip connection, 3×3 separable conv, 3×3 dilated conv with 2 rate. The optimization method is synchronous optimization. With 5.2M total parameters and 4.6G Madds of operations, we achieve comparable results with the-state-of-art performance as shown in Table IV. Prediction results on some partial occluded hard samples can be seen in Figure 7.

b) COCO Keypoint Detection Task: MS-COCO [26] dataset contains more than 200k images and 250k person instances with keypoints label. We use COCO train2017 as our training set, it consists of 57k images and 150k person instances. Val2017 set contains 5k images and test-dev2017 consists of 20k images. It is worth mentioning that some invisible keypoints are labeled on train set and statistics show that around 11.3 % of annotated keypoints are invisible according to train2017 annotations. Object keypoint similarity (OKS) is the standard evaluation metric for keypoints locating accuracy. More detailed information is available in COCO official website [6].

COCO keypoint detection task involves detecting bodies and localizing their keypoints. Based on top-down method, we focus on single pose estimation, therefore we use the detected bounding boxes detected by Faster-RCNN [40] with 60.9 AP persons detection results on COCO test-dev2017 dataset. We respectively use pretrained MobileNet-v2 [45] and ResNet-50 [17] feature blocks as backbone and train two models only on train2017 set. The hyperparameters are: $P = 3, H = 1, d = 8$ and $\mathcal{O}$ includes zero, skip connection, 3×3 separable conv, 3×3 dilated conv with 2 rate. $C = 16/10$ for ours-1/ours-2 models. The optimization method is synchronous optimization. The input size is 384×288 pixels and the OKS-NMS algorithm [35] is utilized to suppress redundant detected bounding boxes . We report average precision (AP) and average recall (AR) on COCO test-dev2017 set. As shown in Table VII, with fewer parameters and low computational complexity, we can achieve a comparable result with state-of-the-art performance without any extra data or ensemble models. Prediction results on some partial occluded hard samples can be seen in Figure 8.

[6]http://cocodataset.org/

V. CONCLUSION

In this work, we made the first attempt to exploit prior knowledge of human body structure to guide NAS for human pose estimation task, and meantime proposed Pose Neural Fabrics Search (PNFS) framework. Experiment results showed that our light-weight models achieved comparable results on MPII dataset with 70%$\sim$ 80% fewer parameters and 80%$\sim$ 90% lower computational complexity than state-of-the-art methods. For more challenging COCO keypoint detection task, our model attained comparable results to state-of-the-art methods with fewer parameters and lower complexity. This actually reveals the existence of parameter redundancy phenomenon in the current human pose estimation systems based on large model capacity, which to some extent can be attributed to failing to exploit the prior structural knowledge of the human body. Actually, our model is still over-parameterized, the size of which also has a potential space to be further reduced

by model pruning. In addition, we empirically demonstrate the effectiveness of representing the human body keypoints as vector entities at image locations. We hope that these ideas may be helpful to adequately leverage NAS to practical application for human pose estimation task or other domains.

APPENDIX A
EXPLANATION FOR THE RELATIONSHIP BETWEEN EXPECTED $\|\vec{v}\|$ AND SUPERVISION LEVEL p

In the section III-D, we use Squash Function to normalize $\vec{v}$ to $\vec{v}_s$ whose length ranges in $[0, 1)$,

$$\vec{v}_s = \frac{\|\vec{v}\|^2}{1 + \|\vec{v}\|^2} \frac{\vec{v}}{\|\vec{v}\|}, \tag{5}$$

$$\|\vec{v}_s\| = \frac{\|\vec{v}\|^2}{1 + \|\vec{v}\|^2}, \tag{6}$$

$$\|\vec{v}\| = \sqrt{\frac{1}{1 - \|\vec{v}_s\|} - 1}, \tag{7}$$

where $\|\vec{v}_s\|$ is supervised by numerical value p in each pixel position from groundtruth score maps. Ideal value of $\|\vec{v}_s\|$, denoted as $\|\vec{v}_*\|$, equals to $p \in [0, 1)$. Therefore, $\|\vec{v}\|$ is supervised by $\sqrt{\frac{1}{1-p} - 1}$.

REFERENCES

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition (CVPR)*, pages 3686–3693, 2014.

[2] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1014–1021. IEEE, 2009.

[3] Vasileios Belagiannis and Andrew Zisserman. Recurrent human pose estimation. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 468–475. IEEE, 2017.

[4] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research (JMLR)*, 13(Feb):281–305, 2012.

[5] Elie Bienenstock, Stuart Geman, and Daniel Potter. Compositionality, mdl priors, and object recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 838–844, 1997.

[6] Adrian Bulat and Georgios Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 717–732. Springer, 2016.

[7] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7291–7299, 2017.

[8] Liang-Chieh Chen, Maxwell Collins, Yukun Zhu, George Papandreou, Barret Zoph, Florian Schroff, Hartwig Adam, and Jon Shlens. Searching for efficient multi-scale architectures for dense image prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8699–8710, 2018.

[9] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7103–7112, 2018.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. Ieee, 2009.

[11] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *arXiv preprint arXiv:1808.05377*, 2018.

[12] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2334–2343, 2017.

[13] Pedro F Felzenszwalb and Daniel P Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision (IJCV)*, 61(1):55–79, 2005.

[14] Pedro F Felzenszwalb, David A McAllester, Deva Ramanan, et al. A discriminatively trained, multiscale, deformable part model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, page 7, 2008.

[15] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V. Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[18] Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with EM routing. In *International Conference on Learning Representations*, 2018.

[19] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.

[20] Eldar Insafutdinov, Mykhaylo Andriluka, Leonid Pishchulin, Siyu Tang, Evgeny Levinkov, Bjoern Andres, and Bernt Schiele. Arttrack: Articulated multi-person tracking in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6457–6465, 2017.

[21] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 34–50. Springer, 2016.

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[23] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. Multiposenet: Fast multi-person pose estimation using pose residual network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 417–433, 2018.

[24] Liam Li and Ameet Talwalkar. Random search and reproducibility for neural architecture search. *arXiv preprint arXiv:1902.07638*, 2019.

[25] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision (ECCV)*, pages 740–755. Springer, 2014.

[27] Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan Yuille, and Li Fei-Fei. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. *arXiv preprint arXiv:1901.02985*, 2019.

[28] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

[29] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[30] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2277–2287, 2017.

[31] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 483–499. Springer, 2016.

[32] X. Nie, J. Feng, J. Xing, S. Xiao, and S. Yan. Hierarchical contextual refinement networks for human pose estimation. *IEEE Transactions on Image Processing*, 28(2):924–936, Feb 2019.

[33] Xuecheng Nie, Jiashi Feng, Jianfeng Zhang, and Shuicheng Yan. Single-stage multi-person pose machines. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[34] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[35] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. Towards accurate multi-person pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4903–4911, 2017.

[36] Seyoung Park, Bruce Xiaohan Nie, and Song-Chun Zhu. Attribute and-or grammar for joint parsing of human pose, parts and attributes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(7):1555–1569, 2017.

[37] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[38] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V. Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4929–4937, 2016.

[39] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. *arXiv preprint arXiv:1802.01548*, 2018.

[40] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 91–99, 2015.

[41] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[42] Matteo Ruggero Ronchi and Pietro Perona. Benchmarking and error diagnosis in multi-instance pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 369–378, 2017.

[43] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision (IJCV)*, 115(3):211–252, 2015.

[44] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3856–3866, 2017.

[45] Mark B. Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4510–4520, 2018.

[46] Shreyas Saxena and Jakob Verbeek. Convolutional neural fabrics. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4053–4061, 2016.

[47] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[48] Min Sun and Silvio Savarese. Articulated part-based model for joint object detection and pose estimation. In *2011 International Conference on Computer Vision (ICCV)*, pages 723–730. IEEE, 2011.

[49] Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. Compositional human pose regression. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2602–2611, 2017.

[50] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 529–545, 2018.

[51] Wei Tang and Ying Wu. Does learning specific features for related parts help human pose estimation? In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[52] Wei Tang, Pei Yu, and Ying Wu. Deeply learned compositional models for human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 190–206, 2018.

[53] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1799–1807, 2014.

[54] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4732, 2016.

[55] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 466–481, 2018.

[56] Saining Xie, Alexander Kirillov, Ross Girshick, and Kaiming He. Exploring randomly wired neural networks for image recognition. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[57] Sirui Xie, Hehui Zheng, Chunxiao Liu, and Liang Lin. Snas: stochastic neural architecture search. *arXiv preprint arXiv:1812.09926*, 2018.

[58] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang. Denseaspp for semantic segmentation in street scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3684–3692, 2018.

[59] Wei Yang, Shuang Li, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Learning feature pyramids for human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1281–1290, 2017.

[60] L. Zhao, X. Gao, D. Tao, and X. Li. Tracking human pose using max-margin markov models. *IEEE Transactions on Image Processing*, 24(12):5274–5287, Dec 2015.

[61] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *arXiv preprint arXiv:1904.07850*, 2019.

[62] Barret Zoph and Quoc V. Le. Neural architecture search with reinforcement learning. *International Conference on Learning Representations (ICLR)*, 2017.

[63] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8697–8710, 2018.