Layout-Graph Reasoning for Fashion Landmark Detection

Weijiang Yu[1], Xiaodan Liang[1,2*], Ke Gong[1,2], Chenhan Jiang[1],
Nong Xiao[1], Liang Lin[1,2]
[1]Sun Yat-sen University, [2]DarkMatter AI Research
weijiangyu8@gmail.com, xdliang328@gmail.com, kegong936@gmail.com,
jchcyan@gmail.com, xiaon6@sysu.edu.cn, linliang@ieee.org

Abstract

Detecting dense landmarks for diverse clothes, as a fundamental technique for clothes analysis, has attracted increasing research attention due to its huge application potential. However, due to the lack of modeling underlying semantic layout constraints among landmarks, prior works often detect ambiguous and structure-inconsistent landmarks of multiple overlapped clothes in one person. In this paper, we propose to seamlessly enforce structural layout relationships among landmarks on the intermediate representations via multiple stacked layout-graph reasoning layers. We define the layout-graph as a hierarchical structure including a root node, body-part nodes (e.g. upper body, lower body), coarse clothes-part nodes (e.g. collar, sleeve) and leaf landmark nodes (e.g. left-collar, right-collar). Each Layout-Graph Reasoning(LGR) layer aims to map feature representations into structural graph nodes via a Map-to-Node module, performs reasoning over structural graph nodes to achieve global layout coherency via a layout-graph reasoning module, and then maps graph nodes back to enhance feature representations via a Node-to-Map module. The layout-graph reasoning module integrates a graph clustering operation to generate representations of intermediate nodes (bottom-up inference) and then a graph deconvolution operation (top-down inference) over the whole graph. Extensive experiments on two public fashion landmark datasets demonstrate the superiority of our model. Furthermore, to advance the fine-grained fashion landmark research for supporting more comprehensive clothes generation and attribute recognition, we contribute the first Fine-grained Fashion Landmark Dataset (FFLD) containing 200k images annotated with at most 32 keypoints for 13 clothes types.

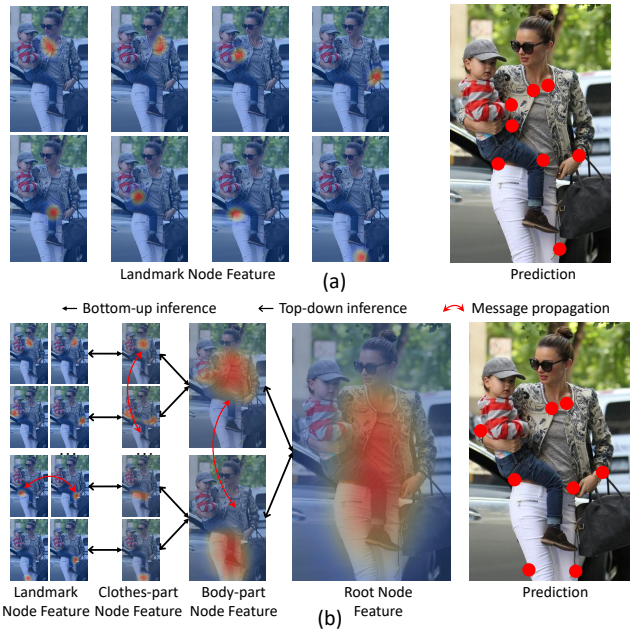

Figure 1: **(a)** Traditional DCNNs [25] suffer from great performance drops when facing heavy overlapping of humans and clothes nesting because of lacking structural constraint and commonsense knowledge. (*e.g.* the left waistline and right sleeve on the lady are wrongly predicted on the child.) **(b)** Our LGR provides graph-based inferences among landmarks, leveraging symmetric and hierarchical relations to constrain landmark's layout in one person.

1. Introduction

Fashion landmark detection that targets at localizing keypoints at the functional regions of clothes (*e.g.* collar, waistline), has attracted research attention and lots of demands driven by the boom of electronic commerce, such as clothing retrieval [36, 30, 5, 16], clothes generation [10, 21] and fashion image classification [36, 27, 8]. To support these comprehensive high-level applications, the landmark detectors need to effectively deal with arbitrary clothing appearances, diverse clothes layouts and styles, multiply human

*Corresponding Author

pose, different lights and background clutters.

Recent research efforts on fashion landmark detection [18, 26, 36, 31, 14, 32, 19] are mainly devoted to designing more advanced deep feature representations [18, 36], attention mechanism [31], pyramid module [32] and so on. These models are limited in regarding fashion landmark detection as an end-to-end regression problem and ignore rich semantic layout relations among different landmarks, such as symmetric relation (*e.g*. left/right collar), subordinate relation (*e.g*. left collar belongs to collar) and human commonsense (*e.g*. one clothing generally owns a pair of sleeves). Consequently, unreasonable detection results that deviate from human and clothes layouts may be generated, as shown in Figure 1(a).

Nevertheless, some researches have resorted to external guidance to enhance the interpretation of CNNs [24, 2, 33, 27, 15, 20]. For example, Yang *et al*. [33] introduced to incorporate domain knowledge for explicitly facilitating features for better localization. Wang *et al*. [27] proposed a grammar model in fashion visual understanding and used a bidirectional recurrent neural network for message passing. Fashion landmarks naturally lie in an underlying hierarchical structure which includes different levels of semantic nodes (e.g. body-part nodes, clothes-part nodes and leaf landmark nodes). However, they used a plain graph structure to represent knowledge and disregarded the intrinsic hierarchical and multi-level layouts of landmarks for better mining subordinate relations.

To address all above-mentioned issues, we propose to endow the deep networks with the capability of structural graph reasoning in order to make detected fashion landmarks be coherent with human and clothes layouts from a global perspective. We define a hierarchical layout-graph that encodes prior commonsense knowledge in terms of human body part layouts and clothes part layouts, consisting of a root node, body-part nodes (*e.g*. upper body, lower body), coarse clothes-part nodes (*e.g*. collar, sleeve) and leaf landmark nodes (*e.g*. left-collar, right-collar). We then propose a novel Layout-Graph Reasoning (LGR) layer that is able to explicitly enforce hierarchical human-clothes layout constraints and semantic relations of fashion landmarks on deep representations for facilitating landmark detection. Our LGR layer is a general and flexible network layer which can be stacked and injected between any convolution layers, containing three modules: 1) a Map-to-Node module that maps convolutional features into each graph leaf node; 2) a layout-graph reasoning module to perform hierarchical graph reasoning over structural graph nodes to achieve global layout coherency; 3) a Node-to-Map module to learn appropriate associations between the evolved graph leaf nodes and convolutional features, which in turn enhance local feature representations with global reasoning.

Given graph node representations for leaf landmark nodes from the Map-to-Node module, our layout-graph reasoning module first performs a graph clustering operation to generate representations of intermediate nodes in the spirit of bottom-up inference, that is, propagating from (leaf landmark nodes)→(clothes-part nodes)→(body-part nodes)→(root node). Then a graph deconvolution operation to evolve representations of bottom nodes guided by the higher-level structure nodes in the spirit of top-down inference, that is, (root node)→(body-part nodes)→(clothes-part nodes)→(leaf landmark nodes). Benefiting from integrating graph clustering and graph deconvolution operations, our LGR layer enables to achieve global structural coherency and effectively enhance each landmark node representation for better predictions.

Moreover, existing fashion landmark datasets [36, 19, 31] annotated with at most 8 landmarks for all clothes appearance. To advance the development of fine-grained domain knowledge in fashion landmark detection research, we contribute a new Fine-grained Fashion Landmark Dataset containing 200k images annotated with at most 32 keypoints for 13 clothes types, named as FFLD. More details of FFLD are presented in supplementary material.

Our contributions are summarized in the following aspects:

1) we propose a general Layout-Graph Reasoning (LGR) layer and incorporate multiply LGR layers into deep networks to seamlessly enforce structural layout relations among clothing landmarks on the intermediate representations to achieve global structure coherency.

2) We define the layout-graph as a hierarchical structure for mining contextual graph semantic information from specific nodes to abstraction nodes. The graph clustering and graph deconvolution operation are integrated into each LGR layer for hierarchical graph reasoning.

3) We construct the first Fine-grained Fashion Landmark Dataset (FFLD) that provides more comprehensive landmark annotations for diverse clothes types.

4) Our model performs the superior ability compared with state-of-the-art approaches over two public fashion landmark datasets (*e.g*. FLD [19] and DeepFashion [36]).

2. Related Work

Fashion Landmark Detection and Localization. Recently many research efforts have been devoted to joint localization and landmark detection [18, 35, 7, 26, 29, 33, 28, 31, 36, 19, 27, 22]. Newell *et al*. [22] proposed a model for human pose estimation using a repeated pooling down and upsampling process to learn the spatial distribution of resolution. Liu *et al*. [19] proposed deep fashion alignment using the pseudo-label scheme to enhance invariability of fashion landmark. Wang *et al*. [27] captured kinematic and symmetry grammar of clothing landmark for mining geometric relationships among landmarks. They

modeled grammar message passing processing as a bidirectional convolutional recurrent neural network for training in an end-to-end manner.

The models of deep learning above demonstrate the powerful representations of neural networks. Few of them consider combining knowledge-guide information with fashion landmark detection in a hierarchical way. Motivated by Rothrock *et al*. [24, 12, 34], we build a hierarchical architecture to model global-local fashion landmark correlations for facilitating contextual information across each landmark.

Knowledge-guide Information in Graph. Recently some research efforts model domain knowledge as graph for mining correlations among labels or objects in images, which has been proved effective in many tasks [12, 3, 23, 34, 13, 20, 24, 1, 4, 6]. Li *et al*. [13] proposed a subgraph-based model for scene graph generation using bottom-up relationship inference of objects in images. Liang *et al*. [15] modeled semantic correlations using semantic neuron graph for explicitly incorporating semantic concept hierarchy during propagation. Yang *et al*. [33] built a prior knowledge-guide graph for body part locations to well consider the global pose configurations.

As far as we know, there is no work considering modeling layout-graph across all scales and levels in the layout among fashion landmarks. We incorporate cross-layer graph relations, low-level and high-level graph relations by layout-graph reasoning layers into a unified model, which consists of Map-to-Node module, layout-graph reasoning module and Node-to-Map module.

Fashion Understanding Datasets. Many human-centric applications depend on reliable fashion image understanding. DeepFashion [17] is a large-scale clothing dataset labeled with clothes categories, attributes, at most 8 clothes landmarks and bounding boxes. FLD [19] is a fashion landmark dataset (FLD) with large pose and scale variations, annotated with at most 8 landmarks and bounding boxes. Yan *et al*. [31] contributed an unconstrained landmark dataset (ULD), which comprises 30k images with at most 8 fashion landmark annotations. To advance the developments of domain knowledge for fine-grained fashion landmark, we propose a large-scale dataset towards the first fine-grained fashion landmark detection task, which contains 200k images annotated with at most 32 key-points for 13 clothes categories.

3. Proposed Approach

3.1. Overview

Considering the layout of fashion landmarks, we model layout-graph reasoning to enforce detected fashion landmarks be coherent with human and clothes layouts from a global perspective. We propose a model that seamlessly enforces layout relationships among landmarks on the intermediate features via multiple stacked Layout-Graph Reasoning (LGR) layers, as shown in Fig.2. Each LGR layer aims to map deep convolutional features into structural graph nodes via Map-to-Node module, perform reasoning over multi-level layout-graph nodes to achieve global layout coherency via a layout-graph reasoning module, and then map evolved graph nodes back to enhanced convolutional features via a Node-to-Map module. Finally a sigmoid function and a 1×1 convolution are employed to produce heatmaps of fashion landmarks. Inspired by Yang *et al*. [32], we enhance intermediate features by pyramid module and decrease data bias by residual addition [9].

3.2. Layout-graph Definition

We define the layout-graph as a hierarchical structure for mining semantic correlations and constraints among different fashion landmarks. Specifically, we define a layout-graph constructed by graph nodes characterizing landmark categories (*e.g*. right-collar, left-sleeve) and graph edges representing spatial layouts (*e.g*. right-collar belongs to collar, collar and sleeve belong to upper body), which is denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The graph nodes $\mathcal{V}$ consist of the set of leaf nodes $\mathcal{V}_{leaf}$ (*e.g*. leaf landmark nodes) and intermediate nodes $\mathcal{V}_{middle}$ (*e.g*. clothes-part nodes, body-part nodes, root node). We define the leaf node representations as $\mathbf{X}_{leaf} \in \mathbb{R}^{N_{leaf} \times d}$, which is generated via Map-to-Node module. We define the intermediate node representations as $\mathbf{X}_{middle} \in \mathbb{R}^{N_{middle} \times d}$, which is generated via layout-graph reasoning module. The N_{leaf} and N_{middle} are the number of leaf nodes and intermediate nodes, d means the feature dimension of each node.

The edges $\mathcal{E}$ consist of the set of leaf edges $\mathcal{E}_{leaf}$ to represent the connections between each leaf nodes (*e.g*. right-collar and left-collar), and intermediate edges $\mathcal{E}_{middle}$ to represent the connections between each intermediate node (*e.g*. collar and sleeve). The leaf node adjacency weight matrix $\mathbf{A}_{leaf} \in \{0, 1\}^{N_{leaf} \times N_{leaf}}$ is initialized according to the edge connections in $\mathcal{E}_{leaf}$ as shown in Fig.3, where 0 means disconnection and 1 means connection. Similarly, we define the intermediate node adjacency weight matrix as $\mathbf{A}_{middle}$. In implementation, we perform normalization on all $\mathbf{A}$ by following [12] to obtained normalized adjacency weight matrix. The $\mathbf{A}$ is crucial for layout-graph information embedded in the fashion joint layouts to benefit point-wise fashion landmark detection, which can be easily designed according to human commonsense about fashion layouts as illustrated in Sec.2 of supplementary material.

3.3. Layout-graph Reasoning Layer

The LGR layer aims to enhance convolutional features by layout-graph reasoning. Each LGR layer consists of three modules: 1) Map-to-Node module to map convolutional features into structural graph leaf nodes; 2) layout-

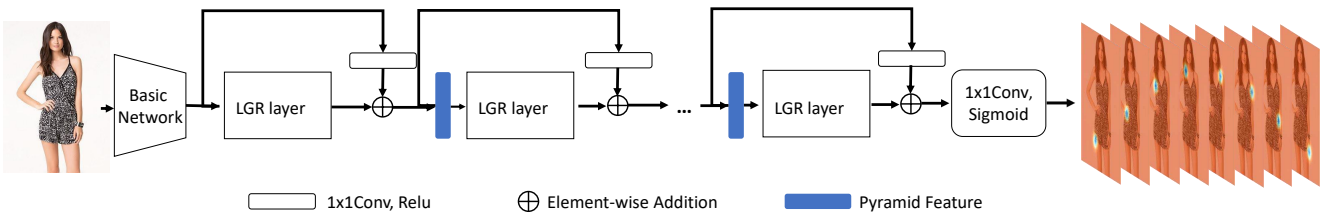

Figure 2: Illustration of our model that incorporates basic convolutional network for features extraction and stacked Layout-Graph Reasoning layers for structural graph reasoning. Residual addition processing and pyramid feature post-processing are appended between each stacked architecture for reducing bias and capturing rich representations across multiple scales. A $1\times$ 1 convolution with $sigmoid$ activation function is utilized to produce final fashion landmark heatmaps. For better viewing of all figures in this paper, please see the original zoomed-in color pdf file.

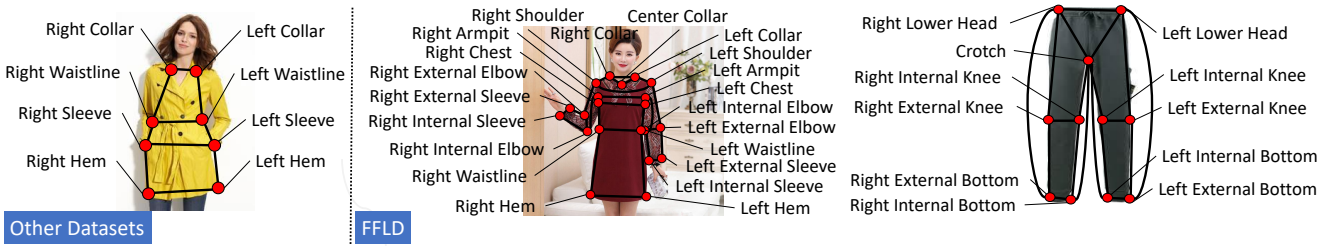

Figure 3: Layout-graph definitions for different fashion landmark datasets. Each leaf node (red circles) represents the position and type of one clothing landmark. Each leaf edge (black lines) indicates the correlations between landmark points.

graph reasoning module to model global-local clothing landmark correlations for feature enhancement, containing a graph clustering operation and a graph deconvolution operation; 3) Node-to-Map module to map evolved leaf node representations back to enhance feature representations.

3.3.1 Map-to-Node Module

This module is to seamlessly map convolutional feature maps to graph node representations. Given the input convolutional feature maps after dimension transformation ($\mathbf{F} \in \mathbb{R}^{H\times W\times C} \rightarrow \mathbf{F} \in \mathbb{R}^{HW\times C}$, where H, W and C represent the height, weight and channel), this module produces graph leaf node representations $\mathbf{X}_{leaf} \in \mathbb{R}^{N_{leaf}\times d}$. The formulation is:

$$\mathbf{X}_{leaf} = \sigma(\Phi(\mathbf{F}\mathbf{W}_m)^T\mathbf{F}\mathbf{W}_t), \tag{1}$$

where $\mathbf{W}_m \in \mathbb{R}^{C\times N_{leaf}}$ and $\mathbf{W}_t \in \mathbb{R}^{C\times d}$ are trainable sampling matrices. The Φ denotes normalized function *softmax* to sum all rows to one, and the σ denotes non-linear function *Relu*.

3.3.2 Layout-Graph Reasoning Module

Given graph leaf node representations $\mathbf{X}_{leaf}$ via the Map-to-Node module, our layout-graph reasoning module first performs a graph clustering operation to generate representations of intermediate nodes in the spirit of bottom-up inference, that is, propagating from (leaf landmark nodes)→(clothes-part nodes)→(body-part nodes)→(root node). Then a graph deconvolution operation to evolve representations of bottom nodes guided by the higher-level structure nodes in the spirit of top-down inference, that is, (root node)→(body-part nodes)→(clothes-part nodes)→(leaf landmark nodes). Benefiting from integrating the graph clustering and graph deconvolution operations, the module achieves global structural coherency.

Graph Clustering Operation. This operation generates intermediate node representations by graph clustering. For different levels of bottom-up inference, the clustered graph nodes and graph edges change as shown in Fig.4. The graph clustering operation of every levels is similar. Here we take $\mathbf{X}_{leaf} \rightarrow \mathbf{X}_{middle}$ as an example to illustrate the clustering operation. Given the input $\mathbf{X}_{leaf}$ and $\mathbf{A}_{leaf}$, this operation generates intermediate node representations $\mathbf{X}_{middle}$ and intermediate node adjacency weight matrix $\mathbf{A}_{middle}$, which is formulated as:

$$\mathbf{X}_{middle} = \sigma\left(\mathbf{W}_{p'}^T\mathbf{A}_{leaf}\mathbf{X}_{leaf}\mathbf{W}_h\right), \tag{2}$$

$$\mathbf{A}_{middle} = \sigma\left(\mathbf{W}_p^T\mathbf{A}_{leaf}\mathbf{W}_p\right), \tag{3}$$

where $\mathbf{W}_{p'} \in \mathbb{R}^{N_{leaf}\times N_{middle}}$ and $\mathbf{W}_p \in \mathbb{R}^{N_{leaf}\times N_{middle}}$ are both trainable clustering matrices. The $\mathbf{W}_h \in \mathbb{R}^{d\times d}$ is a trainable weight matrix. We use graph convolution from [12] for graph reasoning to perform over $\mathbf{X}_{leaf}$ and $\mathbf{A}_{leaf}$ using $\mathbf{W}_h$ to update the leaf graph node representations. Then we utilize $\mathbf{W}_{p'}$ to cluster updated $\mathbf{X}_{leaf}$ into $\mathbf{X}_{middle}$, which is formulated as Eq.2. We cluster $\mathbf{A}_{leaf}$ using $\mathbf{W}_p$ to generate $\mathbf{A}_{middle}$, which is formulated as Eq.3. $\mathbf{W}_p$ is a permutation matrix and obeys distribution of $\mathbf{W}_p^T\mathbf{W}_p =$

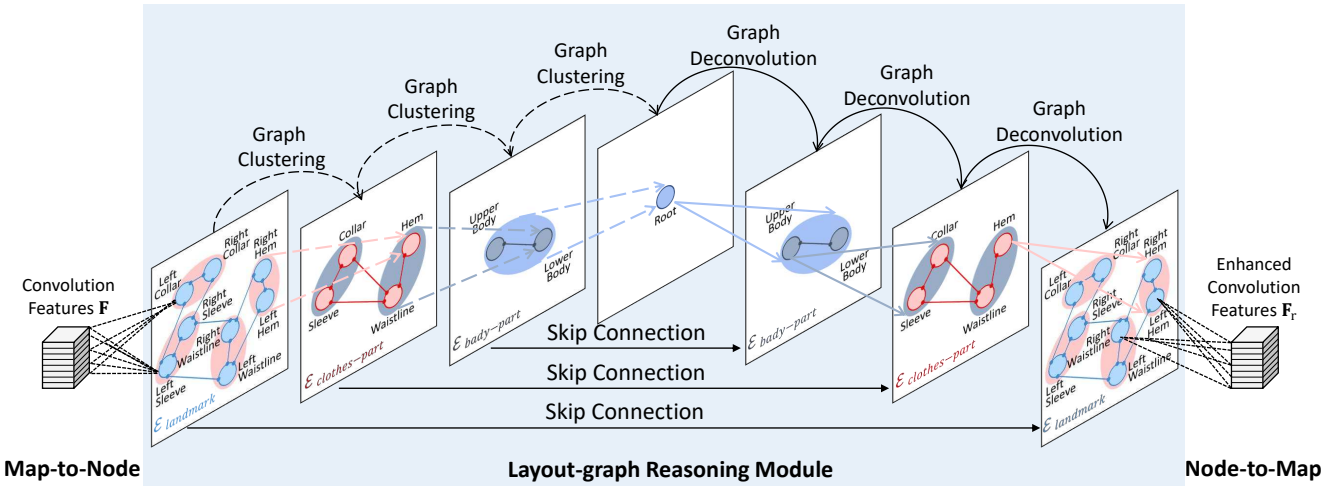

Figure 4: Illustration of our Layout-graph Reasoning (LGR) layer which contains Map-to-Node module, layout-graph reasoning module and Node-to-Map module. In Map-to-Node and Node-to-Map module, weighted sample operations vote all convolution features (evolved leaf landmark nodes) to leaf landmark nodes (enhanced convolution features) by weighted sample. In Layout-Graph Reasoning module, the graph is propagated from leaf landmark nodes to root node by Graph Clustering and Graph Reasoning. The root node is propagated back again by Graph Deconvolution and Graph Reasoning for producing evolved leaf landmark nodes. We use graph convolution from [12] for Graph Reasoning with supervising adjacency matrix. A skip connection is employed for restricting the consistency of clustering and deconvolution operations.

$\mathbf{I}$. We use graph reasoning to perform over $\mathbf{X}_{middle}$ and $\mathbf{A}_{middle}$ to update the clustered graph node representations $\mathbf{X}_{middle} \in \mathbb{R}^{N_{middle} \times d}$.

Graph Deconvolution Operation. This operation evolves representations of bottom nodes guided by the higher-level structure nodes in the spirit of top-down inference as shown in Fig.4. Again, we take $\mathbf{X}_{middle} \rightarrow \mathbf{X}_{leaf}$ an as example to illustrate the deconvolution operation. Given the input intermediate node representations $\mathbf{X}_{middle}$ and adjacency matrix $\mathbf{A}_{middle}$ from higher-level structure, we utilize the formulation like Eq.2 and Eq.3 to produce leaf node representations $\mathbf{X}_{leaf}$ and leaf node adjacency weight matrix $\mathbf{A}_{leaf}$. Furthermore, to integrate the high-level and low-level structure information, we utilize a matrix addition over the node representations before clustering and after deconvolution, followed by the graph reasoning to update the evolved leaf node representations $\mathbf{X}_{leaf}$.

3.3.3 Node-to-Map Module

We map evolved graph nodes into enhanced convolutional features via Node-to-Map module. Given the input convolutional features $\mathbf{F}$ and evolved leaf node representations $\mathbf{X}_{leaf}$, this module aims to generate enhanced convolutional feature representations $\mathbf{F}_r$. We first perform the dimension transformation for $\mathbf{F} \in \mathbb{R}^{HW \times C} \rightarrow \mathbf{F} \in \mathbb{R}^{HW \times N \times C}$ and $\mathbf{X}_{leaf} \in \mathbb{R}^{N_{leaf} \times d} \rightarrow \mathbf{X}_{leaf} \in \mathbb{R}^{HW \times N_{leaf} \times d}$. Then we concatenate $\mathbf{F}$ and $\mathbf{X}_{leaf}$ to $\mathbf{X}_a \in \mathbb{R}^{HW \times N_{leaf} \times (C+d)}$ for richer feature representations. We formulate this module as:

$$\mathbf{F}_r = \sigma(\Phi(\mathbf{X}_a \mathbf{W}_{m'}) \sigma(\mathbf{X}_{leaf} \mathbf{W}_{t'})), \tag{4}$$

Eq 4 is to map node representations $\mathbf{X}_{leaf} \in \mathbb{R}^{N_{leaf} \times d}$ to enhanced convolutional features $\mathbf{F}_r \in \mathbb{R}^{HW \times C}$, where $\mathbf{W}_{m'} \in \mathbb{R}^{C+d}$ is a vector with $C+d$ dimension and $\mathbf{W}_{t'} \in \mathbb{R}^{d \times C}$ is a trainable sampling matrices.

4. Experiments

4.1. Experimental Settings

Network Architecture. Following the baseline of [36, 19, 31, 27], we use VGG16 [25] with four stacked LGR layers for feature extraction and layout-graph reasoning. Each LGR layer contains Map-to-Node module, layout-graph reasoning module and Node-to-Map module. We map convolutional features into graph leaf node representations via Map-to-Node module. On DeepFashion and FLD, we set 8 leaf nodes (*e.g.* left-collar, right-hem), 6 intermediate nodes including 4 clothes-part nodes (collar, hem) and 2 body-part nodes (*e.g.* upper body, lower body), and 1 root node. On FFLD, we set 32 leaf nodes (*e.g.* left-shoulder, crotch), 14 intermediate nodes including 12 clothes-part nodes (*e.g.* sleeve, knee) and 2 body-part nodes (*e.g.* upper body, lower body), and 1 root node. More details of node's layout can be seen in supplementary material. Then we model layout-graph of fashion landmarks via layout-graph reasoning module to evolve graph node representations guiding

by defined graph correlations as shown in Fig.3. The Node-to-Map module to map evolved graph node representations into convolutional features for enhancing the feature representations, which results are fed into a $1\times$ 1 convolution with $sigmoid$ activation to get the prediction. A residual addition and pyramid feature post-processing are appended between each LGR layer for reducing bias and capturing rich representations across multi-scales.

Three Benchmarks and Evaluation. We evaluate and report the results and comparisons on three datasets. DeepFashion [36] is the largest fashion dataset so far, which contains 289,222 images annotated with bounding boxes and at most 8 landmarks. FLD [19] is a fashion landmark dataset with more diverse variations (*e.g*. pose, scale, background), which contains 123, 016 images annotated at most 8 landmarks and bounding boxes per image. FFLD is our contributed fine-grained fashion landmark dataset, which contains 200k images annotated with at most 32 key-points and bounding boxes for 13 clothes categories. Following [27], 209,222 fashion images are used for training; 40, 000 images are used for validation and remaining 40, 000 images are for testing in DeepFashion. Following the protocol in FLD [19], 83, 033 images and 19, 992 fashion images are used for training and validating, 19, 991 images are used for testing. In FFLD, we use 120K images as a training set, 40K images as a validation set and 40K images as a test set. Normalized error(NE) metric [19] is adopted for evaluation. We utilize l_2 function to calculate the distance between predicted heatmaps and ground-truth in normalized coordinate space (*i.e*. normalized by the height and width of image).

Training Strategy and Object Function. We use LGR layer for fashion landmark detection over FLD [19], DeepFashion [36] and FFLD separately without any pre-trained models. Following [27], we first crop each image using labeled bounding boxes, resize the cropped image to 224 $\times$ 224, and extract features for graph reasoning. The training data are augmented by scaling, rotation, and flipping. We train all the models using stochastic gradient descent with a batch size of 16 images, which is optimized by Adam optimizer [11] with an initial learning rate of 1.e-3 on an 11 GB NVIDIA 1080Ti GPU. Betas of Adam are from 0.9 to 0.999. On FLD, we linearly drop the learning rate by a factor of 10 every 20 epochs. On DeepFashion and FFLD, we linearly decrease the learning rate by a factor of 10 every 10 epochs. We stop training when no improvements on the validation set. We set the mean squared error (MSE) equation as an objective function between the final predicted heatmaps and ground-truth.

4.2. Comparison with the state-of-the-arts

LGR achieves an obvious improvement over the two large datasets compared with PyraNet [32], FashionNet [36], DFA [19], DLAN [31] and BCRNNs [27]. Note that PyraNet is human pose estimation model with two stages. We train the PyraNet following the same strategy as [32]. Our LGR outperforms SOTA at 0.0419 on FLD and 0.0336 on DeepFashion, which is much lower than the closest competitor (0.0583 and 0.0484), as shown in Table.1. Compared with traditional DCNNs [36] and grammar model [27], we further model layout-graph reasoning to enforce detected fashion landmarks be coherent with human and clothes layouts from a global perspective. Benefiting from the joint reasoning with hierarchical structures of fashion landmarks, we achieve the SOTA performs over all existing models by a large improvement. Note that our model consistently decreases the NE in all landmarks on DeepFashion.

4.3. Ablation Study

Different Stack Numbers. There are six experiments to display the performance of different stacked LGR layers, which are shown in the first list of Table.2. VGG16 without any graph reasoning achieves 0.0871 average NE, which is the worst result compared with other knowledge-guide model. Comparing the variants of different stages, the performances get better with the stack increasing, which is a coarse-to-fine processing to repeatedly refining the prediction. Five stacked LGR gets the best performance(0.0405 NE), while it needs more GPU memory and time during training/validating/testing progress. The gap of all landmarks between five stacks and four stacks are closed. Limited by device and time, we select four stacks in our stand model and apply it to all extensive experiments.

Different Graph Layers. In the second list of Table.2, we built an ablation study of different normal graph layers on FLD and DeepFashion. To demonstrate the superior ability of graph clustering and graph deconvolution, we replace a LGR layer with one graph layer without graph clustering and deconvolution, which is presented as one-layer. Benefiting from graph reasoning, two graph layers can get the best performance, while the performance tends to be destroyed with the depth increasing of graph layers (*e.g*. two-layer:0.0471, eight-layer:0.0954). Purely increasing the graph layers can not simply get better performance, but the spent time also grows up with the model size increasing. LGR can attenuate the shortcoming as above and get better performance with layer increasing by graph clustering and graph deconvolution. Compared the LGR layer with normal graph layer, eight graph layers got worse performance compared with three-clustering (0.0954 and 0.0419). Note that three-clustering got close speed compared with eight graph layers (*e.g*. 0.00379s and 0.00357s). Due to the LGR layer contains graph clustering, graph deconvolution and graph reasoning operations, which has more processes compared with the same size of normal graph layers.

Different Number of Graph Clustering and Deconvolution. In the third list of Table.2, we explore the effec-

Table 1: Comparison with the state-of-the-art model on the FLD test set and DeepFashion test set using the NE metric.

FLD									
Methods	L.Collar	R.Collar	L.Sleeve	R.Sleeve	L.Waistline	R.Waistline	L.Hem	R.Hem	Avg.
FashionNet [36]	.0784	.0803	.0975	.0923	.0874	.0821	.0802	.0893	.0859
PyraNet [32]	.0341	.0341	.0610	.0620	.0920	.0921	.0314	.0291	.0723
DFA [19]	.048	.048	.091	.089	-	-	.071	.072	.068
DLAN [31]	.0531	.0547	.0705	.0735	.0752	.0748	.0693	.0675	.0672
BCRNNs [27]	.0463	.0471	.0627	**.0614**	.0635	.0692	.0635	.0527	.0583
LGR(ours)	**.0423**	**.0152**	**.0502**	.0735	**.0195**	**.0512**	**.0452**	**.0393**	**.0419**
DeepFashion									
Methods	L.Collar	R.Collar	L.Sleeve	R.Sleeve	L.Waistline	R.Waistline	L.Hem	R.Hem	Avg.
FashionNet [36]	.0854	.0902	.0973	.0935	.0854	.0845	.0812	.0823	.0872
PyraNet [32]	.0343	.0343	.0602	.0613	.0920	.0931	.0308	.0291	.0719
DFA [19]	.0628	.0637	.0658	.0621	.0726	.0702	.0658	.0663	.0660
DLAN [31]	.0570	.0611	.0672	.0647	.0703	.0694	.0624	.0627	.0643
BCRNNs [27]	.0415	.0404	.0496	.0449	.0502	.0523	.0537	.0551	.0484
LGR(ours)	**.0270**	**.0116**	**.0286**	**.0347**	**.0307**	**.0435**	**.0160**	**.0162**	**.0336**

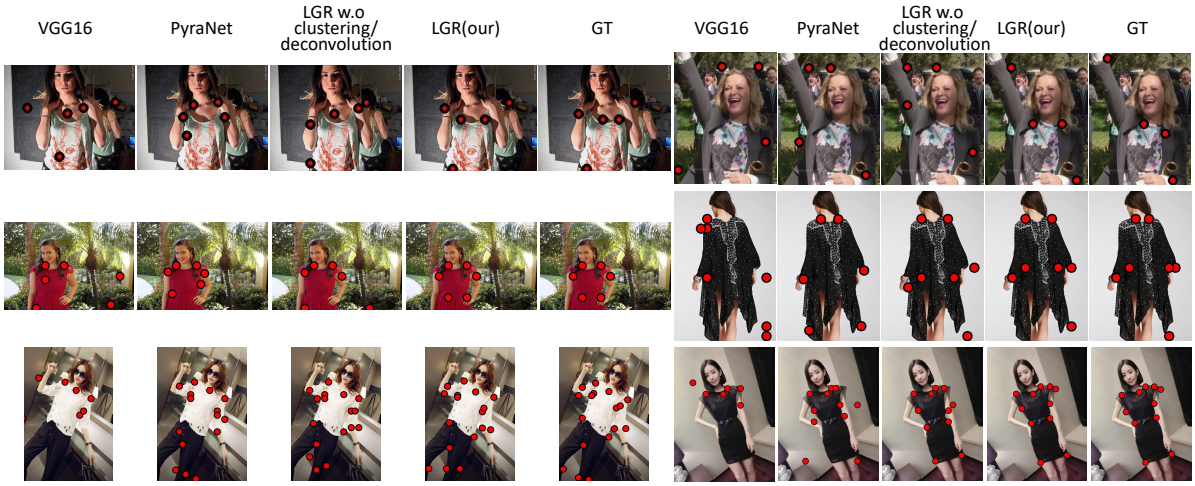

Figure 5: Qualitative results for VGG16 [25], PyraNet [32], LGR w.o clustering/deconvolution (two graph layers without graph clustering and deconvolution) and LGR over DeepFashion (first row), FLD (second row) and FFLD (bottom row). The detected landmarks (red circle) are performed on different variations such as occlusion and complicate background. Please see the zoomed-in color pdf file.

tiveness of different numbers of graph clustering operation and graph deconvolution operation in LGR layer, the diagram as shown in Fig.6. With the depth of operations increasing, more prior commonsense knowledge in terms of richer human body part layouts and clothes part layouts will be got to better guide the learning processing. The experiment results have shown that three-clustering get better performance than one clustering (0.0419 and 0.0488 on FLD, 0.0336 and 0.0403 on DeepFashion).

Different Injected Layers. The fourth list of Table.2 compares the variants of injecting four stacked LGR layers into different convolution blocks (ConvBlock) of VGG16 [25] over FLD. The stacked LGR layers are injected into right before the block. The performance of adding LGR layers after Block3 is worse than adding LGR layers after ConvBlock5 of VGG16. We show the possible reason that the deeper layers can encode more semantically high-level feature representation, which is more suitable for layout-

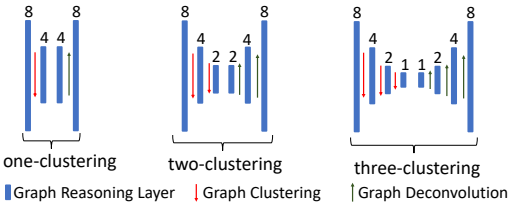

Figure 6: The different structures of graph clustering and graph deconvolution. The number of nodes in each graph reasoning layer is labeled on the top. Please see the zoomed-in color pdf file.

graph reasoning.

4.4. Qualitative Results

The results showed different abilities of traditional DCNNs [25], PyraNet [32], normal graph reasoning without graph clustering and deconvolution and LGR. We select the best structure (two-layer) of normal graph layers demon-

Table 2: Ablation study on FLD and DeepFashion using NE metric (Avg.). The structures with different numbers of graph clustering and deconvolution are shown in Fig.6. We also present the results generate by different numbers of normal graph convolutional layers that replacing the graph clustering and deconvolution. We also compare the average execution time for testing (Time).

Different Stack numbers				
	FLD			
Methods	Avg.	ΔAvg.	Time(s)	ΔTime(s)
VGG16 [25]	.0871	.0452	**.00065**	.00314
one stack	.0711	.0292	.00155	.00224
two stacks	.0535	.0116	.00236	.00143
three stacks	.0529	.0110	.00346	.00033
four stacks	.0419	-	.00379	-
five stacks	**.0405**	.0014	.00472	.00093

Different Graph Layers				
	FLD		DeepFashion	
Methods	Avg.	Time(s)	Avg.	Time(s)
one-layer	.0531	**.00241**	.0482	**.00237**
two-layer	**.0471**	.00273	**.0437**	.00266
four-layer	.0639	.00279	.0562	.00271
six-layer	.0644	.00289	.0638	.00300
eight-layer	.0954	.00357	.0779	.00341

Different Number of Graph Clustering and Deconvolution				
	FLD		DeepFashion	
Methods	Avg.	Time(s)	Avg.	Time(s)
one-clustering	.0488	**.00267**	.0403	**.00261**
two-clustering	.0443	.00302	.0372	.00336
three-clustering	**.0419**	.00379	**.0336**	.00352

Different Injected Layers	
	FLD
Methods	Avg.
VGG16 ConvBlock1	.0811
VGG16 ConvBlock3	.0574
VGG16 ConvBlock5	**.0419**

strated as above. In Fig.5, for complex background, diverse clothes layouts and styles, multiple scales and views, the pure DCNNs, pose estimation model [32] and normal graph layers fail to detect correct fashion landmark. Benefiting from modeling layout-graph relations of landmarks by a hierarchical structure, LGR can mine semantic coherency of layout-graph and enhance the semantic correlations and constrains among landmarks (*e.g*. collar and sleeve belong to upper body). LGR covers difficult variance and generate reasonable results guided by layout-graph reasoning during the bottom-up, top-down inference. For example, LGR can detect correct results by constraining fashion landmarks on one clothing in complex background (first row in Fig.5).

4.5. Fine-grained Fashion Landmark Dataset (FFLD)

Compared with exited fashion landmark datasets [36, 19, 31], FFLD consists of more than 70% consumer images, which is more challenge for multiple view and light, complex background and deformable clothes appearance. Note that the FFLD is the closest fashion landmark dataset with the real application. More detailed definition and statistics of FFLD can be seen in supplementary material.

Table 3: Evaluation of different models on FFLD.

Methods	Avg. NE
FashionNet [36]	.2031
PyraNet [32]	.1423
GCN [12]	.1272
BCRNN [27]	.1226
LGR (ours)	**.1180**

We have shown four existing methods evaluated on FFLD in Table 3 to comprehensively perform FFLD. The FashionNet [36] and BCRNN [27] are the SOTA methods for fashion landmark detection, and the PyraNet [32] is one of SOTA methods for human pose estimation. We utilize VGG16 [25] stacked with two graph convolutional layers, which is regard (GCN) [12] is a typical graph-based methods, which VGG16 in this evaluation. Based on the prior layout-graph as shown in Fig.3, layout-graph reasoning with four stacks is evaluated on FFLD.

As shown in Table.3, the LGR achieved 0.118 average NE on FFLD, which is a worse performance compared with FLD(0.0419 NE) and DeepFashion(0.0336 NE) due to more consumer images, more fine-grained fashion landmarks, more challenge views and backgrounds. To demonstrate the challenge of FFLD on other models, we perform fashion landmark detection model (FashionNet [36] and BCRNN [27]), human pose estimation model (PyraNet [32]) and normal graph layer (GCN [12]) on FFLD to achieve 0.2031 NE, 0.1226 NE, 0.1423 NE and 0.1272 NE. We perform BCRNN on FFLD following the setting of [27], and the fashion landmark grammars of FFLD consist of kinematics grammar and symmetry grammar. More detailed fashion grammars of FFLD can be seen in supplementary material.

5. Conclusion

In this paper, we proposed the Layout-Graph Reasoning (LGR) that consists of three modules for fashion landmark detection to seamlessly utilize structural graph reasoning in a hierarchical way. We use LGR to achieve SOTA performance over recent methods. We contribute a fine-grained fashion landmark dataset to advance the development of knowledge graph to fashion landmark research.

6. Acknowledgements

This work was supported in part by the Sun Yat-sen University Start-up Foundation Under Grant No. 76160-18841201, in part by the National Key Research and Development Program of China under Grant No. 2018YFC0830103, in part by National High Level Talents Special Support Plan (Ten Thousand Talents Program), and in part by National Natural Science Foundation of China (NSFC) under Grant No. 61622214, and 61836012.

References

[1] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[2] X. Chen and A. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *NIPS*, 2014.

[3] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3844–3852. Curran Associates, Inc., 2016.

[4] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam. Large-scale object classification using label relation graphs. In *ECCV*, pages 48–64. Springer, 2014.

[5] W. Di, C. Wah, A. Bhardwaj, R. Piramuthu, and N. Sundaresan. Style finder: Fine-grained clothing style detection and retrieval. In *CVPR Workshops*, pages 8–13, 2013.

[6] C. Gan, M. Lin, Y. Yang, G. de Melo, and A. G. Hauptmann. Concepts not alone: Exploring pairwise relationships for zero-shot video activity recognition. In *AAAI*, 2016.

[7] K. Gong, X. Liang, D. Zhang, X. Shen, and L. Lin. Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing. In *CVPR*, 2017.

[8] X. Han, Z. Wu, P. X. Huang, X. Zhang, M. Zhu, Y. Li, Y. Zhao, and L. S. Davis. Automatic spatially-aware fashion concept discovery. In *ICCV*, 2017.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[10] J. Huang, R. Feris, Q. Chen, and S. Yan. Cross-domain image retrieval with a dual attribute-aware ranking network. In *ICCV*. IEEE, 2015.

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

[12] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[13] Y. Li, W. Ouyang, B. Zhou, Y. Cui, J. Shi, and X. Wang. Factorizable net: An efficient subgraph-based framework for scene graph generation. *arXiv preprint arXiv:1806.11538*, 2018.

[14] X. Liang, K. Gong, X. Shen, and L. Lin. Look into person: Joint body parsing & pose estimation network and a new benchmark. *T-PAMI*, 2018.

[15] X. Liang, H. Zhou, and E. Xing. Dynamic-structured semantic propagation network. In *CVPR*, pages 752–761, 2018.

[16] S. Liu, Z. Song, G. Liu, C. Xu, H. Lu, and S. Yan. Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In *CVPR*, pages 3330–3337, 2012.

[17] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *CVPR*, 2016.

[18] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

[19] Z. Liu, S. Yan, P. Luo, X. Wang, and X. Tang. Fashion landmark detection in the wild. In *ECCV*. Springer, 2016.

[20] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei. Visual relationship detection with language priors. In *ECCV*, pages 852–869. Springer, 2016.

[21] S. L. A. C. B. T. L. B. M. Hadi Kiapour, Xufeng Han. Where to buy it:matching street clothing photos in online shops. In *ICCV*, 2015.

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

[23] M. Niepert, M. Ahmed, and K. Kutzkov. Learning convolutional neural networks for graphs. In *ICML*, pages 2014–2023, 2016.

[24] B. Rothrock and S.-C. Zhu. Human parsing using stochastic and-or grammars and rich appearances. In *ICCV*, pages 640–647. IEEE, 2011.

[25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[26] G. Trigeorgis, P. Snape, M. A. Nicolaou, E. Antonakos, and S. Zafeiriou. Mnemonic descent method: A recurrent process applied for end-to-end face alignment. In *CVPR*, pages 4177–4187, 2016.

[27] W. Wang, Y. Xu, J. Shen, and S.-C. Zhu. Attentive fashion grammar network for fashion landmark detection and clothing category classification. In *CVPR*, pages 4271–4280, 2018.

[28] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *CVPR*, June 2016.

[29] W. Wu, C. Qian, S. Yang, Q. Wang, Y. Cai, and Q. Zhou. Look at boundary: A boundary-aware face alignment algorithm. In *CVPR*, pages 2129–2138, 2018.

[30] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg. Retrieving similar styles to parse clothing. *T-PAMI*, 37(5):1028–1040, 2015.

[31] S. Yan, Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang. Unconstrained fashion landmark detection via hierarchical recurrent transformer networks. In *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, 2017.

[32] W. Yang, S. Li, W. Ouyang, H. Li, and X. Wang. Learning feature pyramids for human pose estimation. In *ICCV*, volume 2, 2017.

[33] W. Yang, W. Ouyang, H. Li, and X. Wang. End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In *CVPR*, 2016.

[34] R. Ying, J. You, C. Morris, X. Ren, W. L. Hamilton, and J. Leskovec. Hierarchical graph representation learning withdifferentiable pooling. *arXiv preprint arXiv:1806.08804*, 2018.

[35] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, pages 2879–2886. IEEE, 2012.

[36] S. Q. X. W. Ziwei Liu, Ping Luo and X. Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *CVPR*, June 2016.