# Pattern-Structure Diffusion for Multi-Task Learning

Ling Zhou, Zhen Cui*, Chunyan Xu, Zhenyu Zhang, Chaoqun Wang, Tong Zhang, Jian Yang
PCA Lab, Key Lab of Intelligent Perception and Systems for High-Dimensional
Information of Ministry of Education, School of Computer Science and Engineering,
Nanjing University of Science and Technology

## Abstract

*Inspired by the observation that pattern structures high-frequently recur within intra-task also across tasks, we propose a pattern-structure diffusion (PSD) framework to mine and propagate task-specific and task-across pattern structures in the task-level space for joint depth estimation, segmentation and surface normal prediction. To represent local pattern structures, we model them as small-scale graphlets*[1]*, and propagate them in two different ways, i.e., intra-task and inter-task PSD. For the former, to overcome the limit of the locality of pattern structures, we use the high-order recursive aggregation on neighbors to multiplicatively increase the spread scope, so that long-distance patterns are propagated in the intra-task space. In the inter-task PSD, we mutually transfer the counterpart structures corresponding to the same spatial position into the task itself based on the matching degree of paired pattern structures therein. Finally, the intra-task and inter-task pattern structures are jointly diffused among the task-level patterns, and encapsulated into an end-to-end PSD network to boost the performance of multi-task learning. Extensive experiments on two widely-used benchmarks demonstrate that our proposed PSD is more effective and also achieves the state-of-the-art or competitive results.*

## 1. Introduction

Dense pixel prediction tasks, e.g., depth estimation, segmentation and surface normal prediction, are fundamental yet challenging in computer vision due to the important applications to intelligent robotic [42], automatic drive [6], etc. Currently, numerous deep learning based methods have obtained great success in each of three tasks. However, the single-task models focus more on the learning of robust regression, but rarely consider the interactions between tasks. As the pixel-level tasks in scene understanding, actually, these three tasks have some common characteristics that can share with each other.

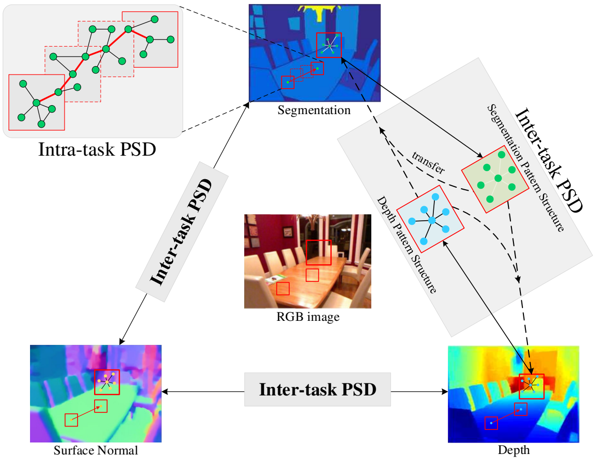


Figure 1: Illustration of our main idea. For multi-task learning, we specifically design intra-task and inter-task pattern-structure diffusion (PSD) to mine and propagate useful pattern structures within/across tasks. Inter-task PSD bridges two tasks to mutually transfer pattern structures for each other through derived correlations, while intra-task PSD propagates within-task patterns through high-order diffusion on graphlets.

Recently joint-task learning methods [29, 55, 8, 38, 53] have sprung up and shown a promising direction to boost the performance through cross-task interactions. Most of them devote to feature fusion (e.g., concatenation) or model sharing (e.g., common network parameters) by following the conventional fusion lines. Due to unintentional integration, these ”black-box” methods cannot concern/know what concrete information is transmitted/interacted between multi-tasks. Ultimately, what information could be available to bridge different tasks has not yet been well revealed and utilized in the study of multi-task pixel prediction.

An observation [58] is that local patch patterns high-frequently recur within the same image, as well as different-

*Corresponding author: zhen.cui@njust.edu.cn

[1] Graphlets are small connected subgraphs of a large graph

scaling images. It implicitly indicates the high-degree similarity of a large number of local pattern structures of natural images. The structures of local patterns provide some strong cues for pixel-level predictions, where the matching pattern structures could result in similar prediction values with high probabilities. More importantly, this observation can be extended into the scenes across different tasks as shown in Fig. 1, where a large number of patches have extremely similar pattern structures at the same spatial positions. For example, the patches (red squares) at the same positions from different tasks have similar pattern structures across depth, segmentation and surface normal. They describe the same objects, and endow the similar information about object shapes/boundaries. Therefore, those local pattern-structures hidden in the image also inter-task should be mined and utilized for pixel-level multi-task learning.

Motivated by the observation on the recurrency of pattern structures within intra-task also across tasks, we propose a pattern-structure diffusion (PSD) framework to mine and propagate task-specific and task-across pattern structures in the task-level space for joint multi-task learning across depth estimation, segmentation and surface normal prediction. To characterize local pattern structures, we construct them as small-scale graphlets, whose topologies represent the pixel-level structure layouts while each vertex is anchored at one pixel position. It means that the graph w.r.t a local region encodes the correlations of pixel-level patterns therein. To transmit pattern structures in the task domain, we construct two types of pattern-structure diffusion process, named intra-task and inter-task PSD. For the former, to overcome the limit of locality of pattern-structure, we propose the high-order recursive diffusion to multiplicatively increase the propagation scope through calculating on the adjacent matrix. Such a recursive pattern-structure diffusion can reduce computation burden as well as memory requirement, in contrast to direct larger-scope or global pattern correlations. In the inter-task PSD process, we derive the similarities of those paired pattern structures corresponding to the same spatial positions, and then mutually transfer the counterpart structures into the task itself based on the learnt similarities. As the long-distance diffusion is done in intra-task, actually, inter-task PSD could implicitly borrow those large-scope pattern structures of the counterpart tasks besides the task itself. Finally, the intra-task and inter-task pattern structures are jointly diffused among the task-level patterns, and encapsulated into an end-to-end PSD network to boost the performance of multi-task learning. We conduct extensive experiments of joint depth, segmentation and surface normal estimation on two public datasets, NYUD-v2 [35], SUNRGB-D [40]. The experiments demonstrate that our proposed PSD method is more effective than those baselines and also achieves the state-of-the-art or competitive results.

In summary, our contributions are in three aspects: i) We propose a novel pattern-structure diffusion framework to attempt to mine and propagate local pattern structures in/across different task domains, ii) We propose two types of pattern-structure diffusions, i.e., intra-task and inter-task, where the former introduces recursive mechanism to learn long-distance propagation, while the latter derives inter-task correlations to transfer cross-task structures, iii) We validate the effectiveness of our proposed PSD method and achieve the state-of-the-art or competitive performance for depth, segmentation and surface normal estimation on two public multi-task learning datasets.

## 2. Related Work

**Semantic Segmentation:** With the great success of deep learning in high-level vision tasks, numerous semantic segmentation approaches [31, 33, 4, 9, 37] are beneficial from CNNs. Long *et al.* [24] proposed a full convolutional neural network (FCN) for semantic segmentation which conduct the pixel-wise classification in an end-to-end fashion. Later, many methods [7, 20, 31] are based on FCN. Due to the large scale RGB-D datasets are published, some RGB-D methods [36, 47, 13, 14] have sprung up. Besides, some methods [44, 15] used the graph-based representation for the problem of segmenting an image into regions. Different from these methods, we only use RGB images as input source and conduct semantic segmentation prediction based on depth prediction rather than depth ground truth. Also, we draw support from other tasks for improving the segmentation prediction.

**Depth Estimation.** Monocular depth estimation has been studied for a long history, previous works on it generally utilized Markov Random Field (MRF) [3, 2]. Recently, several works [45, 19, 34, 50, 38, 27, 18, 52, 57, 56] with CNN architectures have achieved state-of-the-art results. Eigen *et al.* [11] first used CNN and proposed a multi-stage network to solve the monocular depth estimation. Roy *et al.* [39] utilized the regression forest and constructed shallow architectures at each tree nodes to predict depth. Unlike these depth-only prediction methods, we propose to make use of cues from other tasks to boost depth estimation.

**Surface Normal Estimation.** Own to the strong feature representation ability of deep neural networks, most methods [30, 16, 17, 26, 48] for surface normal estimation are based on deep neural networks. Eigen and Fergus [14] adopt a unified coarse-to-fine hierarchical network for depth/normal prediction. Wang *et al.* [46]are the first to regularize dense geometry estimation with planar surface information via only a single RGB image. Recently, Qi *et al.* [38] proposed to use 3D geometric information for predicting surface normal and depth. In our work, our prediction of surface normal is boosted by depth and segmentation

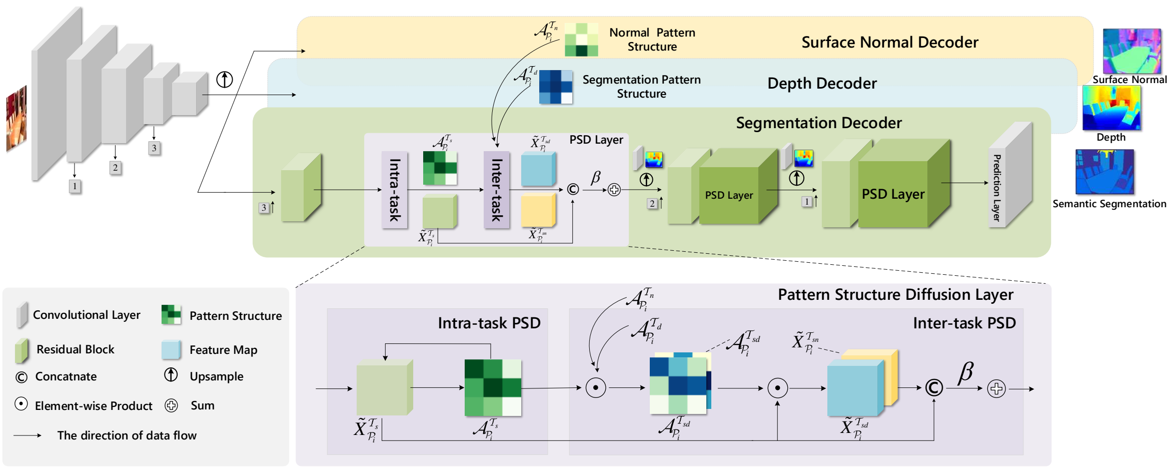


Figure 2: The PSD network architecture for joint prediction of depth, semantic segmentation and surface normal. The whole network is divided into a shared encoder and three task-specific decoder branches. In each branch, we first perform intra-task PSD (Section 3.3) to transmit long-distance pattern-structure information within each task. Then cross-task pattern structures are correlated to transfer to each other, called inter-task PSD (Section 3.4). Finally, intra-task and inter-task PSD are encapsulated into a pattern structure diffusion layer (named PSD layer), which can be stacked in the deep mode for pixel-level multi-task learning.

information.

**Multi-task Learning.** Many multi-task learning methods [1, 23, 43, 22, 53, 41, 54] have achieved great success. Several researchers [22, 53] proposed multi-task learning mechanisms for feature transmitting. Recently, Zhang *et al.* [55] proposed to learn non-local task-specific pattern affinities and obtain the cross-task affinities with fixed parameters for interactiveness. Our method is different from these approaches in the following aspects: i) transmits pattern structures across tasks rather than simple weighting features, ii) mines local patch pattern structures (i.e., graphlets) and multiplicatively diffuses them from local to global region, which has an incidental advantage of high-efficient computation compared to global affinities [55], iii) models with graph topologies versatile to different tasks.

## 3. Pattern-Structure Diffusion

In this section, we first overview the whole network architecture, and then introduce the definition of local pattern structure, intra-task and inter-task pattern-structure diffusion respectively, finally pose the objective function consisting of three different pixel-level prediction tasks.

### 3.1. Network Architecture

The pattern-structure diffusion is encapsulated into an end-to-end deep network as shown in Fig. 2. The whole network can be divided into a shared encoder and three task-specific decoders in which pattern structures are mutually propagated within intra-task also across tasks. Given one RGB image $x$, the encoder produces multi-scale hierarchical feature maps through convolutional neural networks, e.g., ResNet [21]. We feed the response maps from the last convolutional layer of the encoder into each task-branch to decode pixel-level task-related information. To produce refining high-resolution predictions, we decode this convolutional features into higher-resolution feature maps, and then concatenate with the same-scale features at the encoder to feed into a residual block to produce task-specific features.

Next, we perform pattern-structure diffusion on three task-specific feature maps. Concretely, intra-task PSD (Section 3.3) is first performed on the decoded features to transmit long-distance context information within each task, then inter-task PSD (Section 3.4) is used for two different tasks to mutually absorb the counterpart structures. In order for high-efficient PSD, we construct small graphlets on pixel-level local pattern regions instead of the large-scale or global region, which is introduced in Section 3.2. Further, we derive a recursive process on graphlets to propagate into long-distance positions. For inter-task PSD, we correlate those paired graphlets at the common position and weightedly transfer the structure information into the target task. In virtue of the joint PSD on intra-task and inter-task, pattern structures could be widely spread among long-range contexts within/across the three tasks.

Repeatedly, we can continue to upscale feature maps and perform the above decoding process to produce a higher feature scale of our requirement for the final pixel-level prediction. Such a coarse-to-fine process is supervised under multi-loss functions followed by a convolution prediction layer at each scale, where the details are given in Section 3.5.

### 3.2. Local Pattern-Structure Definition

Let us denote the decoded multi-channel feature maps of depth, segmentation, surface normal tasks respectively with $\mathbf{X}^{\mathcal{T}_d}, \mathbf{X}^{\mathcal{T}_s}, \mathbf{X}^{\mathcal{T}_n} \in \mathbb{R}^{H\times W\times C}$. Here $H, W, C$ represent height, width and channel number respectively. We characterize each local pattern with the pixel-level correlations therein, which is named as pattern-structure. To conveniently illustrate the construction of the pattern-structure, we omit the superscript $\mathcal{T}$ in the following description.

At each spatial position of the multi-channel feature $\mathbf{X}$, we can crop out a $l \times l$ square region and denote local pattern as $\mathbf{X}_{P_i} \in \mathbb{R}^{l\times l\times C}$, where $P_i$ means the cropped pattern at position $i$. For simplification, we abuse the notation $\text{vec}(\mathbf{X}_{P_i})$: $\mathbb{R}^{l\times l\times C} \rightarrow \mathbb{R}^{l^2\times C}$, which vectorizes the 2D spatial dimensions in the row-by-row stacking way. For each local patch pattern, we construct a graphlet $\mathcal{G}_{P_i} = \{\mathcal{V}_{P_i}, \mathcal{A}_{P_i}, \mathcal{X}_{P_i}\}$, where one vertex $v_i \in \mathcal{V}_{P_i}$ corresponds to one pixel position, the adjacent matrix $\mathcal{A}_{P_i} \in \mathbb{R}^{l^2\times l^2\times C}$ defines the edge-connection correlation, and $\mathcal{X}_{P_i} = \text{vec}(\mathbf{X}_{P_i}) \in \mathbb{R}^{l^2\times C}$ is the feature matrix. Formally, the adjacent matrix $\mathcal{A}_{P_i}$ of local patch pattern is defined as

$$[\mathcal{A}_{P_i}]_{jk} = \exp\{-\frac{\|[\text{vec}(\mathbf{X}_{P_i})]_j - [\text{vec}(\mathbf{X}_{P_i})]_k\|^2}{\sigma^2}\}, \quad (1)$$

$$\text{s.t. }, i = 1, 2, \ldots, H \times W, \quad j, k = 1, 2, \ldots, l^2, \quad (2)$$

where $[\cdot]_j$ takes the $j$ row of the input matrix, $[\mathcal{A}_{P_i}]_{jk}$ records the pattern-correlation between the position $j$ and $k$ at the $i$-th patch pattern, and $\sigma$ ($\sigma^2 = 2$ as default) is an exponential factor. The similar the patterns of node $j$ and $k$ are, the larger the corresponding value in $[\mathcal{A}_{P_i}]_{jk}$ is. Thus, $\mathcal{A}_{P_i}$ represents the local structure, which are recurred high-frequently and can be used for intra-task/inter-task pattern propagation to boost the performance based on our above observation.

As the global representation, we can collect all local structures into an entire graph defined on feature map $\mathbf{X}$, denoted as $\mathcal{G} = \{\mathcal{V}, \mathcal{A}, \mathcal{X}\}$, where $|\mathcal{V}| = H \times W$ is the set of all vertices w.r.t spatial positions, $\mathcal{X} = \text{vec}(\mathbf{X})$ is the vertorized feature map. The adjacent matrix of the whole graph is written as

$$\mathcal{A} = \text{diag}[\text{vec}(\mathcal{A}_{P_1})^\intercal; \text{vec}(\mathcal{A}_{P_2})^\intercal; \cdot; \text{vec}(\mathcal{A}_{P_{H^2W^2}})^\intercal], \quad (3)$$

where $\text{vec}(\mathcal{A}_{P_j})$: $\mathbb{R}^{l^2\times l^2\times C} \rightarrow \mathbb{R}^{l^4\times C}$ is similar to the above definition, $\text{diag}(\cdot)$ is the diagonalization on block-wise matrices. Obviously, the global matrix $\mathcal{A}$ is sparse because most values are zeros and only those positions w.r.t local patches are non-zeros values. Concretely, the number of non-zero values is $l^2HW \ll H^2W^2$, where $l \ll H, W$ is size of patch kernels. Thus, the calculation on sparse matrix could be utilized to speed up in practice. The local structures can not only benefit for high-efficient computation and low-memory requirement, but also be spread to global region after high-order calculation as introduced in the following section.

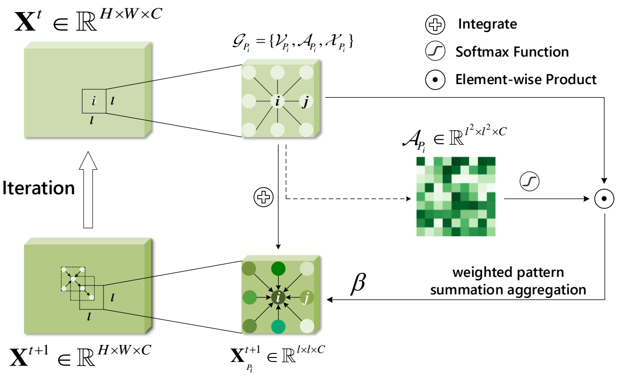

$\mathbf{X}^t \in \mathbb{R}^{H\times W\times C}$
$\mathcal{G}_{P_i} = \{\mathcal{V}_{P_i}, \mathcal{A}_{P_i}, \mathcal{X}_{P_i}\}$
Integrate
Softmax Function
Element-wise Product
i
l
j
$\mathcal{A}_{P_i} \in \mathbb{R}^{l^2\times l^2\times C}$
Iteration
$\beta$
weighted pattern summation aggregation
$\mathbf{X}^{t+1} \in \mathbb{R}^{H\times W\times C}$
$\mathbf{X}^{t+1}_{P_i} \in \mathbb{R}^{l\times l\times C}$

Figure 3: Intra-task diffusion process. A graphlet $\mathcal{G}_{P_i}$ is constructed based on a local region. The pattern structures $\mathcal{A}_{P_i}$ are used to diffuse those adjacent patterns $\mathbf{X}_{P_i}$ to produce new patterns through matrix multiplication, which then are weightedly summarized to original patterns as the enhanced responses. This process is recursively performed for long-distance diffusions.

### 3.3. Intra-Task PSD

The purpose of intra-task diffusion is to capture long-distance semantic information by diffusing local patterns in single task so as to enhance the task-specific patterns. To reduce the influence of scales for different local structures, we normalize the adjacent correlations in each $\mathcal{A}_{P_i}$ to the sum 1, i.e.,

$$\mathcal{A}_{P_i} \leftarrow \mathcal{A}_{P_i}/(\mathbf{1}^\intercal\mathcal{A}_{P_i}\mathbf{1}), \quad (4)$$

where $\mathbf{1}$ is a column vector with all values equal to 1.

In order to propagate information, we take the summation aggregation to weight the patterns of those adjacent vertices, formally,

$$v_i \leftarrow \sum_{j\in\mathcal{N}(v_i)} w(v_i, v_j)f(v_j), \quad (5)$$

where the neighbor set $\mathcal{N}(v_i)$ and the weight $w(v_i, v_j)$ are determined by the adjacent relations $\mathcal{A}_{P_i}$ computed above, $f$ denotes the feature extraction function. The intra-task diffusion process is shown in Fig. 3. To propagate long-distance pattern information, we can recursively iterate the aggregation process in Eqn. (5) and spread local patterns to more distant regions. Concretely, we formulate the recursive process in the following matrix formula,

$$[\text{vec}(\mathbf{X}^{(t+1)})]_i =$$

$$[\mathrm{vec}(\mathbf{X}^{(t)})]_i + \beta \cdot \sum_{j \in \mathcal{N}(v_i)} \mathcal{A}_{ij} \times [\mathrm{vec}(\mathbf{X}^{(t)})]_j, \quad (6)$$

where $t = 1, \cdots, T$ is the iterate step, and $\beta$ (Here $\beta = 0.05$ as [55]) is the balance factor. In the above equation, we take the residual connection, in which the aggregated feature is integrated with the feature of reference vertex by the weight parameter $\beta$. For each iteration, the diffused receptive field will enlarge one time. After multi-step iterations, a local pattern can be spread into those distant regions. Due to locality also sparsity of edge connection, the computation complexity of an iteration step depends on the edge number, i,e., $O(l^2|\mathcal{V}|C)$, where $l^2 \ll |\mathcal{V}|$. Thus, the intra-task diffusion with $T \ll |\mathcal{V}|$ iterations has the complexity $O(l^2 T|\mathcal{V}|C)$, which is obviously lower than the global connectivity with dense edges $O(|\mathcal{V}|^3 C)$. Moreover, the memory requirement, $O(l^2|\mathcal{V}|C)$, is relatively lower in contrast to the global dense connection which is $O(|\mathcal{V}|^2 C)$.

Finally, intermediate features of multiple steps is collected to enhance the local patterns, which can be formulated as

$$[\mathrm{vec}(\widetilde{\mathbf{X}})]_i = g(\Gamma([\mathrm{vec}(\mathbf{X}^{(1)})]_i, \cdot, [\mathrm{vec}(\mathbf{X}^{(T)})]_i), \Theta), \quad (7)$$

where ’$\Gamma$’ means feature concatenation, ’$g$’ is a non-linear function with the parameter $\Theta$ to be learnt, e.g., one $1 \times 1$ convolution layer followed by the ReLU activation unit. Therefore, the new produced feature $\widetilde{\mathbf{X}}$ ensembles those local pattern structures within different-scale receptive fields.

### 3.4. Inter-Task PSD

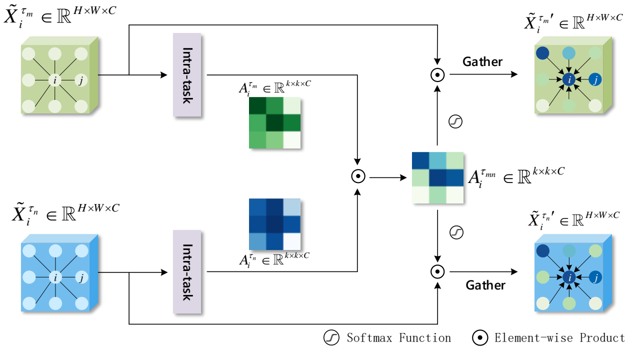
For the same input, the different-task pixel-level predictions own the similar local pattern structures at those corresponding positions, which implies some latent cues to correlate different tasks. To this end, we attempt to transfer local pattern structures from one task to another task to achieve cross-task pattern propagation. In Fig. 4, we show the main process of inter-task pattern-structure diffusion. Below we take segmentation as the target task, and propagate the information of the other two tasks into the segmentation task. Formally, we derive the pattern at the $i$-th position as follows

$$\begin{aligned}
[\mathrm{vec}(\widetilde{\mathbf{X}}^{\mathcal{T}_s})]_i \leftarrow & [\mathrm{vec}(\widetilde{\mathbf{X}}^{\mathcal{T}_s})]_i \\
& + \beta_{\mathcal{T}_{sd}} \cdot \sum_{j \in \mathcal{N}(v_i)} \mathcal{A}_{ij}^{\mathcal{T}_{sd}} \times [\mathrm{vec}(\widetilde{\mathbf{X}}^{\mathcal{T}_s})]_j \\
& + \beta_{\mathcal{T}_{sn}} \cdot \sum_{j \in \mathcal{N}(v_i)} \mathcal{A}_{ij}^{\mathcal{T}_{sn}} \times [\mathrm{vec}(\widetilde{\mathbf{X}}^{\mathcal{T}_s})]_j, \quad (8)
\end{aligned}$$

$$\text{s.t. ,} \quad \mathcal{A}_{P_i}^{\mathcal{T}_{sd}} = \mathcal{A}_{P_i}^{\mathcal{T}_s} \odot \mathcal{A}_{P_i}^{\mathcal{T}_d} / F_{P_i}^{\mathcal{T}_{sd}}, \quad (9)$$

$$\mathcal{A}_{P_i}^{\mathcal{T}_{sn}} = \mathcal{A}_{P_i}^{\mathcal{T}_s} \odot \mathcal{A}_{P_i}^{\mathcal{T}_n} / F_{P_i}^{\mathcal{T}_{sn}}, \quad (10)$$

where $\odot$ is the element-wise product operation, $\{\beta_{\mathcal{T}_{sd}}, \beta_{\mathcal{T}_{sn}}\}$ are the balance factors, $\{F_{P_i}^{\mathcal{T}_{sd}}, F_{P_i}^{\mathcal{T}_{sn}}\}$ are the normal factors to constrain the sum of all elements equal to 1, $\{\mathcal{A}_{P_i}^{\mathcal{T}_s}, \mathcal{A}_{P_i}^{\mathcal{T}_n}, \mathcal{A}_{P_i}^{\mathcal{T}_d}\}$ are the adjacent matrices of the patch $P_i$ for segmentation, surface normal and depth feature maps respectively. In the above Eqns. (9) and (10), the cross-task adjacent structures $\{\mathcal{A}_{P_i}^{\mathcal{T}_{sd}}, \mathcal{A}_{P_i}^{\mathcal{T}_{sn}}\}$ are the transferred structures from depth and surface normal tasks respectively, which are adaptively regularized by weighting structures therein. The strong weight of the edge can be enhanced while the weak weight can be attenuated further. In Eqn. (8), the inter-task diffusion towards to segmentation absorbs the structure information from depth and surface normal. As the joint learning with intra-task PSD, actually, the inter-task diffusion also integrates the long-distance pattern information of counterpart tasks. Similar to the above Eqn. (7), we concatenate the diffused feature and the original feature (i.e.,the intra-task feature) to feed into one $1 \times 1$ convolution layer to reduce the number of channels, which followed by non-linear activation unit such as ReLU. Accordingly, the transfer to other tasks follows the same process.



$\tilde{X}_i^{\tau_m} \in \mathbb{R}^{H\times W\times C}$
$\tilde{X}_i^{\tau_m}{}' \in \mathbb{R}^{H\times W\times C}$
Intra-task
$A_i^{\tau_m} \in \mathbb{R}^{k\times k\times C}$
Gather
$A_i^{\tau_{mn}} \in \mathbb{R}^{k\times k\times C}$
$\tilde{X}_i^{\tau_n} \in \mathbb{R}^{H\times W\times C}$
$\tilde{X}_i^{\tau_n}{}' \in \mathbb{R}^{H\times W\times C}$
Intra-task
$A_i^{\tau_n} \in \mathbb{R}^{k\times k\times C}$
Gather
i
j
Softmax Function
Element-wise Product

Figure 4: Inter-task diffusion process. The cross-task correlations ($\mathcal{A}_{P_i}^{\mathcal{T}_{sd}}$) are defined on intra-task pattern structures ($\mathcal{A}_{P_i}^{\mathcal{T}_s}$, $\mathcal{A}_{P_i}^{\mathcal{T}_d}$). According to $\mathcal{A}_{P_i}^{\mathcal{T}_{sd}}$, local structures can be adaptively transferred from one task to another task.

### 3.5. Loss Function

For different tasks, we take task-specific loss functions. Following a state-of-the-art depth estimation algorithm [27], we use berHu loss for the depth supervision. As for semantic segmentation and surface normal, cross-entropy loss and L1 loss has been adopted respectively.

## 4. Experiment

### 4.1. Datasets

**NYUD-v2.** The NYUD-v2 dataset [35] is a popular indoor-scene RGB image dataset, captured with a Microsoft Kinect. There are only 1449 selected frames from 40 classes labeled for segmentation. Following the standard setting [14], we use 795 images to train our model and 654 images to test the final performance. In addition, we follow the method in [16, 38], randomly sampling around 12k images and generating the surface normal ground truth.

Thus, more data can be used for training the joint depth and surface normal model.

**SUNRGB-D.** The SUNRGB-D dataset [40] is a very large and challenging dataset containing 10355 RGB-D images of indoor scenes. These images are divided into 37 classes including wall, table, floor, etc. All these images have both segmentation and depth labels but no surface normal labels. Therefore, we train our jointly predicting segmentation and depth model with 5285 images and test on 5050 images according to the official documents.

## 4.2. Implementation Details

**Training.** We implement our proposed model on Pytorch with double NVIDIA GeForce RTX2080Ti (12GB of GPU memory for each). We build our framework based on ResNet-50 [21] which is pre-trained on the ImageNet classification task [12]. Our initial learning rate is 1e-4/0.01 for parameters of pre-trained layers and others respectively and decay to 1e-5/0.01 during fine-tuning process. We use a momentum of 0.9 and a weight decay of 1e-4. The network is trained on RGB images for depth, segmentation and surface normal in an end-to-end manner. In order to further speed up and reduce the computation and memory cost, we only concern on the connection of the center node with others within a local graphlet that makes the adjacent matrix more sparse and aggregate all channels of the adjacent matrix instead of taking a channel-wise calculation, *i.e.*, $\mathcal{A}_{P_i} \in \mathbb{R}^{l \times l}$. The original frames of 640×480 pixels are center-cropped to 416×416. To increase the diversity of data, we have taken the same data augmentation strategy as [32]: scaling, flipping, cropping and rotating. For SUN-RGBD dataset, we train the model for 50 epochs and fine-tune it for 30 epochs. For NYUD-v2 dataset, the joint depth-segmentation model is trained for 50 epochs and fine-tuned for another 25 epochs with 12k images. As to the three-task joint model, 200 epochs are firstly taken and 100 epochs are for fine-tuning.

**Metrics.** For the evaluation of depth estimation, we follow the previous works [14, 27] and use the metrics including: root mean square error (RMSE), average relative error (REL), root mean square error in log space (RMSE-Log) and the accuracy with threshold $\delta$, where $\delta \in \{1.25, 1.25^2, 1.25^3\}$. For semantic segmentation, we take the same metrics as [10, 31]: pixel accuracy (PixAcc), mean accuracy (mAcc) and mean intersection over union (mIoU). As for surface normal, we use the following metrics: mean of angle error (Mean), medians of the angle error (Median), root mean square error for normal (Nor-RMSE), and pixel accuracy as percentage of pixels with angle error below threshold $\eta$, where $\eta \in \{11.25°, 22.50°, 30°\}$.

## 4.3. Comparison with the State-of-the-Arts

In this section, we compare our proposed method with various state-of-the-art methods for depth estimation, semantic segmentation and surface normal respectively. In each experiment, we set node number $= 9$ (i.e., region size $= 3 \times 3$) and the iteration step is 9. All the following experiments adopt ResNet-50 as backbone.

**Semantic segmentation.** The comparisons of semantic segmentation are made on widely-used NYUD-v2 and SUNRGB-D dataset. The superior or competitive comparison results on NYUD-v2 dataset are shown in Table 1. Note that here most methods are RGB-D methods which directly take depth map as a source of input. Conversely, our model trained for three tasks only takes 795 RGB images as input which achieves the best PixAcc (outperform TRL [53] by 0.8%) and mIoU (outperform D-CNN [47] by 2.6%) but slightly poor in mAcc than D-CNN [47]. This may due to imperfect depth predictions. Although our PSD can obtain impressive depth estimation results, the predictions are still not as precise as ground truth which results in negative effects on the segmentation predictions. For SUNRGB-D dataset, we train our model for depth and segmentation. As illustrated in Table 2, we can observe that our method is slightly weaker than RDF-ResNet152 [36] on mAcc but superior on PixAcc and mIoU. This is may due to the aforementioned reason as well. Meanwhile, RDF-ResNet152 uses stronger network backbone than ours with ResNet-50. Quantitative results are shown in Fig. 5. All these results demonstrate that our PSD can boost segmentation via information from other tasks.

**Depth estimation.** We mainly compare the proposed PSD with state-of-the-arts for depth estimation on NYUD-v2 dataset. As illustrated in Table 3, our model trained jointly for three tasks (PSD-$\mathcal{T}_d$+$\mathcal{T}_s$+$\mathcal{T}_n$) is able to deliver comparable results against previous state-of-the-art methods though only using 795 images for training. As to the model trained jointly for depth and normal (PSD-$\mathcal{T}_d$+$\mathcal{T}_n$) with more data (12k images), the performance gets better and achieves the best on most metrics except REL and $\delta_1$. Actually, AdaD-S [34] and DORN [18] use large scale data (120k/100k images) for training which is highly beneficial for the model. As quantitative results are shown in Fig. 6, the predictions are more precise which demonstrates that the performance of our proposed PSD is superior.

**Surface normal.** We mainly evaluate our method for surface normal prediction on NYUD-v2 dataset. The results are listed in Table 4. Our PSD consistently outperforms previous approaches on most metrics except $\eta_3 = 30°$. The results well indicate PSD can utilize the task-specific and cross-task correlations for improving the current task performance. Quantitative results are shown in Fig. 7 from which we can find that predictions of our PSD are better and contain more details.

Table 1: Comparisons with state-of-the-art semantic segmentation approaches on NYUD-v2 dataset

| Method | data | PixAcc | mAcc | mIoU |
|---|---|---|---|---|
| FCN [24] | RGB | 60.0 | 49.2 | 29.2 |
| Lin *et al.* [32] | RGB | 70.0 | 53.6 | 40.6 |
| Mousavian *et al.* [33] | RGB | 68.6 | 52.3 | 39.2 |
| TRL [53] | RGB | 76.2 | 56.3 | 46.4 |
| RefineNet [31] | RGB | 72.8 | 57.8 | 44.9 |
| 3DGNN [37] | RGBD | - | 55.7 | 43.1 |
| RDFNet-ResNet50 [36] | RGBD | 74.8 | 60.4 | 47.7 |
| Cheng *et al.* [10] | RGBD | 71.9 | 60.7 | 45.9 |
| Deng *et al.* [13] | RGBD | 63.8 | - | 31.5 |
| Eigen & Fergus [14] | RGBD | 65.6 | 45.1 | 34.1 |
| D-CNN [47] | RGBD | - | **61.1** | 48.4 |
| PSD-ResNet50 | RGB | **77.0** | 58.6 | **51.0** |

Table 2: Comparisons with state-of-the-art semantic segmentation approaches on SUNRGB-D dataset

| Method | data | PixAcc | mAcc | mIoU |
|---|---|---|---|---|
| SegNet [4] | RGB | 72.6 | 44.8 | 31.8 |
| Lin *et al.* [32] | RGB | 78.4 | 53.4 | 42.3 |
| Bayesian-SegNet [25] | RGB | 71.2 | 45.9 | 30.7 |
| RefineNet [31] | RGB | 80.4 | 57.8 | 45.7 |
| TRL [53] | RGB | 83.6 | 58.9 | 50.3 |
| PAP [55] | RGB | 83.8 | 58.4 | 50.5 |
| Cheng *et al.* [10] | RGBD | - | 58.0 | - |
| 3DGNN [37] | RGBD | - | 57.0 | 45.9 |
| D-CNN [47] | RGBD | - | 53.5 | 42.0 |
| RDF-ResNet152 [36] | RGBD | 81.5 | **60.1** | 47.7 |
| PSD-ResNet50 | RGB | **84.0** | 57.3 | **50.6** |

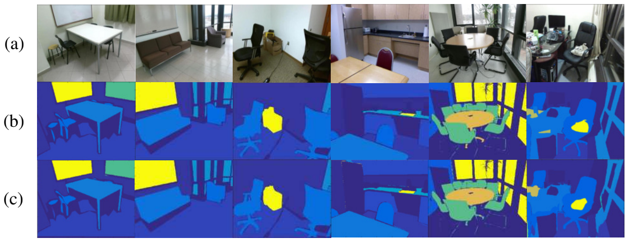


Figure 5: Visual results of semantic segmentation on SUNRGB-D dataset. (a) original RGB images; (b) ground truth; (c) our predictions

Table 3: Comparisons with state-of-the-art depth estimation approaches on NYUD-v2 dataset

| Method | data | Lower is better | | | Higher is better | | |
|---|---|---|---|---|---|---|---|
| | | RMSE | REL | RMSE-log | $\delta_1 = 1.25$ | $\delta_2 = 1.25^2$ | $\delta_3 = 1.25^3$ |
| PAD-Net [49] | 795 | 0.582 | 0.120 | - | 0.817 | 0.954 | 0.987 |
| Wang *et al.* [45] | 795 | 0.745 | 0.220 | 0.262 | 0.605 | 0.890 | 0.970 |
| Li *et al.* [29] | 795 | 0.821 | 0.232 | - | 0.621 | 0.886 | 0.968 |
| Xu *et al.* [51] | 795 | 0.593 | 0.125 | - | 0.806 | 0.952 | 0.986 |
| Lee *et al.* [28] | 795 | 0.538 | 0.148 | 0.180 | 0.837 | **0.971** | 0.994 |
| PAP [55] | 795 | 0.530 | 0.142 | 0.190 | 0.818 | 0.957 | 0.988 |
| Eigen *et al.* [11] | 120k | 0.877 | 0.214 | 0.285 | 0.611 | 0.887 | 0.971 |
| Eigen & Fergus [14] | 120k | 0.641 | 0.158 | 0.214 | 0.769 | 0.950 | 0.988 |
| DORN [18] | 120k | 0.509 | 0.115 | - | 0.828 | 0.965 | 0.992 |
| AdaD-S [34] | 100k | 0.506 | **0.114** | - | **0.856** | 0.966 | 0.991 |
| Multu-scale CRF [50] | 95k | 0.586 | 0.121 | - | 0.811 | 0.954 | 0.987 |
| GeoNet [38] | 16k | 0.569 | 0.128 | - | 0.834 | 0.960 | 0.990 |
| Laina *et al.* [27] | 12k | 0.573 | 0.127 | 0.194 | 0.811 | 0.953 | 0.988 |
| TRL [53] | 12k | 0.501 | 0.144 | 0.181 | 0.815 | 0.962 | 0.992 |
| PSD-$\mathcal{T}_d$+$\mathcal{T}_s$+$\mathcal{T}_n$ | 795 | 0.510 | 0.149 | 0.184 | 0.810 | 0.958 | 0.990 |
| PSD-$\mathcal{T}_d$+$\mathcal{T}_n$ | 12k | **0.488** | 0.132 | **0.172** | 0.840 | 0.966 | **0.994** |

## 4.4. Ablation Study

In this section, we perform extensive experiments to verify the efficacy of our method. All the following experiments adopt ResNet-18 as backbone and are trained for three tasks on the NYUD-v2 dataset.

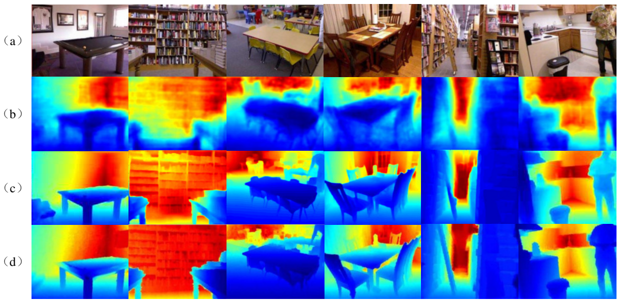


Figure 6: Visual results of depth on NYUD-v2 dataset. (a) original RGB images; (b) predictions of [51]; (c) our predictions; (d) ground truth.

Table 4: Comparisons with state-of-the-art surface normal approaches on NYUD-V2 dataset

| Method | Lower is better | | | Higher is better | | |
|---|---|---|---|---|---|---|
| | Mean | Median | Nor-RMSE | $\eta_1 = 11.25°$ | $\eta_2 = 22.5°$ | $\eta_3 = 30°$ |
| 3DP [16] | 35.3 | 31.2 | - | 16.4 | 36.6 | 48.2 |
| 3DP(MV) [16] | 36.3 | 19.2 | - | 39.2 | 52.9 | 57.8 |
| Eigen & Fergus [14] | 23.7 | 15.5 | - | 39.2 | 62.0 | 71.1 |
| UNFOLD [17] | 35.1 | 19.2 | - | 37.6 | 53.3 | 58.9 |
| Deep3D [48] | 26.9 | 14.8 | - | 42.0 | 61.2 | 68.2 |
| SkipNet [5] | 19.8 | 12.0 | 28.2 | 47.9 | 70.0 | 77.8 |
| Discr. [26] | 33.5 | 23.1 | - | 27.7 | 49.0 | 58.7 |
| SURGE [46] | 20.6 | 12.2 | - | 47.3 | 68.9 | 76.6 |
| Liao *et al.* [30] | 19.7 | 12.5 | - | 45.8 | 72.1 | **80.6** |
| GeoNet [38] | 19.0 | 11.8 | 26.9 | 48.4 | 71.5 | 79.5 |
| PSD-ResNet50 | **18.2** | **11.5** | **24.9** | **48.9** | **72.7** | 79.9 |

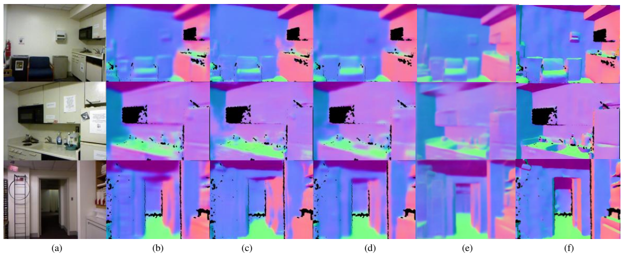


Figure 7: Visual results of surface normal on NYUD-v2 dataset. (a) original RGB images; (b) predictions of [14]; (c) predictions of [5]; (d) predictions of [38]; (e) our predictions; (f) ground truth.

**Single-task versus multi-task learning.** To verify the effectiveness of jointly predicting segmentation, depth and surface normal with our PSD method, we first predict each task separately and then jointly predict three tasks with our intra-task and inter-task PSD. To reflect the essential comparisons, we conduct the experiments on a single scale ( $\frac{1}{16}$ scale of input). As is shown in Table 5, the performance of joint-task model outperforms the single-task model by 7.0% totally, and each task is indeed promoted after joint learning.

**Analysis on network settings.** We perform a series of experiments to evaluate the influence of each module in our

Table 5: Results of single-task versus multi-task learning

| Settins | mIoU | RMSE | Nor-RMSE |
|---|---|---|---|
| Sementation only | 42.4 | - | - |
| Depth only | - | 0.572 | - |
| Surface normal only | - | - | 29.0 |
| Three task jointly | **44.9** | **0.548** | **26.9** |

Table 6: Analysis on network settings on NYUD-v2 dataset

| Model | mIoU | RMSE | Nor-RMSE |
|---|---|---|---|
| baseline | 40.9 | 0.585 | 29.2 |
| naive fusion | 42.1 | 0.576 | 28.7 |
| cross-stitch [22] | 43.4 | 0.554 | 28.4 |
| + intra-task PSD | 42.2 | 0.578 | 28.3 |
| + inter-task PSD | 43.1 | 0.565 | 27.8 |
| + intra&inter-task PSD on small scale | 44.9 | 0.548 | 26.9 |
| + intra&inter-task PSD on middle scale | 46.3 | 0.534 | 26.6 |
| + intra&inter-task PSD on big scale | 47.2 | 0.526 | 26.1 |
| + intra&inter-task PSD on all scales | **50.0** | **0.498** | **25.8** |

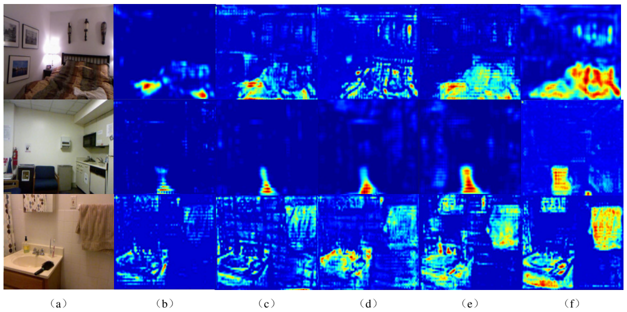


Figure 8: Visualization of response maps. (a) original RGB image; (b) baseline; (c) naive fusion; (d) intra-task PSD; (e) inter-task PSD; (f) intra&inter-task PSD on middle scale

proposed network. As shown in Table 6, the first five rows show results of experiments conducted on $\frac{1}{16}$ scale of the input. The **baseline** denotes the model jointly trained on three tasks without any interactiveness. We also compare two feature-fusion approaches under the same settings, i.e., **naive fusion** and **cross-stitch** [22]. The former directly concatenates cross-task features. The latter adds the cross-stitch unit to baseline. We can observe that both performances are poorer than ours. The reason behind it should be that, these two methods just combine features, but not mine/utilize pattern structures. Next, we add intra-task or inter-task PSD to baseline. The performance improvement indicates the benefit of each module. Further, we study the influence of different scales. The results are reported in the last four rows of Table 6. The larger scale results in a better performance, because the finer patterns at larger scale can be decoded to better estimate pixel-level subtle information. In addition, we show some qualitative visual results in Fig. 8. We can find that both intra-task and inter-task PSD can well boost pixel-level semantic understanding.

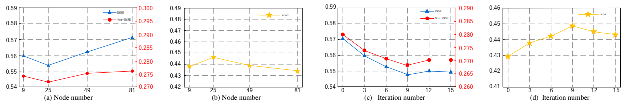


Figure 9: (a) RMSE (left axis, lower the better) and Nor-RMSE (right axis, lower the better) and (b) mIoU (higher the better) of PSD w.r.t graphlet size; (c) RMSE and Nor-RMSE and (d) mIoU of PSD w.r.t iteration numbers.

**Analysis on graphlet size.** Here we perform experiments to investigate the influence of graphlet size (i.e., the node number). From Fig. 9, we can observe that the performance becomes better as the graphlet size increases, then reaches the best at the size of 25. The reason should be two folds: i) as the graphlet size increases, more pattern structures will be diffused which makes the correlations become more complicated and is more sensitive to feature response, ii) to some extend, some details may be fuzzed up as the diffused receptive field becomes larger. In addition, the size of 25 brings limited improvements over size of 9 while cost heavier memory and computation, which can be seen as a trade-off.

**Analysis on the iteration number of diffusion.** In Fig. 9, we show the results of different iteration numbers. Here we set intra-task and inter-task PSD only on the $\frac{1}{16}$ scale of input with the graphlet size 9. We can observe that the performance increases at first and tends to saturate when the iteration number is 9. It reveals that the model can capture longer distance correlations as the iteration number increases. Nevertheless, transmitting too long-distance patterns might bring negative impacts for the current region pattern to a certain extent. This reason might be that, in the pixel-level prediction task, each pixel highly depends on its neighbours instead of too far away pixels unless the similar pattern structures.

## 5. Conclusion

In this paper, we proposed the pattern-structure diffusion (PSD) framework for multi-task learning. Two types of pattern-structure diffusion stage were designed to effectively mine and propagate the relationships within/across tasks. In virtue of these two PSD strategies, the interactions across tasks can be connected with pattern-structures as well as the correlations. Besides, graphlets are utilized to model the pattern-structures which can bring the additional benefit of low computation and memory burden. Finally, all these diffusion models were capsulated into a PSD layer, which can be flexible to be incorporated those general deep networks. Extensive experiments verified we can benefit from pattern-structure diffusion for joint prediction of depth, segmentation and surface normal. In the future, we may generalize our method to other tasks in computer vision.

## Acknowledgement

This work was supported by the National Natural Science Foundation of China (Grants Nos.61772276, 61906094, 61972204, U1713208) and the Natural Science Foundation of Jiangsu Province (Grants No.BK20190019).

## References

[1] Jalali Ali, Sanghavi Sujay, Ruan Chao, and Ravikumar Pradeep K. A dirty model for multi-task learning. In *NeurIPS*, pages 964–972, 2010.

[2] Saxena Ashutosh, Sun Min, and Ng Andrew Y. Make3d: Learning 3d scene structure from a single still image. *TPAMI*, 31(5):824–840, 2008.

[3] Saxena Ashutosh, Chung Sung H, and Ng Andrew Y. Learning depth from single monocular images. In *NeurIPS*, pages 1161–1168, 2006.

[4] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.

[5] Aayush Bansal, Bryan Russell, and Abhinav Gupta. Marr revisited: 2d-3d alignment via surface normal prediction. In *CVPR*, pages 5965–5974, 2016.

[6] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *CVPR*, pages 2722–2730, 2015.

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 40(4):834–848, 2017.

[8] Po-Yi Chen, Alexander H. Liu, Yen-Cheng Liu, and Yu-Chiang Frank Wang. Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation. In *CVPR*, June 2019.

[9] Yuhua Chen, Wen Li, Xiaoran Chen, and Luc Van Gool. Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach. In *CVPR*, pages 1841–1850, 2019.

[10] Yanhua Cheng, Rui Cai, Zhiwei Li, Xin Zhao, and Kaiqi Huang. Locality-sensitive deconvolution networks with gated fusion for rgb-d indoor semantic segmentation. In *CVPR*, pages 3029–3037, 2017.

[11] Eigen David, Puhrsch Christian, and Fergus Rob. Depth map prediction from a single image using a multi-scale deep network. In *NeurIPS*, pages 2366–2374, 2014.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009.

[13] Zhuo Deng, Sinisa Todorovic, and Longin Jan Latecki. Semantic segmentation of rgbd images with mutex constraints. In *CVPR*, pages 1733–1741, 2015.

[14] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *CVPR*, pages 2650–2658, 2015.

[15] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004.

[16] David F Fouhey, Abhinav Gupta, and Martial Hebert. Data-driven 3d primitives for single image understanding. In *CVPR*, pages 3392–3399, 2013.

[17] David Ford Fouhey, Abhinav Gupta, and Martial Hebert. Unfolding an indoor origami world. In *ECCV*, pages 687–702. Springer, 2014.

[18] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, pages 2002–2011, 2018.

[19] Clement Godard, Mac Aodha Oisin, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth estimation. In *ICCV*, October 2019.

[20] Adam W Harley, Konstantinos G Derpanis, and Iasonas Kokkinos. Segmentation-aware convolutional networks using local attention masks. In *CVPR*, pages 5038–5047, 2017.

[21] Kaiming He, Xiangyu Zhang, Ren Shaoqing, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[22] Misra Ishan, Shrivastava Abhinav, Gupta Abhinav, and Hebert Martial. Cross-stitch networks for multi-task learning. In *CVPR*, pages 3994–4003, 2016.

[23] Yosinski Jason, Clune Jeff, Bengio Yoshua, and Lipson Hod. How transferable are features in deep neural networks? In *NeurIPS*, pages 3320–3328, 2014.

[24] Long Jonathan, Shelhamer Evan, and Darrell Trevor. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

[25] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*, 2015.

[26] L Ladickỳ, Bernhard Zeisl, and Marc Pollefeys. Discriminatively trained dense surface normal estimation. In *ECCV*, volume 2, page 4, 2014.

[27] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *3DV*, pages 239–248. IEEE, 2016.

[28] Jae-Han Lee and Chang-Su Kim. Monocular depth estimation using relative depth maps. In *CVPR*, pages 9729–9738, 2019.

[29] Bo Li, Chunhua Shen, Yuchao Dai, Anton Van Den Hengel, and Mingyi He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *CVPR*, pages 1119–1127, 2015.

[30] Shuai Liao, Efstratios Gavves, and Cees G. M. Snoek. Spherical regression: Learning viewpoints, surface normals and 3d rotations on n-spheres. In *CVPR*, June 2019.

[31] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR*, pages 1925–1934, 2017.

[32] Guosheng Lin, Chunhua Shen, Anton Van Den Hengel, and Ian Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*, pages 3194–3203, 2016.

[33] Arsalan Mousavian, Hamed Pirsiavash, and Jana Košecká. Joint semantic segmentation and depth estimation with deep convolutional networks. In *3DV*, pages 611–619. IEEE, 2016.

[34] Jogendra Nath Kundu, Phani Krishna Uppala, Anuj Pahuja, and R Venkatesh Babu. Adadepth: Unsupervised content

congruent adaptation for depth estimation. In *CVPR*, pages 2656–2665, 2018.

[35] Silberman Nathan, Hoiem Derek, Kohli Pushmeet, and Fergus Rob. Indoor segmentation and support inference from rgbd images. In *ECCV*, pages 746–760, 2012.

[36] Seong-Jin Park, Ki-Sang Hong, and Seungyong Lee. Rdfnet: Rgb-d multi-level residual feature fusion for indoor semantic segmentation. In *CVPR*, pages 4980–4989, 2017.

[37] Xiaojuan Qi, Renjie Liao, Jiaya Jia, Sanja Fidler, and Raquel Urtasun. 3d graph neural networks for rgbd semantic segmentation. In *CVPR*, pages 5199–5208, 2017.

[38] Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *CVPR*, June 2018.

[39] Anirban Roy and Sinisa Todorovic. Monocular depth estimation using neural regression forest. In *CVPR*, pages 5506–5514, 2016.

[40] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, pages 567–576, 2015.

[41] Gjorgji Strezoski, Nanne van Noord, and Marcel Worring. Many task learning with task routing. In *ICCV*, October 2019.

[42] Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab. Cnn-slam: Real-time dense monocular slam with learned depth prediction. In *CVPR*, pages 6243–6252, 2017.

[43] Gebru Timnit, Hoffman Judy, and Li Feifei. Fine-grained recognition in the wild: A multi-task domain adaptation approach. In *CVPR*, pages 1349–1358, 2017.

[44] Koen EA Van de Sande, Jasper RR Uijlings, Theo Gevers, and Arnold WM Smeulders. Segmentation as selective search for object recognition. In *ICCV*, pages 1879–1886. IEEE, 2011.

[45] Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Price Brian, and Yuille Alan L. Towards unified depth and semantic prediction from a single image. In *CVPR*, pages 2800–2809, 2015.

[46] Peng Wang, Xiaohui Shen, Bryan Russell, Scott Cohen, Brian Price, and Alan L Yuille. Surge: Surface regularized geometry estimation from a single image. In *NeurIPS*, pages 172–180, 2016.

[47] Weiyue Wang and Ulrich Neumann. Depth-aware cnn for rgb-d segmentation. In *ECCV*, pages 135–150, 2018.

[48] Xiaolong Wang, David Fouhey, and Abhinav Gupta. Designing deep networks for surface normal estimation. In *CVPR*, pages 539–547, 2015.

[49] Dan Xu, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *CVPR*, pages 675–684, 2018.

[50] Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In *CVPR*, pages 5354–5362, 2017.

[51] Dan Xu, Wei Wang, Hao Tang, Hong Liu, Nicu Sebe, and Elisa Ricci. Structured attention guided convolutional neural fields for monocular depth estimation. In *CVPR*, pages 3917–3925, 2018.

[52] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *ICCV*, October 2019.

[53] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Zequn Jie, Xiang Li, and Jian Yang. Joint task-recursive learning for semantic segmentation and depth estimation. In *ECCV*, pages 235–251, 2018.

[54] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Zequn Jie, Xiang Li, and Jian Yang. Joint task-recursive learning for rgb-d scene understanding. *TPAMI*, 2019.

[55] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan, Nicu Sebe, and Jian Yang. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *CVPR*, June 2019.

[56] Zhenyu Zhang, Chunyan Xu, Jian Yang, Junbin Gao, and Zhen Cui. Progressive hard-mining network for monocular depth estimation. *TIP*, 27(8):3691–3702, 2018.

[57] Zhenyu Zhang, Chunyan Xu, Jian Yang, Ying Tai, and Liang Chen. Deep hierarchical guidance and regularization learning for end-to-end depth estimation. *PR*, 83:430–442, 2018.

[58] Maria Zontak and Michal Irani. Internal statistics of a single natural image. In *CVPR*, pages 977–984. IEEE, 2011.