






A Novel Iterative Fusion Multi-Task Learning Framework for Solving Dense Prediction

Jiaqi Wang and Jianping Luo

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

February 14, 2024

A Novel Iterative Fusion Multi-Task Learning Framework for Solving Dense Prediction

Jiaqi Wang¹  and Jianping Luo¹  

Guangdong Key Laboratory of Intelligent Information Processing, Shenzhen Key Laboratory of Media Security and Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), shenzhen University, China
wangjiaqi2021@email.szu.edu.cn, ljp@szu.edu.cn

Abstract. Dense prediction tasks are hot topics in computer vision that aim to predict each input image pixel, such as Semantic Segmentation, Monocular Depth Estimation, Edge Estimation, etc. With advanced deep learning, many dense prediction tasks have been greatly improved. Multi-task learning is one of the top research lines to boost task performance further. Properly designed multi-task model architectures have better performance and minor memory usage than single-task models. This paper proposes a novel Multi-task Learning (MTL) framework with a Task Pair Interaction Module (TPIM) to tackle several dense prediction tasks. Different from most widely used MTL structures which share features on some specific layer and branch to task-specific layer, the output task-specific features are remixed via a TPIM to get more shared features in this paper. Due to joint learning, tasks are mutually supervised and provide rich shared information to each other for improving final results. The TPIM includes a novel Cross-task Interaction Block (CIB) which comprises two attention mechanisms, self-attention and pixel-wise global attention. In contrast with the commonly used global attention mechanism, an Iterative Fusion Block (IFB) is introduced to effectively fuse affinity information between task pairs. Extensive experiments on two benchmark datasets (NYUD-v2 and PASCAL) demonstrate that our proposal is effective in comparison to existing methods.

Keywords: dense prediction · multi-task learning · cross-task interaction · iterative fusion.

1 Introduction

Real-world is inundated with many complex problems that must tackle multiple tasks simultaneously. For example, if an automated vehicle wants to drive safely on the road, one must strictly detect and locate all the objects around and accurately understand traffic signs, lane lines, etc., in the scene [12]. Similarly, in human face detection [4], not only should the landmarks on the human face be located, but expression recognition, such as smiling and crying, is needed to detect human faces accurately. There are countless instances. Multi-task learning

(MTL) [3] is proposed for computationally efficiently solving multiple related tasks.

Traditional MTL methods fall into two categories [22,18]. One is that most parameters are shared among all the tasks with a tiny part of the private parameters, called the hard parameter sharing method. Shared information is propagated via the shared parameters, and task-specific outputs are obtained by independent private parameters [13,1]. The other is that each task has a complete and separate network, and features are fused by specific means such as weighted sum across tasks, named soft parameter sharing [16,9,19]. Recently, a variant of hard parameter sharing MTL has been proposed, i.e., to append a multi-modal distillation module at the end of typical hard parameter sharing MTL to improve the information exchange across tasks [24]. It is well-known that multi-modal data improve the performance of deep predictions [24]. For example, a Convolutional Neural Network (CNN) trained with RGB-D data perform better than trained with RGB data. However, obtaining depth data requires additional cost. An economical approach is to use a CNN to predict the depth maps and use them as input. Besides the depth maps, we can also use CNN to predict more related information. Inspired by this, previous works [24,26,27,21,2,25,14] propose to use a CNN-based MTL to obtain several related information, including but not limited to depth, semantic information, etc., and then use them as multi-modal input which are fed into the following CNN named multi-modal distillation module to fuse features from different tasks to have better MTL performance. It gains improvement with few parameters. However, we found that the previously proposed feature fusion methods are relatively simple and there is much room for improvement. In this paper, we propose a new type of attention-driven multi-modal distillation scheme for better cross-task information fusion.

PAP-Net [26] models relationships between pairs of pixels and uses obtained affinity maps to perform interaction across tasks. PAD-Net [24] and MTI-Net [21] prove the effectiveness of self-attention on MTL, i.e., one task can further mine meaningful representations by applying self-attention to the task itself to help the other tasks. DenseMTL [14] introduces an attention module based on PAD-Net, called correlation-guided attention, which calculates the correlation between features from two tasks to guide the construction of exchanged messages. ATRC [2] explores four attention-based contexts dependent on tasks' relations and use Neural Architecture Search (NAS) to find optimal context type for each source-target task pair. We partially follow the same direction as in works mentioned above, using self-attention and global attention in our model. In contrast to them, we propose an iterative mechanism called Iterative Fusion Block (IFB) that further fuses the pixel-wise affinity maps with the original features.

In summary, our contributions are threefold: (i) We propose a novel multi-modal distillation design, named Task Pair Interaction Module (TPIM) (Sec. 3.2) for MTL comprising several Cross-task Interaction Blocks (CIB). (ii) We introduce a new mechanism to fuse further the pixel-wise affinity maps between task-pair (Sec. 3.4), which we call Iterative Fusion Block (IFB). IFB adaptively integrates shared and task-specific features and retains the original features to

the greatest extent. (iii) Extensive experiments on the challenging NYUD-v2 [20] and PASCAL [8] datasets validate the effectiveness of the proposed method (Sec. 4.2). Our method achieves state-of-the-art results on both NYUD-v2 and PASCAL datasets. More importantly, the proposed method remarkably outperforms state-of-the-art works that optimize different tasks jointly.

2 Related works

2.1 Multi-Task Learning (MTL)

To learn common representations, MTL methods are classified into two paradigms, hard parameter sharing MTL and soft parameter sharing MTL [22,18]. The former typically comprises two stages. Architectures share the intermediate representations among the tasks at the first stage, usually a shared feature extractor, and branch to the independent task-specific representations layer in the second stage. Tasks used in soft parameter sharing have their network; cross-task interaction is conducted by bridging these networks. For example, [16] proposed to use a “cross-stitch” unit to combine features from different independent networks to adaptively learn a proper combination of shared and task-specific representations. Though all the previous works show great multi-task learning potential, they uncover a few challenges. Most notable is the negative transfer phenomenon [11], where learning some less related tasks jointly leads to degrading task performance. Some works [6,10] attribute negative transfer to not balancing the losses among independent tasks and introduce mechanisms to weigh the loss terms carefully. Kendall et al. [10] proposed to use each task’s homoscedastic uncertainty to balance the losses. Chen et al. [6] proposed an algorithm named GradNorm to tune the magnitude of each task’s gradients dynamically. Liu et al. [13] proposed Dynamic Weight Averaging (DWA) to weigh the tasks based on the task-specific losses dynamically.

2.2 Cross-task interaction mechanisms

Close to our work are methods that distill shared features from task-specific features. Inspired by the acknowledgment that multi-modal data improves the performance of dense predictions, PAD-Net [24] introduced a multi-modal distillation module to refine information across multiple tasks. Vandenhende et al. [21] extended PAD-Net [24] to multi-scale level to better utilize multi-scale cross-task interaction. Zhang et al. [26] proposed to obtain pixel-wise affinity maps of all tasks, which are then diffused to other tasks to perform cross-task interaction. Similarly, Zhou et al. [27] further proposed Pattern Structure Diffusion (PSD) to mine and propagate patch-wise affinities via graphlets. Lopes et al. [14] introduced a cross-task attention mechanism comprising correlation-guided attention and self-attention to carry out multi-modal distillation. ATRC [2] explores four attention-based contexts dependent on tasks’ relations and use Neural Architecture Search (NAS) to find optimal context type for each source-target task

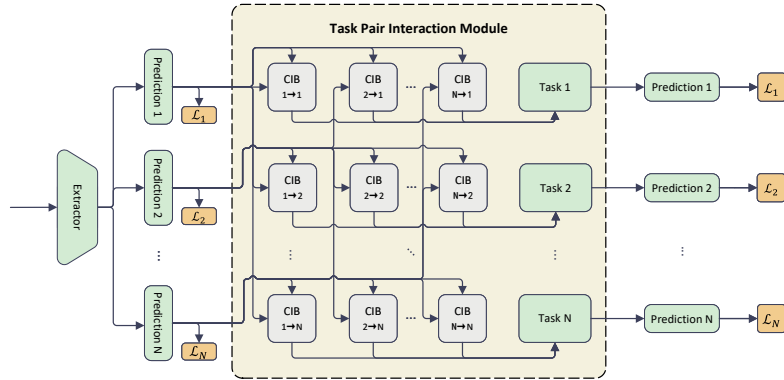


Fig. 1. Overview of our MTL framework with the proposed Task Pair Interaction Module (TPIM). Input features are first through a shared Extractor and several prediction modules. Consequently, the features from prediction modules are fed into the TPIM and fusion in pairs. Within TPIM, each task is routed as a target task to N Cross-task Interaction Blocks (CIB) (n -th row of CIBs) and as a source task to N CIBs (n -th column). The outputs of CIBs are summed together for each task independently (denoted as 'Task n ') and fed through the consequent prediction module ('Prediction n '). **Legend:** Green blocks denote modules with learned weights, and orange blocks denote loss functions. Best viewed in color.

pair. In this paper, we propose a novel cross-task attention mechanism to refine task-specific features. We use a self-attention to explore related features from the source task and a pixel-wise global attention to construct affinity information between task pairs. Besides, an iterative mechanism named Iterative Fusion Block (IFB) is introduced to deeply fuse cross-task affinity information with original task-specific information and combine the cross-task affinities with self-attention by addition operation to learn complementary representations for the target task.

3 Methods

This section will describe the proposed framework used to simultaneously figure out related dense prediction tasks. We first present an overview of the proposed framework and then introduce our framework's details.

3.1 Overall Structure

Fig. 1 shows our overall MTL structure for dense prediction tasks. The proposed MTL model consists of four main modules. The first is a shared feature extraction module that extracts shared information among tasks. The second is an intermediate prediction module, which takes the shared features extracted by the

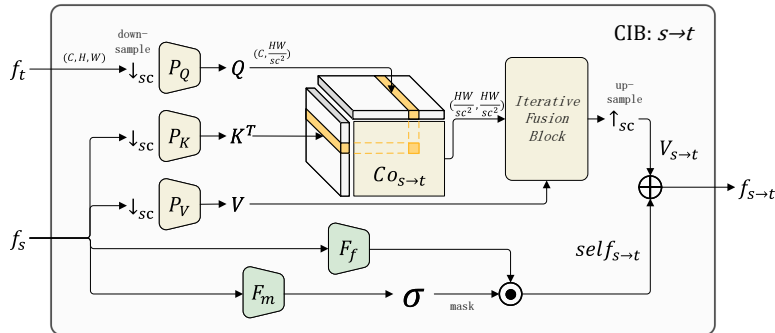


Fig. 2. Overview of our proposed Cross-task Interaction Block (CIB). CIB enables cross-task interaction between task pairs (s, t) . It relies on two attention mechanisms. First, pixel-wise global attention (yellow, upper part) to discover $V_{s \rightarrow t}$, the features from task s contributing to task t . Second, a self-attention (green, lower part) to discover complementary features $self f_{s \rightarrow t}$ from s . \downarrow_{sc} is a sc -times downscale for memory efficiency and reversely \uparrow_{sc} is a sc -times upscale operation. Best viewed in color.

previous module as input and outputs the prediction of the corresponding task. The third is a Task Pair Interaction Module (TPIM) which uses the predictions from the intermediate prediction module to carry out pair-wise feature interactions. The fourth is a final prediction module, consisting of N task-specific heads to decode the distilled information, to obtain task-specific predictions, where N is the number of tasks. The input of the proposed MTL structure is RGB images during training and testing, and the final output is N maps with the exact resolution as the input RGB images.

3.2 Task Pair Interaction Module (TPIM)

Building on recent works [24,26,14,21,27,2,25] illustrating the effectiveness of cross-task interaction, we propose a module that can capture spatial correlation of features at the pixel level while maintaining dedicated task-specific information. Fig. 1 depicts our TPIM, which helps to blend task-specific features to enhance cross-task information communication. TPIM comprises N^2 Cross-task Interaction Blocks (CIB) and N feature aggregation blocks. Considering i as the primary task, we perform an element-wise addition on the refined features $f_{j \rightarrow i} | j \in N$, contributing towards task i , in the feature aggregation block marked 'Task i '.

3.3 Cross-task Interaction Block (CIB)

This block seeks to capture the shared pair-wise task knowledge while preserving their non-shared knowledge. We are committed to exploiting features from

pair of tasks denoted (s, t) . Fig. 2 illustrates our components, which help knowledge distillation between tasks. CIB takes as input the task features (f_s, f_t) and returning the corresponding complementary features $f_{s \rightarrow t}$.

Note that, for task pair (i, i) , CIB is an identity transformation, i.e. $f_{i \rightarrow i} = f_i$.

Considering here t as the target task, and s as the source task, we aim to capture related features from task s to improve the performance of task t . For achieving this, two attentions are employed: (i) pixel-wise global attention, which is used to obtain the spatial correlation between two tasks, and (ii) self-attention on source task s to self-discover supplementary features for target task t . We fuse two attentions via an equally element-wise addition operation. Visualizations of the two attentions are colored in yellow and green in Fig. 2, respectively.

Note that each attention contributes differently to the target task t . The pixel-wise global attention relies on identifying shared s and t knowledge, while the other relies on exclusive s knowledge.

Self-attention We employ a spatial self-attention, see green blocks in Fig. 2, which aims to self-discover important information of source task s that may be helpful to solve relative task t . We formulate self-attention as follows:

$$self_{s \rightarrow t} = F_s(f_s) \odot \sigma(F_m(f_s)) \quad (1)$$

where $F_*(\cdot)$ is a 3×3 convolution supervised by the target task t to learn to extract related information from features f_s , $\sigma(\cdot)$ the sigmoid function to normalize the attention map to 0-1, and \odot denotes element-wise product. The self-attention features $self_{s \rightarrow t}$ is produced by calculating multiplication between the features coming from F_s and the normalized attention mask from F_m .

Pixel-wise Global Attention We rely on the spatial correlation between tasks for pixel-wise global attention; see yellow blocks in Fig. 2. We follow Non-local Block [23] to obtain affinity maps between task s and t . We perform downscale and dimension reduction before calculating for smaller memory footprints and faster inference speed. The correlation maps between task t and task s can be formulated as:

$$\begin{aligned} Q &= P_Q(\downarrow_{sc} f_t), K = P_K(\downarrow_{sc} f_s), V = P_V(\downarrow_{sc} f_s) \\ Co_{s \rightarrow t} &= softmax\left(\frac{K^T Q}{\sqrt{d}}\right) \end{aligned} \quad (2)$$

where $P_*(\cdot)$ here denotes a 1×1 convolution, following a BatchNorm and ReLU function. \downarrow_{sc} the downscale operator with sc the scale factor. The spatial-correlation matrix $Co_{s \rightarrow t}$ is then obtained by applying a softmax on the matrix multiplication normalized by \sqrt{d} where d is the length of vector K . The softmax function is used to generate probabilities. Intuitively, $Co_{s \rightarrow t}$ has high values where features from s and t are highly correlated and low values otherwise.

Subsequently, Iterative Fusion Block (IFB) takes as input the obtained matrix $Co_{s \rightarrow t}$ and the vector V , and we obtain our pixel-wise global attention features by upsampling the output of IFB:

$$V_{s \rightarrow t} = \uparrow_{sc} IFB(V, Co_{s \rightarrow t}) \quad (3)$$

where \uparrow_{sc} the upscale operator with sc the scale factor. Details of IFB will be discussed in Sec. 3.4 later.

Feature Aggregations The final features $f_{s \rightarrow t}$ are built by combining the features from two attention blocks as:

$$f_{s \rightarrow t} = sel f_s + V_{s \rightarrow t} \quad (4)$$

Finally, the corresponding features of task t from multiple source tasks are aggregated as:

$$f_t^o = \sum_{i=0}^N f_{i \rightarrow t} \quad (5)$$

where $f_{t \rightarrow t} = f_t^i$. Output feature f_t^o is consequently fed into the prediction module.

3.4 Iterative Fusion Block (IFB)

Usually, after obtaining the correlation matrix $Co_{s \rightarrow t}$, it is multiplied with the matrix V to get global attention. Considering that a single matrix multiplication may not be able to fully integrate the information in the affinity map with the original features V , we introduce an iterative mechanism to fuse the original feature information with details in affinity maps profoundly and effectively. In this paper, we develop and investigate three different iteration block designs, as shown in Fig. 3. The IFB A represents a naive iteration of matrix multiplication of affinity map $Co_{s \rightarrow t}$ and features V . The IFB B introduces a residual term to avoid gradient vanishing and a learnable parameter α to adaptively adjust the attention term’s weights. The IFB C, based on the IFB B, adds a weight $1 - \alpha$ on the residual term to adaptively balance the weights of the attention term and residual term and keep the magnitude of the input and output consistent.

Iterative Fusion Block A A common way to iterate is to repeat the operation several times. We also consider this simple scheme as our basic iteration block, which can be formulated as follows:

$$V^{t+1} = CoV^t, t \geq 0 \quad (6)$$

where t is the number of iterations.

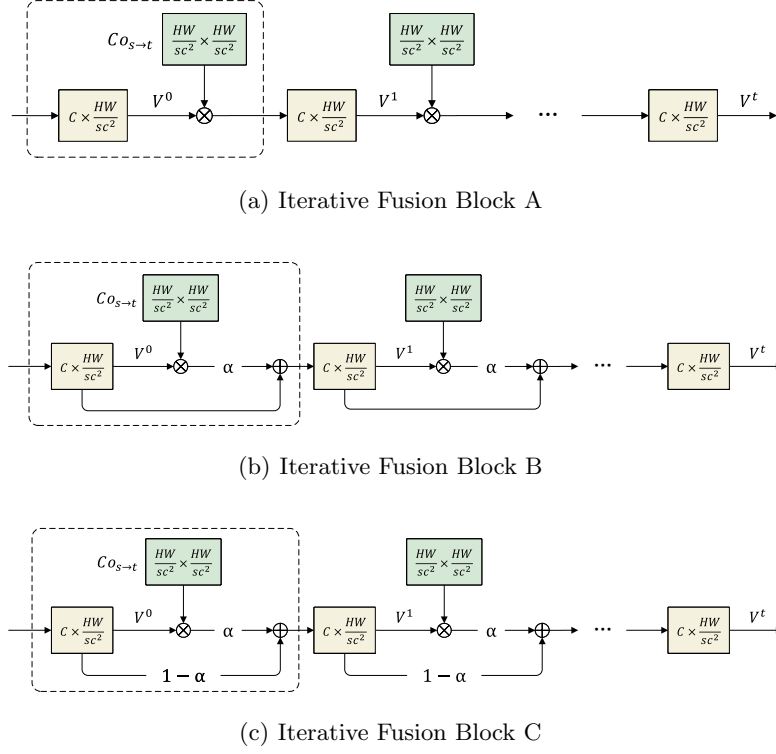


Fig. 3. Illustration of the designed different Iterative Fusion Blocks. The green blocks denote affinity map $C o_{s \rightarrow t}$ of task pairs (s, t) , and yellow blocks V indicate features from source task s . The IFB A is a naive iteration of matrix multiplication of affinity map $C o_{s \rightarrow t}$ and features V ; the IFB B introduces a residual term and a learned weight α to adjust the weight of the fused feature adaptively; the IFB C adds a weight $1 - \alpha$ to the residual term to maintain the magnitude of the feature.

However, the values in the affinity map are all probability values, which means the values are between 0 and 1. Multiplication with the affinity map, again and again, will cause the values in the feature map to gradually decrease. As the number of iterations increases, the values in the feature map approach 0, resulting in performance degradation.

Iterative Fusion Block B To overcome the abovementioned problem, we introduce a residual term to maintain the original feature information and stop the values in the feature map from reaching 0.

Here comes another question: How to combine residual term and fused term? Simply adding them together? As the number of iterations continues to increase, simply adding the two terms will lead to an increasing proportion of the original information, losing the meaning of fusion. To tackle this, a learnable parameter

α is introduced to adaptively balance the weight of residual and fused terms, and details can be seen in Fig. 3(b).

The learnable parameter α is one of the CNN model parameters, which are dynamically updated with training and supervised by the corresponding ground truth labels. There’s no need to set the value of α manually. Besides, proper initialization of α may bring about the improvement; we discuss it in Sec. 4.3.

With a learnable parameter α , IFB B can be formulated as:

$$V^{t+1} = \alpha CoV^t + V^t, t \geq 0 \quad (7)$$

where t is the number of iterations.

Iterative Fusion Block C However, IFB B still has problems. With the iteration increases, the output features V^t are significantly larger than the input feature V^0 , causing some distortion. To solve this, we add another weight $1 - \alpha$ to the residual term to keep the magnitude of the input and output consistent; see Fig. 3(c).

As mentioned above, we proposed an Iterative Fusion Block to fuse the information from affinity map $Co_{s \rightarrow t}$ and features V . Facing the problem of feature vanishing, we append a residual term. To tackle the imbalance of residual and fused terms, we introduce a learnable parameter α and $1 - \alpha$ to adaptively adjust the weights of the two terms. IFB C can be formulated as follows:

$$V^{t+1} = \alpha CoV^t + (1 - \alpha) V^t, t \geq 0 \quad (8)$$

where t is the number of iterations.

4 Experiments

To demonstrate the effectiveness of the proposed method for simultaneously solving dense prediction tasks, we conduct experiments on two publicly available benchmark datasets, NYUD-v2 and PASCAL. Sec. 4.1 describes our experimental setups, including datasets, task setups, baseline models, etc. Then we show the performance of our method on two benchmark datasets compared to state-of-the-art approaches in Sec. 4.2. Finally, we present three ablation studies in Sec. 4.3, including (i) three designs of IFB, (ii) the number of iterations, and (iii) different initial values for α on the NYUD-v2 dataset.

Table 1. Training details of our experiments.

Dataset	Model	Epoch	LR	Optimizer	Scheduler
NYUD-v2	STL	100	10e-4	SGD	Poly
	Others	100	10e-4	Adam	Poly
	STL	60	10e-2	SGD	Poly
PASCAL	Others	100	10e-4	Adam	Poly

4.1 Experimental Setups

Datasets We conduct experiments on NYUD-v2 and PASCAL datasets, which are widely used in dense predictions. The NYUD-v2 dataset contains 1449 indoor RGB depth images captured from the Microsoft Kinect and is split into a training subset and a testing subset. The former contains 795 image pairs of images and annotated images for semantic segmentation and monocular depth estimation task; the latter contains 654 pairs. The RGB and annotated images are randomly flipped horizontally and scaled for data augmentation. The PASCAL we used is a split of PASCAL-Context with dense annotations for semantic segmentation, human parts segmentation, and edge detection. It contains 4998 pairs of images, including RGB images and the corresponding ground truth labels for training and 5105 for testing. For data augmentation, the RGB and annotated images are randomly flipped horizontally, rotated, and scaled on the training set.

Task Setup Since the two datasets provide ground truth labels of different tasks, we use two sets of tasks. The first one is a two-task setup: $T = \{\text{Semantic Segmentation (SemSeg)}, \text{Monocular Depth Estimation (Depth)}\}$ on NYUD-v2 dataset. The second one is a three-task set-up: $T = \{\text{Semantic Segmentation (SemSeg)}, \text{Human Parts Segmentation (PartSeg)}, \text{Saliency Estimation (Sal)}\}$ on PASCAL. Note that, PASCAL does not provide annotations for saliency estimation task, we use the labels from [15], that distilled them from pre-trained state-of-the-art model[5]. As far as we know, there is no necessary relationship between the number of tasks and the performance of the multi-task model. Therefore, different task setups may have different results [21].

Evaluation Metrics For evaluating the performance of the semantic segmentation, human parts segmentation, and saliency estimation, pixel-level mean Intersection over Union ($mIoU$) is used. The root mean square error in meters ($RMSE$) is used for monocular depth estimation. We formulate *multi-task performance* of model m [15]: $\Delta_m = \frac{1}{N} \sum_{i=1}^N (-1)^{\gamma_i} (M_{m,i} - M_{b,i}) / M_{b,i}$ as the average of gained performance w.r.t single-task baseline b , where $\gamma_i = 1$ if a lower value means better performance for metric M_i of task i , and 0 otherwise. In our cases, $\gamma_i = 1$ only when the evaluation metric is $RMSE$.

Baseline We compare the proposed framework against a single-task learning baseline (STL), which is predicted separately by several independent networks without any cross-task interaction and a typical multi-task learning baseline (MTL) consisting of a shared encoder and several task-specific decoders. Moreover, the proposed model is compared against state-of-the-art PAD-Net [24], MTI-Net [21], DenseMTL [14], and ATRC [2]. We replace our TPIM with the distillation modules proposed by the above-mentioned works before the final prediction module. Neural Architecture Search (NAS) is needed to search optimal architecture for ATRC; we use their published search results for simplicity.

Table 2. Comparison with the state-of-the-arts on two validation sets.

(a) Comparison with the state-of-the-arts on NYUD-v2 validation set				(b) Comparison with the state-of-the-arts on PASCAL validation set.				
Model	SemSeg \uparrow	Depth \downarrow	$\Delta_m(\%) \uparrow$	Model	SemSeg \uparrow	PartSeg \uparrow	Sal \uparrow	$\Delta_m(\%) \uparrow$
STL	35.0091	0.6610	0.00	STL	59.3427	60.3365	66.892	0.00
MTL	35.0475	0.6679	-0.59	MTL	56.2943	60.1417	65.682	-2.42
PAD-Net[24]	35.8012	0.6571	1.30	PAD-Net[24]	51.9943	60.5255	65.894	-4.52
MTI-Net[21]	37.6181	0.6066	7.71	MTI-Net[21]	63.3808	<u>62.1468</u>	<u>67.413</u>	3.53
DenseMTL[14]	37.986	<u>0.6027</u>	8.53	DenseMTL[14]	63.8954	65.0074	67.519	3.79
ATRC[2]	<u>38.4576</u>	0.6098	<u>8.66</u>	ATRC[2]	<u>64.9036</u>	62.0583	66.986	<u>4.12</u>
Ours	39.1596	0.6011	10.32	Ours	65.5761	63.1563	66.944	5.08

Loss Scheme All the loss schemes are reused from [21]. Specifically, We use the L1 loss for depth estimation and the cross-entropy loss for semantic segmentation on NYUD-v2. On PASCAL, we use the balanced cross-entropy loss for saliency estimation and the cross-entropy loss for others. We do not adopt a particular loss-weighting strategy but sum the losses together with solid weights as in [22], i.e., $\mathcal{L} = \sum_{i=0}^N w_i \mathcal{L}_i$.

Training Details The proposed network structure is implemented base on Pytorch library [17] and on Nvidia GeForce RTX 3090. The backbone model HRNet18 is pre-trained with ImageNet [7]. The training configuration of all models is shown in Table 1 following [22]. A poly learning rate scheduler: $lr = lr \times (1 - \frac{epoch}{TotalEpoch})^{0.9}$ is used to adjust the learning rate.

4.2 Comparison with State-of-the-arts

Table 2(a) reports our experimental results compared to baseline models on NYUD-v2, while Table 2(b) reports our experimental results on PASCAL. On indoor densely labeled NYUD-v2, our model remarkably outperforms all baselines. Our model on the PASCAL dataset achieves the best results except for the Saliency Estimation task. A likely explanation of low performance is that the ground truth labels for saliency are distilled from pre-trained state-of-the-art model [5] as in [15]. The annotations we used are biased from the ground truth. The improvement of PAD-Net over the MTL baseline confirms the necessity to remix the task-specific features and the efficiency of the self-attention mechanism. MTI-Net retains the multi-modal distillation module of PAD-Net and adds the FPM module to fuse features at multiple scales. The remarkable improvement of MTI-Net over PAD-Net suggests that task interaction varies at different scales and emphasizes the effectiveness of multi-scale cross-task interaction. DenseMTL proposed a correlation-guided attention module, adaptively combined with a self-attention module. The improvement of DenseMTL over PAD-Net validates the importance of cross-task correlation attention. Unlike the aforementioned methods of constructing task correlation on feature space, ATRC explored task relationships on both feature and label space. The high

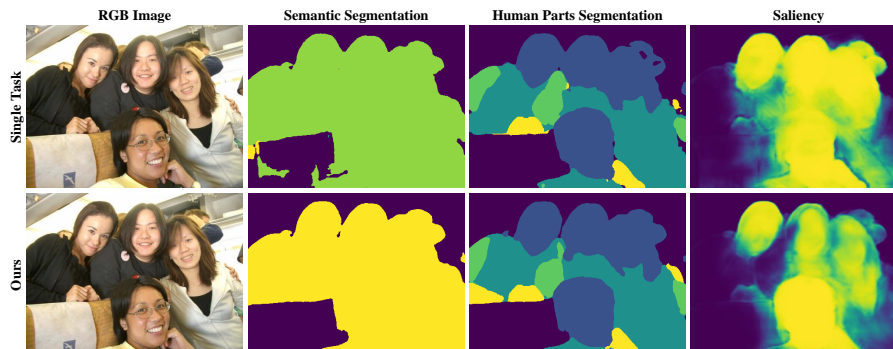


Fig. 4. Qualitative results on PASCAL dataset. We compare the predictions made by a set of single-task models against the predictions made by our model. Our model produces more precise results and smoother boundaries. And the attention distribution is more concentrated, which can be seen from Saliency.

performance of ATRC uncovers the potential of exploring task relationships on label space.

Note that our model outperforms all baseline models on both multi-task performance and single-task performance on the NYUD-v2 dataset and obtains the best on three out of four metrics on the PASCAL dataset, which not only demonstrates the effectiveness of our proposed method but indicates the benefits of jointly solving multiple related tasks and shows the excellent potential of multi-task learning.

Fig. 4 shows qualitative results on the PASCAL validation dataset. We can see the advantage of our multitask learning approach over vanilla single-task learning, where we separate objects better especially in the details.

4.3 Ablation Studies

In Table 3, we visualize the results of our ablation studies on NYUD-v2 and PASCAL to verify how IFB contributes to the multi-task improvements.

Table 3. Influence of with and without IFB on the validation set.

(a) Ablation on NYUD-v2 validation set. (b) Ablation on PASCAL validation set.
 $n_{iter} = 18$ for all three IFB. $n_{iter} = 4$.

Method	SemSeg \uparrow	Depth \downarrow	$\Delta_m(\%) \uparrow$
Ours (w/o IFB)	38.4499	0.6016	9.27
Ours (w/ IFB A)	38.1003	0.6125	7.95
Ours (w/ IFB B)	37.9603	0.5996	8.73
Ours (w/ IFB C)	39.1596	0.6011	10.32

Method	SemSeg \uparrow	PartSeg \uparrow	Sal \uparrow	$\Delta_m(\%) \uparrow$
Ours (w/o IFB)	64.0299	62.4933	67.194	3.97
Ours (w/ IFB A)	64.9804	62.5252	66.590	4.23
Ours (w/ IFB B)	65.4426	62.8790	66.791	4.78
Ours (w/ IFB C)	65.5761	63.1563	66.944	5.08

Table 4. Ablating the initialization of α in the proposed IFB on NYUD-v2 validation set. $n_{iter} = 18$.

α	SemSeg \uparrow	Depth \downarrow	$\Delta_m(\%) \uparrow$
0	<u>38.8510</u>	0.6030	<u>9.74</u>
0.1	38.5180	0.6021	9.33
0.3	39.1596	0.6011	10.32
0.5	38.2666	0.6002	9.11
0.7	38.1899	<u>0.6003</u>	9.00
0.9	38.5608	0.6047	9.20

We focus on the smaller NYUD-v2 dataset first. We alter our method using three IFBs and without IFB. As seen in Table 3(a), IFB A and IFB B lead to decreased performance, -1.32% and -0.54%, respectively. With the iteration increases, models with IFB A may degenerate into MTI-Net, where $V_{s \rightarrow t} = 0$ and CIB contains only self-attention. When $n_{iter} = 18$, The model using IFB A performs only +0.24% better than MTI-Net, proving that pixel-wise global attention plays little role in the whole model. Using IFB B tackles the problem but has a trial that compared with the input features, the output feature is prominent, which may be several times larger than the input features. It gets more prominent with the iteration increases. Instead, improvements (+1.05%) gained by IFB C confirm the rationality of our design. IFB C tackles the above problems and fully uses the cross-task information in the affinity map. A similar trend still appears in the PASCAL, but because of the smaller n_{iter} , the gap is not as obvious as in the NYUD-v2, and we will not conduct specific analysis due to space limitations.

Besides, we vary the number of iterations for better performance. 20 iterations are tested on both NYUD-v2 and PASCAL dataset, resulting in $n_{iter} = 18$ on NYUD-v2 and $n_{iter} = 4$ on PASCAL. Due to limited space, we only show the results of the ablation experiments. The experimental results show the effectiveness of further aggregate affinity information.

We believe that good initialization of α brings better performance. Therefore we perform ablation experiments on the initial value of α . We range it from 0.1 to 0.9 with $n_{iter} = 18$ on the NYUD-v2 dataset. Table 4 shows the performance with different initializations of α . Through experiments, we find that if the initial value of α is set to 0.3, the performance of the overall model is optimal, and we apply $\alpha = 0.3$ to all the above-mentioned experiments unless expressly stated. The results show that with proper initialization, the performance of our framework can be further improved.

5 Conclusions

In this paper, we proposed a novel cross-task interaction block for multi-task learning, which employs two types of attention mechanisms to build cross-task interactions to refine and distill task-specific features. One is the commonly

used self-attention, and the other is pixel-wise global attention with an iterative fusion block. Three different designs of IFB are developed to enhance cross-task interaction more effectively. Extensive experiments on two benchmark datasets demonstrate the effectiveness of our proposed framework over state-of-the-art MTL baselines.

6 Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 62176161, and the Scientific Research and Development Foundations of Shenzhen under Grant JCYJ20220818100005011

References

1. Bilen, H., Vedaldi, A.: Integrated perception with recurrent multi-task neural networks. *Advances in neural information processing systems* **29** (2016)
2. Brüggemann, D., Kanakis, M., Obukhov, A., Georgoulis, S., Van Gool, L.: Exploring relational context for multi-task dense prediction. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 15869–15878 (2021)
3. Caruana, R.: Multitask learning. *Machine learning* **28**(1), 41–75 (1997)
4. Chen, B., Guan, W., Li, P., Ikeda, N., Hirasawa, K., Lu, H.: Residual multi-task learning for facial landmark localization and expression recognition. *Pattern Recognition* **115**, 107893 (2021)
5. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: *Proceedings of the European conference on computer vision (ECCV)*. pp. 801–818 (2018)
6. Chen, Z., Badrinarayanan, V., Lee, C.Y., Rabinovich, A.: Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In: *International conference on machine learning*. pp. 794–803. PMLR (2018)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: *2009 IEEE conference on computer vision and pattern recognition*. pp. 248–255. Ieee (2009)
8. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. *International journal of computer vision* **88**(2), 303–338 (2010)
9. Gao, Y., Ma, J., Zhao, M., Liu, W., Yuille, A.L.: Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 3205–3214 (2019)
10. Kendall, A., Gal, Y., Cipolla, R.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 7482–7491 (2018)
11. Kokkinos, I.: Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 6129–6138 (2017)

12. Lee, S., Kim, J., Shin Yoon, J., Shin, S., Bailo, O., Kim, N., Lee, T.H., Seok Hong, H., Han, S.H., So Kweon, I.: Vpignet: Vanishing point guided network for lane and road marking detection and recognition. In: Proceedings of the IEEE international conference on computer vision. pp. 1947–1955 (2017)
13. Liu, S., Johns, E., Davison, A.J.: End-to-end multi-task learning with attention. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1871–1880 (2019)
14. Lopes, I., Vu, T.H., de Charette, R.: Cross-task attention mechanism for dense multi-task learning. arXiv preprint arXiv:2206.08927 (2022)
15. Maninis, K.K., Radosavovic, I., Kokkinos, I.: Attentive single-tasking of multiple tasks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1851–1860 (2019)
16. Misra, I., Shrivastava, A., Gupta, A., Hebert, M.: Cross-stitch networks for multi-task learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3994–4003 (2016)
17. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* **32** (2019)
18. Ruder, S.: An overview of multi-task learning in deep neural networks. arXiv preprint arXiv:1706.05098 (2017)
19. Ruder, S., Bingel, J., Augenstein, I., Søgaard, A.: Latent multi-task architecture learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 4822–4829 (2019)
20. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgb-d images. In: European conference on computer vision. pp. 746–760. Springer (2012)
21. Vandenhende, S., Georgoulis, S., Gool, L.V.: Mti-net: Multi-scale task interaction networks for multi-task learning. In: European Conference on Computer Vision. pp. 527–543. Springer (2020)
22. Vandenhende, S., Georgoulis, S., Van Gansbeke, W., Proesmans, M., Dai, D., Van Gool, L.: Multi-task learning for dense prediction tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence* (2021)
23. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7794–7803 (2018)
24. Xu, D., Ouyang, W., Wang, X., Sebe, N.: Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 675–684 (2018)
25. Zhang, Z., Cui, Z., Xu, C., Jie, Z., Li, X., Yang, J.: Joint task-recursive learning for semantic segmentation and depth estimation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 235–251 (2018)
26. Zhang, Z., Cui, Z., Xu, C., Yan, Y., Sebe, N., Yang, J.: Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4106–4115 (2019)
27. Zhou, L., Cui, Z., Xu, C., Zhang, Z., Wang, C., Zhang, T., Yang, J.: Pattern-structure diffusion for multi-task learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4514–4523 (2020)