DeMT: Deformable Mixer Transformer for Multi-Task Learning of Dense Prediction

Yangyang Xu [1], **Yibo Yang** [3], **Lefei Zhang** [1,2*]

[1] National Engineering Research Center for Multimedia Software, School of Computer Science, Wuhan University, China
[2] Hubei Luojia Laboratory, China [3] JD Explore Academy, China
{yangyangxu, zhanglefei}@whu.edu.cn, ibo@pku.edu.cn

Abstract

Convolution neural networks (CNNs) and Transformers have their own advantages and both have been widely used for dense prediction in multi-task learning (MTL). Most of the current studies on MTL solely rely on CNN or Transformer. In this work, we present a novel MTL model by combining both merits of deformable CNN and query-based Transformer for multi-task learning of dense prediction. Our method, named DeMT, is based on a simple and effective encoder-decoder architecture (*i.e.*, deformable mixer encoder and task-aware transformer decoder). First, the deformable mixer encoder contains two types of operators: the channel-aware mixing operator leveraged to allow communication among different channels (*i.e.*, efficient channel location mixing), and the spatial-aware deformable operator with deformable convolution applied to efficiently sample more informative spatial locations (*i.e.*, deformed features). Second, the task-aware transformer decoder consists of the task interaction block and task query block. The former is applied to capture task interaction features via self-attention. The latter leverages the deformed features and task-interacted features to generate the corresponding task-specific feature through a query-based Transformer for corresponding task predictions. Extensive experiments on two dense image prediction datasets, NYUD-v2 and PASCAL-Context, demonstrate that our model uses fewer GFLOPs and significantly outperforms current Transformer- and CNN-based competitive models on a variety of metrics. The code are available at https://github.com/yangyangxu0/DeMT.

1 Introduction

Human vision capability is powerful and can perform different tasks from one visual scene, such as classification, segmentation, recognition, etc. Therefore, multi-task learning (MTL) research is topical in computer vision. We expect to develop a powerful vision model to do multiple tasks simultaneously in different visual scenarios, and this model is expected to work efficiently. As shown in Figure 1, in this paper, we aim to develop a powerful vision model to learn multiple tasks, including semantic segmentation, human parts segmentation, depth estimation, boundary detection, saliency estimation, and normal estimation simultaneously, and this model is expected to work efficiently.

Recently, existing works (Liu, Johns, and Davison 2019; Vandenhende et al. 2020; Phillips et al. 2021; Ghiasi et al. 2021; Bruggemann et al. 2021; Xu et al. 2022b; Bhattacharjee et al. 2022) have adopted CNN and Transformer technologies to advance the MTL of dense prediction. Although CNN-based MTL models are carefully proposed to achieve promising performance on the multi-task dense prediction task, these models still suffer from the limitations of convolutional operations, *i.e.*, lacking global modeling and cross-task interaction capability. Some works (Bruggemann et al. 2021; Vandenhende et al. 2020) develop a distillation scheme to increase the expressiveness of the cross-task and global information passing via enlarging the receptive field and stacking multiple convolutional layers but still cannot build global dependency directly. For modeling global and cross-task interaction information, Transformer-based MTL models (Bhattacharjee et al. 2022; Xu et al. 2022b) utilize the efficient attention mechanism (Vaswani et al. 2017) for global modeling and task interactions. However, such a self-attention approach may fail to focus on task awareness features because the *queries*, *keys* and *values* are based on the same feature. Regular self-attention may lead to high computational costs and limit the ability to disentangle task-specific features.

We can see that the CNN-based models better capture the multiple task context in a local field but suffer from a lack of global modeling and task interaction. The Transformer-based models better focus on global information of different tasks. However, they ignore task awareness and introduce many computation costs. Therefore, a technical challenge in developing a better MTL model is how to combine the merits of CNN-based and Transformer-based MTL models.

To address the challenges, we introduce the **De**formable **M**ixer **T**ransformers (DeMT): a simple and effective method for multi-task dense prediction based on combining both merits of deformable CNN and query-based Transformer. Specifically, our DeMT consists of the deformable mixer encoder and task-aware transformer decoder. Motivated by the success of deformable convolutional networks (Zhu et al. 2019) in vision tasks, our deformable mixer encoder learns different deformed features for each task based on more efficient sampling spatial locations and channel location mixing

[*] Corresponding Author.

($i.e.$, deformed feature). It learns multiple deformed features highlighting more informative regions with respect to the different tasks. In the task-aware transformer decoder, the multiple deformed features are fused and fed into our task interaction block. We use the fused feature to generate task-interacted features via a multi-head self-attention for model task interactions. To focus on the task awareness of each individual task, we use deformed features directly as *query* tokens. We expect the set of candidate *key/value* to be from task-interacted features. Then, our task query block tasks the deformed features and task-interacted features as input and generates the task awareness features. In this way, our deformable mixer encoder selects more valuable regions as deformed features to alleviate the lack of global modeling in CNN. The task-aware transformer decoder performs the task interactions by self-attention and enhances task awareness via a query-based Transformer. This design both reduces computational costs and focuses on task awareness features. Through extensive experiments on several publicly MTL dense prediction datasets, we demonstrate that the proposed DeMT method achieves state-of-the-art results on a variety of metrics.

The contributions of this paper are as follows: 1) We propose a simple and effective DeMT method for MTL of dense prediction via combining both merits of CNN and Transformer. Most importantly, our approach not only alleviates the lack of global modeling in MTL models using CNNs but also avoids the lack of task awareness in MTL models using Transformers. 2) We introduce a deformable mixer transformer (DeMT) model which consists of the deformable mixer encoder (Section 3.3) and task-aware transformer decoder (Section 3.4). The deformable mixer encoder produces the deformed features. The task-aware transformer decoder uses the deformed features to model task interaction via a self-attention and focus on the task awareness features via a query-based transformer. 3) The extensive experiments on NYUD-v2 (Silberman et al. 2012) and PASCAL-Context (Chen et al. 2014) and visualization results show the efficacy of our model. DeMT's strong performance on MTL can demonstrate the benefits of combining the deformable CNN and query-based Transformer.

2 Related Work

2.1 Multi-Task Learning (MTL)

MTL has dramatically evolved with the development of Deep Neural Networks and Vision Transformers. MTL tasks are mainly distributed in two aspects: model structures (Bruggemann et al. 2021) and task loss weighting optimization (Liu et al. 2021a). In the vision domain, the core idea of MTL is to use a single model to predict semantic segmentation, human parts segmentation, depth, surface normal, boundary, *etc.*, which is an interesting topic. MuST (Ghiasi et al. 2021) model uses the knowledge in independent specialized teacher models to train a general model for creating general visual representations. Several recent MTL frameworks follow different technologies: (Gao et al. 2019; Vandenhende et al. 2020) is CNN-based MTL model, (Bruggemann et al. 2021) is Neural Architecture Search (NAS)-based model and (Bhattacharjee et al. 2022; Xu et al. 2022b,a) are Transformer-based models. (Vandenhende et al. 2021) states that the MTL structures in vision tasks can be summarized into two categories: encoder- and decoder-focused architectures. Some encoder-focused works (Kendall, Gal, and Cipolla 2018; Chen et al. 2018) rely on a shared encoder to learn a general visual feature. The features from the shared encoder are input into the task-specific heads to perform the dense predictions for every task. The decoder-focused works (Gao et al. 2019; Zhang et al. 2019; Bruggemann et al. 2021) use a shared backbone network to extract a shared feature for each task. Then, the designed task-specific module captures valuable information from the shared feature. However, these MTL methods primarily focus on the shared feature, which is hard to disentangle the task-specific features.

2.2 Deformable CNNs and Transformers.

Deformable CNNs. Deformable ConvNets (Dai et al. 2017; Zhu et al. 2019) harness the enriched modeling capability of CNNs (Yang et al. 2020) via deformable convolution and deformable spatial locations. Deformable ConvNets is the first to achieve competitive results in vision tasks ($e.g.$, object detection and semantic segmentation, *etc.*) using deformable convolution. Deformable transformers (Zhu et al. 2021; Xia et al. 2022) attend to a small set of crucial sampling points around a reference and capture more informative features.

Transformers. Transformer and attention (Vaswani et al. 2017) mechanism models were first employed in natural language processing (NLP) tasks with good performance. Recently, the transformer structures have also produced impressive results in computer vision and tend to replace CNN progressively. ViT (Dosovitskiy et al. 2021) is the first work to derive from the attention mechanism for computer vision tasks. More Transformer-based approaches (Carion et al. 2020; Liu et al. 2021b; Ranftl, Bochkovskiy, and Koltun 2021; Wang et al. 2021; Lan et al. 2022; Ru et al. 2022) have been introduced by improving the attention mechanism for dense prediction tasks. Recently, these works are also extended to the MTL domain to learn good representations for multiple task predictions. In contrast, we find the deformed features to focus on the valuable region for different tasks. In addition, we use the query-based transformer approach for modeling and leverage deformed features as queries in transformer calculations to enhance task-relevant features. These queries can naturally disentangle the task-specific feature from the fused feature. Our approach combines the respective advantages of CNN and Transformer, achieving state-of-the-art on NYUD-v2 and PASCAL-Context datasets.

3 The DeMT Method

3.1 Overall Architecture

We describe the overall framework of our architecture in Figure 1. DeMT is the result of a non-shared encoder-decoder procedure: First, we design a deformable mixer encoder to encode task-specific spatial features for each task. Second, the task interaction block and task query block are

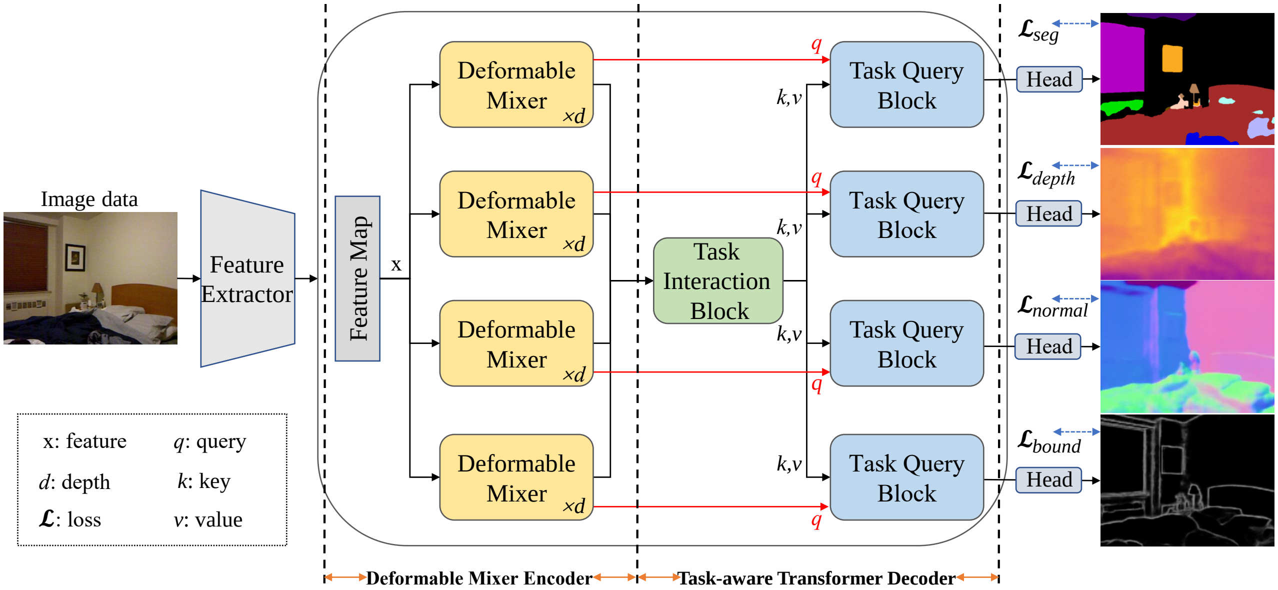

Figure 1: An overview of our model jointly handles multiple tasks with a unified encoder-decoder architecture. Our DeMT model consists of the deformable mixer encoder and task-aware transformer decoder. The depth d is the number of repetitions of the Deformable Mixer (ablation on the d in Table 3b).

proposed to model the decode the task interaction information and decode task-specific features via self-attention. In the following section, we describe our task losses.

3.2 Feature Extractor

The feature extractor is utilized to aggregate multi-scale features and manufacture a shared feature map for each task. The initial image data $X_{in} \in \mathbb{R}^{H\times W\times 3}$ (3 means image channel) is input to the backbone, which then generates four stages of image features. Then the four stage image features are up-sampled to the same resolution, and then they are concatenated along the channel dimension to obtain an image feature $X \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times C}$, where H, W, and C are the height, width, and channel of the image feature, respectively.

3.3 Deformable Mixer Encoder

The motivation. Inspired by the success of the Deformable ConvNets (Zhu et al. 2019) and Deformable DETR models (Zhu et al. 2021), we propose the deformable mixer encoder that adaptively provides more efficient receptive fields and sampling spatial locations for each task. For this purpose, the deformable mixer encoder is designed to separate the mixing of spatial-aware deformable spatial features and channel-aware location features. As shown in Figure 2 (left), the spatial-aware deformable and channel-aware mixing operators are interleaved to enable interaction of both input feature dimensions $(HW \times C)$.

Specifically, we propose a deformable mixer encoder to capture the unique receptive regions corresponding to the individual task. The deformable mixer only attends to a small set of crucial sampling points which are learnable offset. The spatial-aware deformable is capable of modeling spatial context aggregation. Then the spatial-aware deformable, channel-aware mixing, and layer normalization operators are stacked to form one deformable mixer. The effect of the depth of the deformable mixer stack on the model is shown in the Table 3b ablation experiment.

The deformable mixer encoder structure is shown in Figure 2. First, a linear layer reduces the channel dimension of the image feature $X \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times C}$ from C to a smaller dimension C'. The linear layer can be written as follows:

$$X = W \cdot \mathrm{Norm}(X), \tag{1}$$

where Norm means LayerNorm function. After the linear layer, we obtain a smaller dimension image feature map $X \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times C'}$ as the input for the downstream.

Channel-aware mixing. The channel-aware mixing allows communication between different channels. The channel-aware mixing applies the standard point-wise convolution (the convolving kernel is 1×1) to mix channel locations. It can be formulated as:

$$X_{C'} = \sum_{C'=0}^{C'-1} W_1 \cdot X_{C'} + b, \tag{2}$$

where the W_1 is the point-wise convolution weight. b is a learnable bias. Subsequently, we add GELU activation and BatchNorm as well. This operation is calculated as:

$$X_{C'} = \mathrm{BN}(\sigma(X_{C'})), \tag{3}$$

where $\sigma(\cdot)$ is the non-linearity function (GELU); BN is the BatchNorm operation.

Spatial-aware deformable. Given the input image feature $X_{i,j} \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times C'}$ from Eq.(3), the point (i, j) is the spatial location on the single channel.

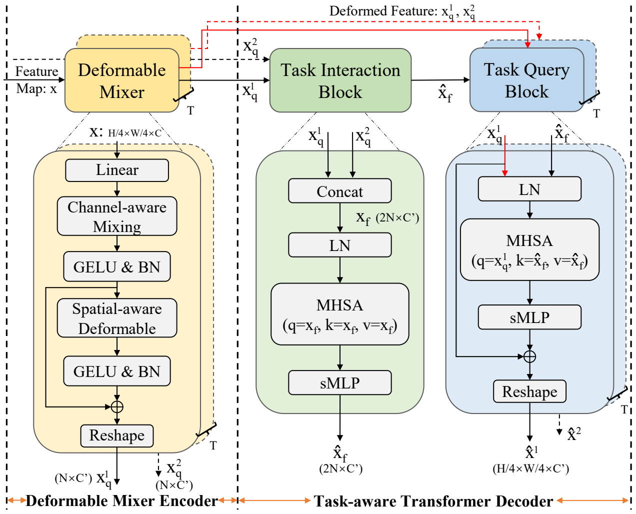

Figure 2: Illustration of our DeMT components. For simplicity, we assume there are two tasks (T=2) in this figure. Small MLP (sMLP) only consists of Linear and LayerNorm functions.

To generate the relative offsets with respect to the reference point, the image feature $X_{C'}$ is fed to the convolution operator to learn the corresponding offsets $\Delta_{(i,j)}$ for all reference points. For each location point (i,j) on the image feature X, the spatial deformable can be written as:

$$D_S(X_{i,j}) = \sum_{C'=0}^{C'-1} W_2 \cdot X((i,j) + \Delta_{(i,j)}, C'), \tag{4}$$

where the W_2 is a deformable weight. The $\Delta_{(i,j)}$ is the learnable offset. The spatial-aware deformable is followed by a GELU activation, BatchNorm, and residual connection:

$$X_q = \text{Reshape}(X_{C'} + \text{BN}(\sigma(D_S(X_{i,j})))), \tag{5}$$

where the *Reshape* is applied to flatten the feature $X_q \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times C'}$ to a sequence $\mathbb{R}^{N\times C'}$ ($N = \frac{H}{4} \times \frac{W}{4}$). When there are T tasks, the deformable mixer encoder generate a feature set $(X_q^1, X_q^2, \cdots X_q^T)$ (T means task number) (See Figure 2). These output task-specific features are learned by a deformable mixer that we refer to as *deformed features*, which we add to the input of the downstream blocks (task interaction block and task query block).

3.4 Task-aware Transformer Decoder

In the task-aware transformer decoder, we design the task interaction block and task query block (See Figure 2). It is important for MTL to consider task interactions. Thus, we propose a task interaction block to capture the task interactions at every task via an attention mechanism. Each task interaction block is composed of two parts, i.e., a multi-head self-attention module (MHSA) and a small Multi-Layer Perceptron (sMLP). The downstream task query block also consists of the MHSA and the sMLP. The difference between the task interaction block and the task query block is that their query features are fundamentally different. The feature is projected into the queries (Q), keys (K) and values (V) of dimension d_k and self attention is being computed by the Q, K and V. The self-attention operator is calculated as:

$$\text{MHSA}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V, \tag{6}$$

where $Q \in \mathbb{R}^{N\times C'}$, $K \in \mathbb{R}^{N\times C'}$ and $V \in \mathbb{R}^{N\times C'}$ are the query, key and value matrices; $\text{MHSA}(Q, K, V) \in \mathbb{R}^{N\times C'}$.

Task interaction block. As illustrated in Figure 2 (center), We first concatenate the deformed features from the deformable mixer encoder output.

$$X_f = \text{Concat}(X_q^1, X_q^2, \cdots X_q^T), \tag{7}$$

where $X_f \in \mathbb{R}^{TN\times C'}$ is the fused feature. The T means task number in $X_q^T \in \mathbb{R}^{N\times C'}$. Then, for efficient task interaction, we construct a self-attention strategy via the fused feature X_f:

$$X'_f = \text{MHSA}(Q = \text{LN}(X_f), K = \text{LN}(X_f), V = \text{LN}(X_f)), \tag{8}$$

$$\hat{X}_f = \text{sMLP}(X'_f), \tag{9}$$

where $\hat{X}_f \in \mathbb{R}^{TN\times C'}$ is the task-interacted feature. LN means LayerNorm function. sMLP consists of a linear layer and a LayerNorm.

Task query block. As illustrated in Figure 2 (right), we take the deformed feature X_q as task query and the task-interacted feature $\hat{X}_f$ as key & value to MHSA. The deformed feature is applied as a query in MHSA to decode the task awareness feature from the task-interacted feature for each task prediction. We first apply the LayerNorm in parallel to generate queries Q, keys K and values V:

$$\hat{Q} = \text{LN}(X_q), \quad \hat{K} = \text{LN}(\hat{X}_f), \quad \hat{V} = \text{LN}(\hat{X}_f), \tag{10}$$

where LN is the layer normalization. X_q and $\hat{X}_f$ are the output of deformable mixer encoder and task interaction block, respectively. Then, the task query block operation using a MHSA is calculated as:

$$\hat{X}_q = \text{MHSA}(\hat{Q}, \hat{K}, \hat{V}), \tag{11}$$

$$\hat{X} = \text{Reshape}(X_q + \text{sMLP}(\hat{X}_q)), \tag{12}$$

where the residual feature X_q comes from Eq. (5). The task awareness feature $\hat{X} \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times C'}$ is reshaped from $\mathbb{R}^{N\times C'}$ ($N = \frac{H}{4} \times \frac{W}{4}$) via *Reshape* operation.

3.5 Loss Function

For balancing the loss contribution for each task, we set the weight α_t to decide the loss contribution for the task t. A weighted sum $\mathcal{L}_{total}$ of task-specific losses:

$$\mathcal{L}_{total} = \sum_{t=1}^{T} \alpha_t \mathcal{L}_t, \tag{13}$$

where the $\mathcal{L}_t$ is a loss function for task t. For fair comparisons, we use α_t and $\mathcal{L}_t$ consistent with ATRC (Bruggemann et al. 2021) and MQTransformer (Xu et al. 2022b).

Table 1: Comparison of the MTL models with state-of-the-art on NYUD-v2 dataset. The notation '↓': lower is better. The notation '↑': higher is better. Δ_m denotes average per-task performance drop. "Params" denotes parameters.

Model	Backbone	Params (M)	GFLOPs (G)	SemSeg (IoU)↑	Depth (rmse)↓	Normal (mErr)↓	Bound (odsF)↑	$\Delta_m[\%]\uparrow$
single task baseline	HRNet18	16.09	40.93	38.02	0.6104	20.94	76.22	0.00
multi-task baseline	HRNet18	4.52	17.59	36.35	0.6284	21.02	76.36	-1.89
Cross-Stitch(Misra et al. 2016)	HRNet18	4.52	17.59	36.34	0.6290	20.88	76.38	-1.75
Pad-Net(Xu et al. 2018)	HRNet18	5.02	25.18	36.70	0.6264	20.85	76.50	-1.33
PAP(Zhang et al. 2019)	HRNet18	4.54	53.04	36.72	0.6178	20.82	76.42	-0.95
PSD(Ling et al. 2020)	HRNet18	4.71	21.10	36.69	0.6246	20.87	76.42	-1.30
NDDR-CNN(Gao et al. 2019)	HRNet18	4.59	18.68	36.72	0.6288	20.89	76.32	-1.51
MTI-Net(Vandenhende et al. 2020)	HRNet18	12.56	19.14	36.61	0.6270	20.85	76.38	-1.44
ATRC(Bruggemann et al. 2021)	HRNet18	5.06	25.76	38.90	0.6010	20.48	76.34	1.56
DeMT (Ours)	HRNet18	4.76	22.07	39.18	0.5922	20.21	76.40	2.37
single task baseline	Swin-T	115.08	161.25	38.02	0.6104	20.94	76.22	0.00
multi-task baseline	Swin-T	32.50	96.29	38.78	0.6312	21.05	75.60	-3.74
MQTransformer(Xu et al. 2022b)	Swin-T	35.35	106.02	43.61	0.5979	20.05	76.20	0.31
DeMT (Ours)	Swin-T	32.07	100.70	46.36	0.5871	20.65	76.90	3.36
single task baseline	Swin-S	200.33	242.63	48.92	0.5804	20.94	77.20	0.00
multi-task baseline	Swin-S	53.82	116.63	47.90	0.6053	21.17	76.90	-1.96
MQTransformer(Xu et al. 2022b)	Swin-S	56.67	126.37	49.18	0.5785	20.81	77.00	1.59
DeMT (Ours)	Swin-S	53.03	121.05	51.50	0.5474	20.02	78.10	4.12

4 Experiment

In this section, we conduct extensive experiments on two widely-used dense prediction datasets to evaluate the performance of our method on different metrics. We also show the visualization results on different datasets.

4.1 Experimental Setup

Implementation. All the leveraged backbones generate four scales (1/4, 1/8, 1/16, 1/32) features to perform multi-scale aggregation in our feature extractor (Section 3.2). We train our model with SGD setting the learning rate to 10^{-3} and weight decay to 5×10^{-4}. The whole experiments are performed with pre-trained models on ImageNet. All our experiments are performed on the Pytorch platform with eight A100 SXM4 40GB GPUs.

Datasets. We conduct experiments on two publicly accessible datasets, NYUD-v2 (Silberman et al. 2012) and PASCAL-Context (Chen et al. 2014). NYUD-V2 is comprised of pairs of RGB and Depth frames that 795 images are used for training and 654 images for testing. NYUD-V2 usually is mainly adopted for semantic segmentation ('SemSeg'), depth estimation ('Depth'), surface normal estimation ('Normal'), and boundary detection ('Bound') tasks by providing dense labels for every image. PASCAL-Context training and validation contain 10103 images, while testing contains 9637 images. PASCAL-Context usually is adopted for semantic segmentation ('SemSeg'), human parts segmentation ('PartSeg'), saliency estimation ('Sal'), surface normal estimation ('Normal'), and boundary detection ('Bound') tasks by providing annotations for the whole scene.

Metrics. We adopt five evaluation metrics to compare our model with other prior multi-task models: mean Intersection over Union (mIoU), root mean square error (rmse), mean Error (mErr), optimal dataset scale F-measure (odsF), and maximum F-measure (maxF). The average per-task performance drop (Δ_m) is used to quantify multi-task performance. $\Delta_m = \frac{1}{T}\sum_{i=1}^{T}(F_{m,i} - F_{s,i})/F_{s,i} \times 100\%$, where m, s and T mean multi-task model, single task baseline and task numbers. Δ_m: the higher is the better.

Backbones. We test our method using several CNN and Vision Transformer backbones: HRNetV2-W18-small (HRNet18), HRNetV2-W48 (HRNet48) (Sun et al. 2019), Swin-Tiny (Swin-T), Swin-Small (Swin-S) and Swin-Base (Swin-B) (Liu et al. 2021b).

4.2 Comparison with the state-of-the-art

We compare our model with CNN-based and Transformer-based models to show the advantages of our method.

NYUD-v2. The Comparisons with state-of-the-art models on the NYUD-v2 dataset are shown in Table 1. We first report results comparison with three different backbones: HRNet18, Swin-T, and Swin-S. We demonstrate simultaneous performance improvements over prior work in having smaller parameters, a smaller number of GFLOPs, and better semantic segmentation, depth estimation, surface normal and boundary detection accuracies. For example, a performance comparison between MQTransformer and DeMT proves the effectiveness of our framework. Besides this, DeMT also consistently outperforms previous state-of-the-art Transformer-based models, such as ATRC (Bruggemann et al. 2021) and MQTransformer (Xu et al. 2022b). In addition, we also observe that using a transformer as a backbone model is more promising compared to CNN as the backbone. Because Transformer-based and CNN-based models use similar GFLOPs, the former shows higher accuracy in all metrics. Our DeMT obtains 46.36 SemSeg accuracy, which is 6.3% higher than that of MQTransformer with the same Swin-T backbone and slightly lower FLOPs (100.7G vs. 106.02G). MuIT (Bhattacharjee et al. 2022) reports a 13.3% and 8.54% increase in relative performance for semantic

Table 2: Comparison of the MTL models with state-of-the-art on PASCAL-Context dataset. The notation '↓': lower is better. The notation '↑': higher is better. Δ_m denotes average per-task performance drop (the higher is the better).

Model	Backbone	SemSeg (IoU)↑	PartSeg (IoU)↑	Sal (maxF)↑	Normal (mErr)↓	Bound (odsF)↑	Δ_m[%]↑
single task baseline	HRNet18	62.23	61.66	85.08	13.69	73.06	0.00
multi-task baseline	HRNet18	51.48	57.23	83.43	14.10	69.76	-6.77
PAD-Net (Xu et al. 2018)	HRNet18	53.60	59.60	65.80	15.3	72.50	-4.41
ATRC (Bruggemann et al. 2021)	HRNet18	57.89	57.33	83.77	13.99	69.74	-4.45
MQTransformer(Xu et al. 2022b)	HRNet18	58.91	57.43	83.78	14.17	69.80	-4.20
DeMT (Ours)	HRNet18	59.23	57.93	83.93	14.02	69.80	-3.79
single task baseline	Swin-T	67.81	56.32	82.18	14.81	70.90	0.00
multi-task baseline	Swin-T	64.74	53.25	76.88	15.86	69.00	-3.23
MQTransformer(Xu et al. 2022b)	Swin-T	68.24	57.05	83.40	14.56	71.10	1.07
DeMT (Ours)	Swin-T	69.71	57.18	82.63	14.56	71.20	1.75
single task baseline	Swin-S	70.83	59.71	82.64	15.13	71.20	0.00
multi-task baseline	Swin-S	68.10	56.20	80.64	16.09	70.20	-3.97
MQTransformer(Xu et al. 2022b)	Swin-S	71.25	60.11	84.05	14.74	71.80	1.27
DeMT (Ours)	Swin-S	72.01	58.96	83.20	14.57	72.10	1.36
single task baseline	Swin-B	74.91	62.13	82.35	14.83	73.30	0.00
multi-task baseline	Swin-B	73.83	60.59	80.75	16.35	71.10	-3.81
DeMT (Ours)	Swin-B	75.33	63.11	83.42	14.54	73.20	1.04

segmentation and depth tasks. While we have a 14.74% and 9.43% increase. The comparison results show our model also achieves good performance, evaluating the flexibility of our model. By comparison, our DeMT achieves new records on the NYUD-v2, which are remarkably superior to previous CNNs and Transformers models in terms of all metrics.

PASCAL-Context. We also evaluate the proposed DeMT on PASCAL-Context with three backbones: HRNet18, Swin-T, Swin-S, and Swin-B. Table 2 shows the comparison results. Our model obtains significantly better results when compared with the baseline and other models. For example, DeMT improves MQTransformer (Xu et al. 2022b) with the same Swin-T backbone by 1.47 point in SemSeg. Our DeMT achieves the best performance among models on several metrics and can reach a high performance of 75.33 in the SemSeg task.

4.3 Ablation Studies

We ablate DeMT to understand the contribution of each component and setting using Swin-T on NYUD-v2 dataset.

Ablation on modules. The DeMT model consists of three components: deformable mixer, task interaction, and task query blocks. As shown in Table 3a, we demonstrate the advantages of the deformable mixer, task interaction, and task query blocks. We observe that task interaction block has more effect on the performance, and it is essential to interact the whole task features for task interaction information. This indicates that task interaction and task query blocks are essential to the task-aware transformer decoder. From the Figure 4 and Table 3a it can be observed that different components are playing a beneficial role.

Ablation on the depths d. As shown in Figure 2, the depth d is the number of repetitions of the deformable mixer. We add the d to analyze the effect of the depth of the deformable mixer on the DeMT model. In Table 3b, We vary the number of used deformable mixer depth ($e.g.$, 1, 2, 4, 8) and compare their performances. Comparing the first to last row in Table 3b, we observe the best performance when the depth is set to 4. However, as increasing the depth, the parameters and GFLOPs also become more extensive. Practically, we choose a depth d = 1 for all models in this paper.

Ablation on scales. We explore the influence of using different scale features. The backbone outputs four-scale (1/4, 1/8, 1/16, 1/32) features. Table 3c shows the influence of using a different number of scales. Note that the model performance increases obviously with the increasing number of scales. Our method can capture valuable semantic information for multiple tasks. Practically, we choose four-scale features for all models in this paper.

Ablation on backbones. Table 3d shows the results using the different backbones. To deeper explore the capacity of the our DeMT, we employ extensive backbones to conduct the ablation experiment. It is worth noting that our DeMT leads to the best performance on all metrics when using Swin-B on NYUD-v2. In addition, we also observe the inspiring fact that using a larger transformer backbone can easily reach top-tier performance. The different backbones are compared to demonstrate the generalization of our method.

4.4 Visualization

To deeper understand our DeMT model, we visualize the multiple task predictions. We show the qualitative results in different dimensions. For visual analysis (see Figure 3 and Figure 4), we employ a trained model with Swin-T. Figure 3 shows the capability of DeMT with Swin-T backbone to perform dense predictions with strong expressive power and successfully capture the task-specific features. As illustrated in Figure 3 (last two rows), the second and third columns focus mainly on specific semantics such as human, animal, and other objects. Figure 4 showcases the impact of our approach using different components: while only the deformable mixer encoder fails to visualize some objects,

Table 3: Ablation studies and analysis on NYUD-v2 dataset using a Swin-T backbone. Deformable mixer (DM), task interaction (TI) block, and task query (TQ) block are the parts of our model. HR48 denotes HRNet48. The notation '↓': lower is better. The notation '↑': higher is better. The **w/** indicates **"with"**.

(a) Ablation on components

Model	SemSeg (IoU)↑	Depth (rmse)↓	Normal (mErr)↓	Bound (odsF)↑
baseline	38.78	0.6312	21.05	75.6
w/ DM	42.40	0.6069	20.83	76.2
w/ DM+TI	44.44	0.5969	20.75	76.4
w/ DM+TI+TQ	46.36	0.5871	20.65	76.9

(b) Ablation on the depths (d)

d	SemSeg (IoU)↑	Depth (rmse)↓	Normal (mErr)↓	Bound (odsF)↑
1	46.36	0.5871	20.65	76.9
2	46.90	0.5622	20.05	77.0
4	47.71	0.5613	19.90	77.1
8	47.16	0.5518	19.87	77.1

(c) Ablation on scales

Scale	SemSeg (IoU)↑	Depth (rmse)↓	Normal (mErr)↓	Bound (odsF)↑
1/4	7.51	1.1961	33.26	66.1
1/4, 1/8	12.85	1.0433	27.66	70.4
1/4, 1/8, 1/16	40.32	0.6966	21.44	76.3
1/4, 1/8, 1/16, 1/32	46.36	0.5871	20.65	76.9

(d) Ablation on backbones

Backbone	SemSeg (IoU)↑	Depth (rmse)↓	Normal (mErr)↓	Bound (odsF)↑
HR48 baseline	41.96	0.5543	20.36	77.6
HR48 **w/** ours	43.84	0.5517	19.88	77.7
Swin-B baseline	51.44	0.5813	20.44	77.0
Swin-B **w/** ours	54.34	0.5209	19.21	78.5

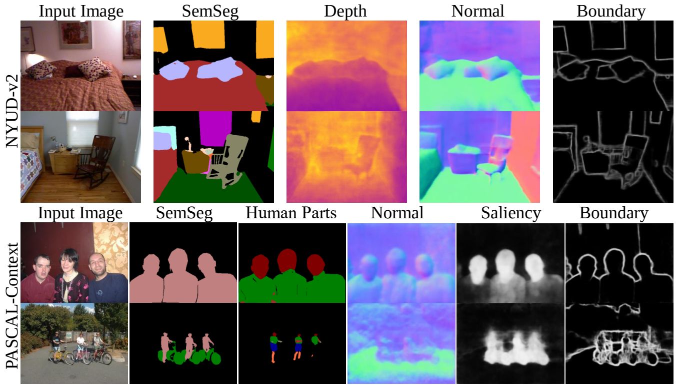

Figure 3: The first two rows of the visualization illustrate two examples from the NYUD-v2. The last two rows of the visualization illustrate two examples from the PASCAL-Context.

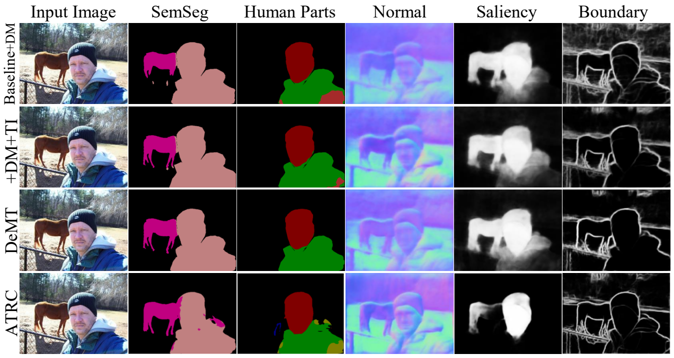

Figure 4: Qualitative analysis of the components on PASCAL-Context. Visualizations show the components in Table 3a. The last row shows the ATRC model visualization results as a comparison.

the third row shows DeMT's improvements to multiple task predictions. Note that we not only report these results for qualitative understanding of the model but also evaluate it quantitatively in Table 3a. We compared the prediction results of the DeMT model with the ATRC (Figure 4 last row), and our results are significantly better than ATRC, especially on SemSeg and Human Parts tasks. Our DeMT model produces higher-quality predictions than both the Swin baseline and the existing CNN-based MTL model.

5 Conclusion

In this work, we introduce DeMT, a simple and effective method that leverages the combination of both merits of deformable CNN and query-based Transformer for multi-task learning of dense prediction. Significantly, the deformed feature produced by the deformable mixer encoder is leveraged as a task query in the task-aware transformer decoder to disentangle task-specific features. Extensive experiments on dense prediction datasets ($i, e.$, NYUD-v2 and PASCAL-Context) validate the effectiveness of our DeMT model.

Limitations and future work. This work only uses a naive operation to aggregate multi-scale features and could be further improved in two aspects: considering using the FPN or FPN variant to aggregate multi-scale features and how to design flexible attention to learn more valuable information.

Acknowledgements

This work was done when Yangyang Xu was a research intern at JD Explore Academy. This work was supported by the National Natural Science Foundation of China under Grants 62122060, 62076188, and the Special Fund of Hubei Luojia Laboratory under Grant 220100014.

References

Bhattacharjee, D.; Zhang, T.; Süsstrunk, S.; and Salzmann, M. 2022. MulT: An End-to-End Multitask Learning Transformer. In *CVPR*, 12031–12041.

Bruggemann, D.; Kanakis, M.; Obukhov, A.; Georgoulis, S.; and Gool., L. V. 2021. Exploring relational context for multi-task dense prediction. In *ICCV*, 15869–15878.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *ECCV*, 213–229.

Chen, X.; Mottaghi, R.; Liu, X.; Fidler, S.; Urtasun, R.; and Yuille, A. 2014. Detect what you can: Detecting and representing objects using holistic models and body parts. In *CVPR*, 1971–1978.

Chen, Z.; Badrinarayanan, V.; Lee, C.-Y.; and Rabinovich, A. 2018. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *ICML*, 794–803.

Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; and Wei, Y. 2017. Deformable convolutional networks. In *ICCV*, 764–773.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*.

Gao, Y.; Ma, J.; Zhao, M.; Liu, W.; and Yuille, A. L. 2019. Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction. In *CVPR*, 3205–3214.

Ghiasi, G.; Zoph, B.; Cubuk, E. D.; Le, Q. V.; and Lin, T.-Y. 2021. Multi-Task Self-Training for Learning General Representations. In *ICCV*, 8856–8865.

Kendall, A.; Gal, Y.; and Cipolla, R. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, 7482–7491.

Lan, M.; Zhang, J.; He, F.; and Zhang, L. 2022. Siamese Network with Interactive Transformer for Video Object Segmentation. In *AAAI*, 1228–1236.

Ling, Z.; Zhen, C.; Chunyan, X.; Zhenyu, Z.; Chaoqun, W.; Tong, Z.; and Jian, Y. 2020. Pattern-Structure Diffusion for Multi-Task Learning. In *CVPR*, 4514–4523.

Liu, L.; Li, Y.; Kuang, Z.; Xue, J.; Chen, Y.; Yang, W.; Liao, Q.; and Zhang, W. 2021a. Towards impartial multi-task learning. In *ICLR*.

Liu, S.; Johns, E.; and Davison, A. J. 2019. End-to-end multi-task learning with attention. In *CVPR*, 1871–1880.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021b. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *ICCV*, 10012–10022.

Misra, I.; Shrivastava, A.; Gupta, A.; and Hebert, M. 2016. Cross-stitch networks for multi-task learning. In *CVPR*, 3994–4003.

Phillips, J.; Martinez, J.; Bârsan, I. A.; Casas, S.; Sadat, A.; and Urtasun, R. 2021. Deep multi-task learning for joint localization, perception, and prediction. In *CVPR*, 4679–4689.

Ranftl, R.; Bochkovskiy, A.; and Koltun, V. 2021. Vision transformers for dense prediction. In *CVPR*, 12179–12188.

Ru, L.; Zhan, Y.; Yu, B.; and Du, B. 2022. Learning Affinity from Attention: End-to-End Weakly-Supervised Semantic Segmentation with Transformers. In *CVPR*, 16846–16855.

Silberman, N.; Hoiem, D.; Kohli, P.; and Fergus, R. 2012. Indoor segmentation and support inference from rgbd images. In *ECCV*, 746–760.

Sun, K.; Xiao, B.; Liu, D.; and Wang, J. 2019. Deep High-Resolution Representation Learning for Human Pose Estimation. In *CVPR*, 5693–5703.

Vandenhende, S.; Georgoulis, S.; Gansbeke, W. V.; Proesmans, M.; Dai, D.; and Gool, L. V. 2021. Multi-Task Learning for Dense Prediction Tasks: A Survey. *IEEE TPAMI*.

Vandenhende, S.; Georgoulis, S.; Van Gool, L.; and Van Gool, L. 2020. Mti-net: Multi-scale task interaction networks for multi-task learning. In *ECCV*, 527–543.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *NeurIPS*.

Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; and Shao, L. 2021. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction Without Convolutions. In *ICCV*, 568–578.

Xia, Z.; Pan, X.; Song, S.; Li, L. E.; and Huang, G. 2022. Vision transformer with deformable attention. In *CVPR*, 4794–4803.

Xu, D.; Ouyang, W.; Wang, X.; and Sebe, N. 2018. Padnet: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *CVPR*, 675–684.

Xu, X.; Zhao, H.; Vineet, V.; Lim, S.-N.; and Torralba, A. 2022a. MTFormer: Multi-task Learning via Transformer and Cross-Task Reasoning. In *ECCV*, 304–321.

Xu, Y.; Li, X.; Yuan, H.; Yang, Y.; Zhang, J.; Tong, Y.; Zhang, L.; and Tao, D. 2022b. Multi-Task Learning with Multi-query Transformer for Dense Prediction. *arXiv preprint arXiv:2205.14354*.

Yang, Y.; Li, H.; Li, X.; Zhao, Q.; Wu, J.; and Lin, Z. 2020. Sognet: Scene overlap graph network for panoptic segmentation. In *AAAI*.

Zhang, Z.; Cui, Z.; Xu, C.; Yan, Y.; Sebe, N.; and Yang, J. 2019. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *CVPR*, 4106–4115.

Zhu, X.; Hu, H.; Lin, S.; and Dai, J. 2019. Deformable convnets v2: More deformable, better results. In *CVPR*, 9308–9316.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2021. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In *ICLR*.