SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

January 2026, Vol. 69, Iss. 1, 112108:1–112108:15
https://doi.org/10.1007/s11432-023-4648-7

Task prior attention network for multi-task learning of dense prediction

Yangyang XU[1], Yibo YANG[2], Lefei ZHANG[1*] & Bo DU[1]

[1]*National Engineering Research Center for Multimedia Software, Hubei Key Laboratory of Multimedia and Network Communication Engineering, School of Computer Science, Wuhan University, Wuhan 430072, China*
[2]*Visual Computing Center, King Abdullah University of Science and Technology, Thuwal 23955-6900, Saudi Arabia*

Received 17 August 2023/Revised 16 August 2025/Accepted 18 October 2025/Published online 4 January 2026

Abstract Transformer-based methods have been popular for a variety of visual perception tasks due to their better global modeling via attention. However, a plain transformer-based architecture is known for lacking inductive biases, which will impede the performance in multi-task learning (MTL) of dense prediction due to the incapability of capturing task-relevant prior information. To this end, we propose the task prior attention network (TPANet), which introduces task-relevant prior information into the whole architecture. Our TPANet consists of three tailored modules: task prior extractor, adaptive task mixing, and cross attention modules. First, the proposed task prior extractor is applied for introducing task-relevant prior information with inductive biases via convolution for each task, adapting them to the downstream module simultaneously. Second, for task interaction efficiency, our method relies on the adaptive task mixing equipped with spatial and channel mixing to capture the task interaction. Third, the proposed cross attention module is leveraged to query task-specific feature maps with task-relevant prior information via query-based attention. Our method allows compatibility with different backbones. TPANet (with Swin-L) performance surpasses the previous state-of-the-art by a large margin of +4.6 mIoU on NYUD-v2 and +0.8 mIoU on PASCAL-Context dataset, demonstrating the potential of our method as a robust MTL model.

Keywords scene understanding, multi-task learning, dense prediction, vision transformer, task prior

Citation Xu Y Y, Yang Y B, Zhang L F, et al. Task prior attention network for multi-task learning of dense prediction. Sci China Inf Sci, 2026, 69(1): 112108, https://doi.org/10.1007/s11432-023-4648-7

1 Introduction

Humans use all of their visual senses to accomplish different vision tasks in everyday activities. While in practical scenarios, many AI applications can be designed as multi-task systems to conduct multiple vision tasks simultaneously. Thus, multi-task learning (MTL) [1] is an integral part of the computer vision domain. The potential benefit of the multi-task model compared to the single-task model is an efficient prediction with fewer parameters and less computational cost. Such success and good properties of MTL frameworks have inspired many following studies that apply them in various computer vision tasks.

Convolutional neural networks (CNNs) [2] achieve great success in domains such as videos, images and text. The CNN-based MTL methods improve the domain-specific information for multiple tasks and also enjoy great improvement in dense prediction such as [3–7]. However, these CNN-based MTL methods tend to focus only on the local visual information, neglecting the global information. Recently, the transformer-based methods [8–10] show remarkable success in a wide range of computer vision fields. Therefore, recent advances in MTL of dense prediction mainly leverage transformers for further enhancing the MTL performance via the self-attention mechanism. The transformer-based MTL methods [11–14] capture the long-range dependency and global relationships of all tasks by stacking self-attention blocks. The typical transformer-based MTL models, MulT [11] and MTFormer [15], develop a self-task attention framework via multi-head self-attention to learn effective feature maps for multiple task predictions. Adopting Swin Transformer [10] as the backbone to generate multi-scale features, MulT [11] designs a decoder via a shared attention mechanism for the respective tasks and further improves the performance of each vision task.

However, a well-known drawback of using a plain transformer for vision tasks is that inductive biases will be lacking due to the pure-attention architecture [16,17]. In MTL, inductive biases are particularly important because they can bring task-relevant prior information, which facilitates the extraction of rich task-dependent local features.

* Corresponding author (email: zhanglefei@whu.edu.cn)

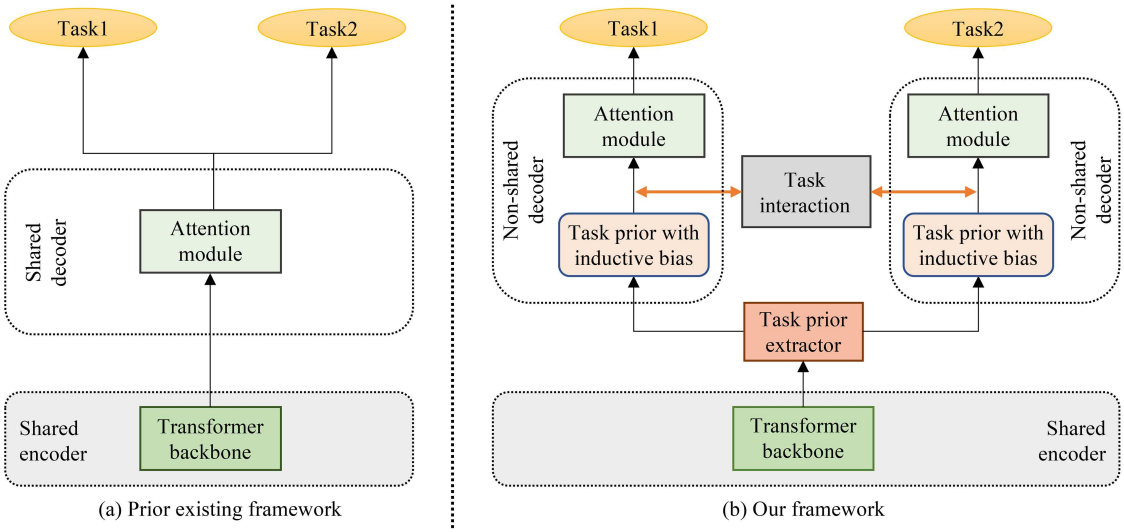

Figure 1 (Color online) Previous transformer-based MTL framework vs. our framework. (a) Previous transformer-based MTL framework (e.g., MTFormer) is designed by stacking plain self-attention. (b) We propose a novel method to boost MTL performance by introducing task-relevant prior information into self-attention. Compared to the previous method, our method designs a non-shared decoder for each task and thus could provide task-relevant prior information.

In this paper, our aim is to develop a method to introduce the task-relevant prior information with inductive bias into the plain transformer-based MTL architecture to boost the task performance for MTL of dense prediction.

We illustrate the differences between the previous framework and our framework in Figures 1(a) and (b). We point out two crucial differences. First, we develop a simple yet efficient task prior extractor module to produce task-relevant prior information with rich inductive biases for every task in Figure 1(b). Then, the task-relevant prior information is leveraged in transformer via self-attention. Second, as shown in Figure 1(b), we design a non-shared decoder for each task. To connect different task decoders, we design an adaptive task mixing module to interact adaptively among different tasks. The whole architecture is dubbed as TPANet due to the task prior attention that learns to solve the lack of task-relevant prior information for MTL of dense prediction. Specifically, we design three made-to-order modules for TPANet, including task prior extractor, adaptive task mixing, and cross attention. Task prior extractor is proposed to focus on producing task-relevant prior information with inductive bias into transformer architecture for each individual task. Task-relevant prior information with the introduced inductive biases can be adopted to promote local visual information for individual tasks. Adaptive task mixing consists of spatial and channel mixing. Adaptive task mixing is employed to learn adaptive task interactions for all tasks. The other core module is cross attention, which is adopted to produce task-specific feature maps for task prediction and further enhance performance. The proposed TPANet model shows a large superiority to the existing models (shown in Figure 2). The implementation of our method is available at https://github.com/yangyangxu0/TPANet.

The contributions of this work are three-fold.

(1) We propose a novel MTL method, named TPANet, which is effective, efficient and robust by introducing task-relevant prior information into transformer-based architecture to facilitate task-dependent local information for MTL of dense prediction.

(2) We design the task prior extractor module to produce task-relevant prior information. Adaptive task mixing is adopted to perform task interactions. Cross attention is proposed to incorporate the task-relevant prior information into the task-specific features via a query-based self-attention.

(3) We evaluate the TPANet on two challenging benchmarks, including NYUD-v2 [18] and Pascal-Context [19]. Extensive experiments demonstrate that TPANet achieves state-of-the-art results in a variety of metrics. We also perform ablations to investigate how it benefits from different modules.

2 Related work

2.1 Multi-task learning of dense prediction

The MTL method for dense prediction is proposed to train a single deep neural network to simultaneously perform semantic segmentation, depth estimation and object detection tasks. The MTL approaches [7, 13] can precisely

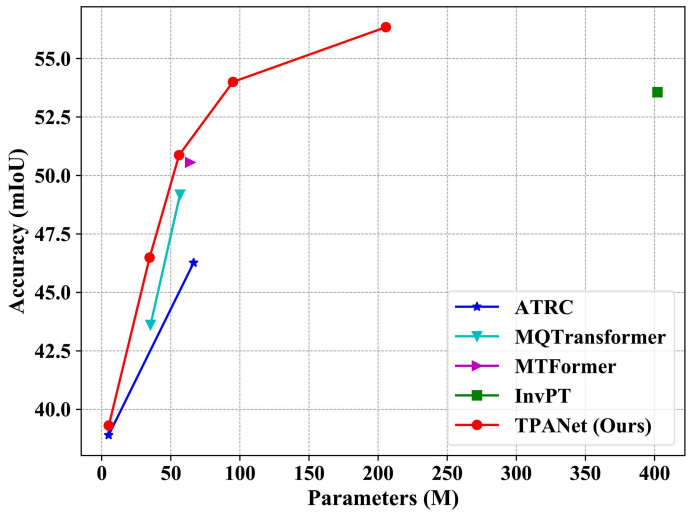

Figure 2 (Color online) Performance comparison between the proposed TPANet and existing MTL models. Segmentation accuracies on NYUD-v2.

capture relationships among different types of data and then are naturally well-suited for dealing with multiple visual tasks simultaneously in dense prediction. The potential benefits of the multi-task model compared to the single-task model are efficient prediction, fewer parameters and less computational cost. The MTL approaches [20–24] directly use the shared representation to perform all dense predictions simultaneously. However, these methods fail to conduct the task interactions and thus fail to capture complementary information among tasks. Follow-up papers have improved how to perform the task interaction in MTL of dense prediction. Refs. [13–15, 25–30] aimed to use the interaction information between tasks to promote performance. The proposed [30] method develops an MTL model, which is used for mining and leveraging the latent interaction cues by leveraging the powerful transformer. More recently, a transformer-based method in [13] proposes cross-task reasoning via a task-relevant query self-attention [31] for boosting the MTL. Similarly, MTFormer [15] presents that information from different task domains can benefit each other, and they conduct cross-task reasoning via shared self-attention among the tasks. Although the transformer-based frameworks have achieved the best performance in the multiple computer vision domains compared to CNN-based frameworks, existing transformer-based MTL frameworks employ stacked self-attention while have not explored the effectiveness of self-attention with inductive biases in the MTL domain.

2.2 CNNs and transformers

The inductive biases are hard-coded into the architecture of CNNs [2] in the form of strong constraints on the locality and weight sharing [32]. Ref. [33] designed a prior learning module to learn the pixel's difficulty prior to guild adaptive segmentation. Vision transformer [8] is the first method that applies plain self-attention to vision tasks and achieves better performance. Then, the transformer-based methods are applied to multiple vision tasks, including classification [8, 10, 17], object detection [9, 34], and semantic segmentation [35–37]. To jointly model global and local information, the methods [38–46] employ the parallel individual convolution and transformer branches, while inductive biases from convolutions are introduced into transformers [47]. For example, Mobile-Former [39] leverages the advantages of convolution at local processing and transformer at global interaction. Specifically, the transformer in Mobile-Former employs fewer patch tokens that are randomly initialized to learn global information. While 48vitaev2 [48], 49MGSNet [49], and 50SOT-Net [50] adopt multi-stage guidance for task-specific representation learning, our method emphasizes inductive bias injection via task-prior extractors and attention. Besides, whether inductive bias can still help transformer-based MTL models achieve better performance remains unexplored. This paper introduces such an inductive bias to the transformer-based MTL model by utilizing multiple convolutions in the task prior extractor to encode task-relevant prior information with the convolutional inductive bias into the task-specific feature. Experimental results confirm that introducing task-relevant prior information with inductive biases can reach higher performance in MTL of dense prediction.

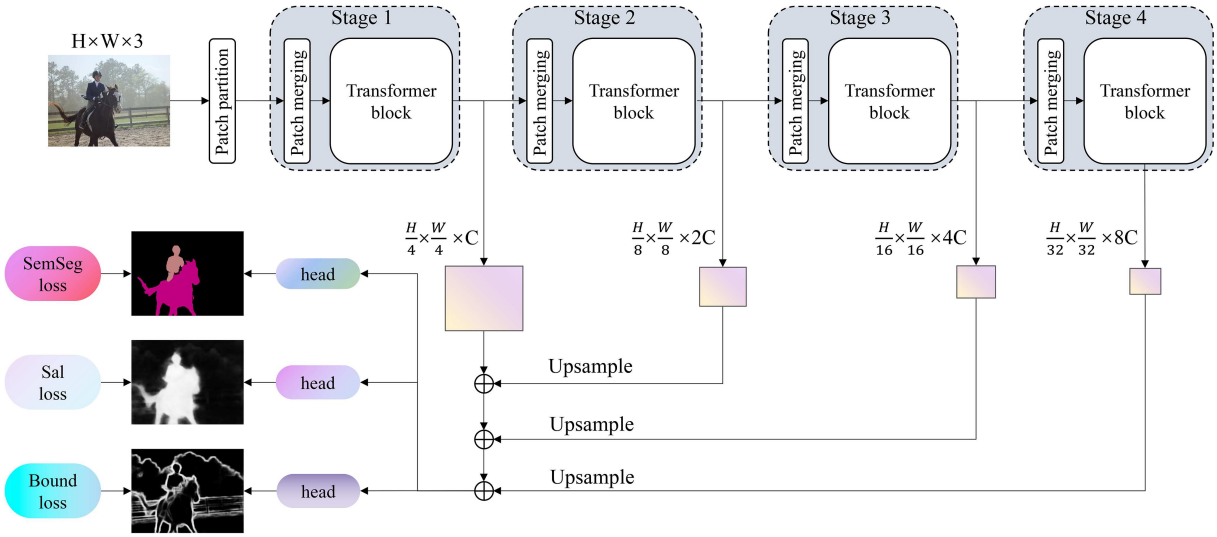

Figure 3 (Color online) Illustration of the multi-task baseline with Swin Transformer that performs dense predictions. In our framework, the backbone can support CNNs-based and transformer-based models. The original input image feature is $H \times W \times 3$, where the H, W, and 3 are the height, width, and image channel, respectively. C denotes the channel. The multi-task model performs semantic segmentation ('SemSeg'), saliency estimation ('Sal') and boundary detection ('Bound') tasks.

3 Approach

3.1 Overall architecture

As shown in Figure 3, we describe the multi-task baseline model. The framework of our TPANet is summarized in Figure 4. In the following, we first show the multi-task and single-task baselines in Subsection 3.2. Then we introduce how we capture the task-relevant prior information with inductive biases and their respective characteristics (Subsection 3.3, Figure 4(a)). Then, we introduce our spatial mixing and channel mixing in the adaptive task mixing module for adaptive task interaction (Subsection 3.4, Figure 4(b)). Further, we show a cross attention module for querying task-specific features (Subsection 3.5, Figure 4(c)). Finally, we show the loss functions of the all tasks (Subsection 3.6).

3.2 Multi-task and single-task baselines

Our TPANet model is compatible with both CNN-based backbones and transformer-based backbones. We extract the features of each stage from the CNN or transformer backbones, as shown in Figure 3.

First, for the multi-task baseline method, the input image $x_{img} \in \mathbb{R}^{H\times W\times 3}$ is first fed into the backbone (swin or HRNet), where the image is processed through four stages. We carefully collect the output of each stage of the backbone. Second, the stage 2, stage 3 and stage 4 feature maps are up-sampled to match the resolution of the stage 1 output feature ($\mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times C}$) via bilinear interpolation. Thus, after up-sampling, the stage 2, 3, and 4 output feature map is up-sampled to $\mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times 2C}$, $\mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times 4C}$ and $\mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times 8C}$, respectively. Finally, we concatenate the four stage outputs along with the channel dimension to obtain a feature map $x \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times 15C}$, where H, W, and C are the height, width, and channel of the image feature, respectively. In this way, we aggregate the multi-scale feature maps from the backbone as a shared feature map ($x \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times 15C}$). The image feature from the backbone is then used by the task-specific heads to perform the dense predictions for every task. Single-task baseline also employ the shared feature map ($x \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times 15C}$). Unlike the multi-task baseline, the single-task baseline trains an individual model for each task. Therefore, when a single-task model is used for multiple vision tasks, multiple models need to be trained, and more training time is spent. The potential benefit of the multi-task model compared to the single-task model is an efficient prediction with fewer parameters and less computational cost. However, on some tasks, the single-task model performs better than the multi-task model.

We use the baseline and proposed modules for our TPANet multi-task method. Our TPANet contains three tailored designs, including (a) a task prior extractor module for providing task-relevant prior information from the convolution, (b) an adaptive task mixing module for conducting task interaction, and (c) a cross attention module for querying the task-specific feature map with task-relevant information. Finally, we obtain multiple feature maps according to the task number, which can be used to conduct dense prediction tasks.

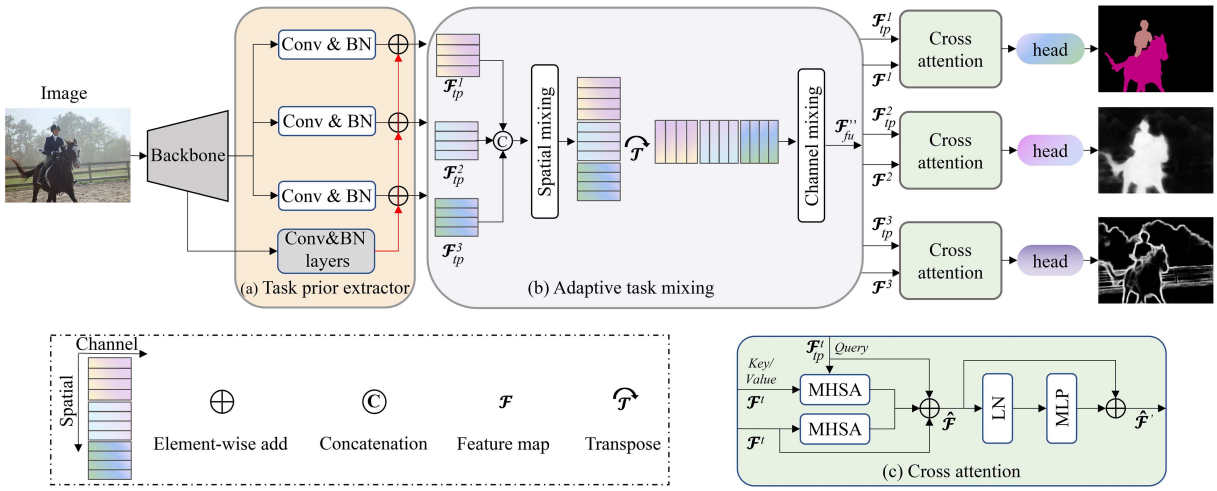

Figure 4 (Color online) Illustration of the TPANet framework. Our TPANet consists of three key designs: (a) task prior extractor, (b) adaptive task mixing, and (c) cross attention. We first process an image by backbone to generate feature maps. (a) Task prior extractor provides task-relevant prior information from the convolution. The outputs of the task prior extractor are concatenated along the channel dimension before passing them through (b) adaptive task mixing. We adapt the adaptive task mixing via spatial and channel mixing for task interaction. Cross attention (c) generates a task-specific feature map $\hat{\mathcal{F}}'$ corresponding to a specific task, which is then fed into the task-specific head to complete the final prediction.

3.3 Task prior extractor module

We design the task prior extractor to produce the task-relevant prior information with inductive bias from convolution. In particular, each task has an independent Conv & BN block that generates one task-specific prior feature incorporating inductive bias. The output of a shared Conv & BN layers block is then added to each task-specific feature. We introduce additional inductive biases into task-specific features, inspired by local window attention with CNN [10] and the reduction cell with CNN [48]. These convolutional operations encode well-established inductive biases such as locality, translation invariance, and spatial continuity, which are intrinsic to CNN architectures. By embedding these task-relevant prior information, the proposed task prior extractor module enhances the model's learning efficiency.

Conv & BN block. The feature map $\mathbf{x}$ is fed into the task prior extractor module. As shown in Figure 4(a), the number of the Conv & BN block is according to the task numbers in the task prior extractor module. We leverage a convolution with batch normalization (Norm) to obtain a task-specific feature map $\mathcal{F}_{te}^{t}$ ($t \in [1, T]$, t indicates task number) for each task. This procedure can be written as

$$\mathcal{F}_{te}^{t} = \mathrm{Norm}(W_t(\mathbf{x}) + b_t), \tag{1}$$

where W_t is the the learnable weights; b_t is the learnable bias. According to the task number, we collect the task-specific feature maps $\{\mathcal{F}_{te}^{1}, \mathcal{F}_{te}^{2}, \ldots, \mathcal{F}_{te}^{T}\} \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times D}$ where D is the channel dimension.

Conv & BN layers block. Convolutions naturally equip with inductive bias and compute local correlation for neighbor pixels. It specializes in capturing local features (i.e., boundaries and corners). We design the Conv & BN layers block to introduce the task-relevant prior information and locality inductive biases from multiple convolution layers into TPANet. We also analyze the effect of Conv & BN layers block depths. Specifically, 3×3 convolution layers generate more task-relevant prior information and then add it to the task-specific feature maps (i.e., Eq. (1)). The feature map $\mathbf{x}$ from the backbone is fed directly into a Conv & BN layers block to extract the inductive biases ($Depth = 1$ in practice). Such an output feature is considered to have task-relevant prior information, i.e.,

$$\mathcal{F}_{tp} = \mathrm{Norm}(W_{tp}(\mathbf{x}) + b), \tag{2}$$

where the W_{tp} is the the learnable weights; the $\mathcal{F}_{tp} \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times D}$. Next, the feature map is employed to element-wise add with each $\mathcal{F}_{te}^{t}$, which could be formulated as

$$\mathcal{F}_{tp}^{t} = \mathcal{F}_{te}^{t} + \mathcal{F}_{tp}, \tag{3}$$

where $\mathcal{F}_{tp}^{t} \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times D}$ is called task-relevant prior information. The complete set of features $\mathcal{E}_T = [\mathcal{F}_{te}^{1}, \mathcal{F}_{te}^{2}, \ldots, \mathcal{F}_{te}^{T}]$, $(\mathcal{F}_{tp}^{t} \in \mathcal{E}_T)$.

3.4 Adaptive task mixing module

We first concatenate the collected features set $\mathcal{E}_T$ along the channel, denoted as $\mathcal{F}_{fu} \in \mathbb{R}^{S\times TD}$ ($S = \frac{H}{4} \times \frac{W}{4}$), that represents the fused feature map. The visual illustration of the proposed adaptive task mixing can be found in Figure 4(b). The adaptive task mixing module consists of spatial mixing and channel mixing, which are responsible for spatial and channel interaction, respectively. The MLP consists of two fully-connected layers and a GELU nonlinearity:

$$\mathrm{MLP}(\mathbf{x}) = W_2\sigma\left(W_1\,\mathrm{LN}(\mathbf{x})\right), \tag{4}$$

where the W_1 and W_2 are learnable weights. LN is a layer norm operation. The σ is a nonlinearity function (GELU).

Spatial mixing. As shown in Figure 4(b), we first perform spatial mixing. Spatial mixing acts on spatial dimension of $\mathcal{F}_{fu}$ (it is transposed input feature map $\mathcal{F}_{fu}^{\mathcal{T}}$) and maps $\mathbb{R}^S \mapsto \mathbb{R}^S$. This spatial mixing is calculated with residual connection:

$$\mathcal{F}'_{fu} = \mathcal{F}_{fu} + \text{Spatial-MLP}(\mathrm{LN}(\mathcal{F}_{fu})), \tag{5}$$

where LN refers to LayerNorm; $\mathcal{F}'_{fu} \in \mathbb{R}^{S\times TD}$.

Channel mixing. Channel mixing acts on the channel dimension of $\mathcal{F}_{fu}$ (it is the transposed input feature map from spatial mixing) and maps $\mathbb{R}^{TD} \mapsto \mathbb{R}^{TD}$. This channel mixing equation is expressed with a residual connection as follows:

$$\mathcal{F}''_{fu} = \mathcal{F}'_{fu} + \text{Channel-MLP}(\mathrm{LN}(\mathcal{F}'_{fu})), \tag{6}$$

where $\mathcal{F}''_{fu} \in \mathbb{R}^{S\times TD}$. The adaptive task mixing module can facilitate spatial and channel interactions.

We can perform a split operation along the channel dimension of the feature to match the dimension of a single task:

$$\mathrm{Split}(\mathcal{F}''_{fu}) = \{\mathcal{F}^1, \mathcal{F}^2, \ldots, \mathcal{F}^T\}, \tag{7}$$

where $\mathcal{F}^T \in \mathbb{R}^{S\times D}$.

3.5 Cross attention module

As shown in Figure 4, this module is leveraged to increase task awareness by a query-based transformer to integrate the task-relevant prior and task interaction information. We follow [47] to multi-head self-attention (MHSA) in computing similarity:

$$\mathrm{MHSA}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^{\mathcal{T}}}{\sqrt{d}}\right)V, \tag{8}$$

where Q, K, and V are the *query*, *key*, and *value* matrices. d is the *query*/*key* dimension. The cross attention module is applied to generate task-specific features via self-attention.

$\mathcal{F}_{tp}^t$ and $\mathcal{F}^T$ are then processed by a cross attention module to generate the task-specific feature map. As shown in Figure 4(c), we leverage a shared MHSA in a cross attention module for a task. This process can be formulated as follows:

$$\hat{\mathcal{F}}_a = \mathrm{MHSA}(Q = \mathcal{F}^T, K = \mathcal{F}^T, V = \mathcal{F}^T), \tag{9}$$

where $\mathcal{F}^T$ is applied as *query, key*, and *value* from (7). We then develop query-based self-attention:

$$\hat{\mathcal{F}}_q = \mathrm{MHSA}(Q = \mathcal{F}_{tp}^t, K = \mathcal{F}^T, V = \mathcal{F}^T), \tag{10}$$

in which the $\mathcal{F}_{tp}^t$ is applied as *query* from (3); $\hat{\mathcal{F}}^T$ is applied as the *key* and *value* in MHSA. Notice that in practice, the weights of MHSA are shared in (9) and (10) in the cross attention module. We use element-wise adds, represented as

$$\hat{\mathcal{F}} = \hat{\mathcal{F}}_a + \hat{\mathcal{F}}_q + \mathcal{F}_{tp}^t + \mathcal{F}^T, \tag{11}$$

where $\hat{\mathcal{F}} \in \mathbb{R}^{S\times D}$. Finally, it is fed into MLP with a residual connection to get the output feature:

$$\hat{\mathcal{F}}' = \mathrm{MLP}(\hat{\mathcal{F}}) + \hat{\mathcal{F}}. \tag{12}$$

As shown in Figure 4, each task corresponds to a cross attention module. We feed the feature map $\hat{\mathcal{F}}'$ to a task-specific head to get the final prediction.

Table 1 The task-specific hyper-parameters and losses of our TPANet framework.

Task (t)	λ_t	$\mathcal{L}_t$
Semantic segmentation ('SemSeg')	1.0	Cross entropy loss
Human parts segmentation ('PartSeg')	2.0	Cross entropy loss
Depth estimation ('Depth')	1.0	L1 loss
Saliency estimation ('Sal')	5.0	Cross entropy loss
Surface normal estimation ('Normal')	10.0	L1 loss
Boundary detection ('Bound')	50.0	Binary cross entropy loss

The cross-attention modules are non-shared, meaning that attention weights and parameters are not reused across tasks. This design allows each task to exploit its task-specific priors in a tailored manner. The output of each cross-attention module is subsequently fed into the corresponding task-specific prediction head (e.g., SemSeg, Depth, Saliency, etc.). Each task-specific head comprises three components: a bottleneck, a final prediction layer, and an upsampling module to restore the original spatial resolution. The bottleneck employs a Conv-BN-ReLU block to reduce the dimensionality of the shared features and refine them according to the target task's requirements, ensuring that the shared representation is effectively adapted to each task's unique feature space. The final prediction layer applies a 1×1 convolution to map the adapted features to the appropriate number of output channels (e.g., the number of semantic classes, a single channel for depth, etc.). Finally, the upsampling layer uses bilinear interpolation to resize the logits to the original input resolution, enabling dense prediction.

3.6 Overall loss functions

The multi-task loss is a key component in multi-task learning, representing the combined loss function that is optimized across multiple related tasks simultaneously. The formulation of the multi-task loss depends on the specific tasks and their corresponding functions. The overall TPANet loss $\mathcal{L}_{total}$ is the weighted sum of the presented loss components:

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

with λ_t being a hyper-parameter weighting in a task loss $\mathcal{L}_t$. T denotes the total number of tasks ($t \in [1, T]$) (see Table 1).

The NYUD-v2 dataset contains four tasks: semantic segmentation (SemSeg), depth estimation (Depth), surface normal estimation (Normal), and boundary detection (Bound). The losses can be written as

$$\mathcal{L}_{total} = \lambda_{seg}\mathcal{L}_{seg} + \lambda_{depth}\mathcal{L}_{depth} + \lambda_{normal}\mathcal{L}_{normal} + \lambda_{bound}\mathcal{L}_{bound}, \tag{14}$$

where $\lambda_{seg} = 1.0$, $\lambda_{depth} = 1.0$, $\lambda_{normal} = 10.0$, $\lambda_{bound} = 50.0$. $\mathcal{L}_{seg}$ is cross entropy loss, which computes the cross entropy loss between the input and target. $\mathcal{L}_{depth}$ and $\mathcal{L}_{normal}$ are L1Loss, which measures the mean absolute error (MAE) between each element in the input and target. $\mathcal{L}_{bound}$ is a binary cross entropy loss, which measures binary cross entropy between target and input logits.

The PASCAL-Context contains five tasks: SemSeg, human parts segmentation (PartSeg), saliency estimation (Sal), Normal, and Bound tasks. The losses can be written as

$$\mathcal{L}_{total} = \lambda_{seg}\mathcal{L}_{seg} + \lambda_{partseg}\mathcal{L}_{partseg} + \lambda_{sal}\mathcal{L}_{sal} + \lambda_{normal}\mathcal{L}_{normal} + \lambda_{bound}\mathcal{L}_{bound}, \tag{15}$$

where $\lambda_{seg} = 1.0$, $\lambda_{partseg} = 2.0$, $\lambda_{sal} = 5.0$, $\lambda_{normal} = 10.0$, and $\lambda_{bound} = 50.0$. $\mathcal{L}_{seg}$, $\mathcal{L}_{partseg}$ and $\mathcal{L}_{sal}$ are cross entropy losses, which compute the cross entropy loss between input and target. $\mathcal{L}_{normal}$ is L1Loss, which measures the MAE between each element in the input and target. $\mathcal{L}_{bound}$ is a binary cross entropy loss, which measures binary cross entropy between target and input logits.

3.7 Algorithm pseudocode

We include the pseudocode of our TPANet model in Algorithm 1. Our TPANet consists of three tailored modules: task prior extractor, adaptive task mixing and cross attention modules.

Algorithm 1 Framework of TPANet.

Require: x_{img}; task-learning rate: 0.00002;
Ensure: multi-task model Θ_{mt};
while *iteration* $\leqslant$ 40k **do**
  if task prior extractor is used **then**
    Update $\mathcal{F}_{te}^{t}$ by (1) and (2);
    Update $\mathcal{F}_{tp}^{t}$ by (3);
  end if
  if adaptive task mixing is used **then**
    Update $\mathcal{F}_{fu}^{'}$ by (5);
    Update $\mathcal{F}_{fu}^{''}$ by (6);
  end if
  if cross attention is used **then**
    Update $\hat{\mathcal{F}}$ by (9)–(11);
    Update $\hat{\mathcal{F}}^{'}$ by (12);
  end if
  Calculate multi-task loss via (13);
  Update the model parameters Θ_{mt} on loss using back propagation;
end while
Perform multi-task predictions using model Θ_{mt}.

4 Experiment

4.1 Experimental setup

NYUD-v2 dataset and metrics. NYUD-v2 comprises RGB and Depth frames. 795 images are used for training and 654 images for testing. NYUD-v2 is adopted for SemSeg, Depth, Normal and Bound tasks by providing dense labels for every image. There are four evaluation metrics to evaluate our model: mean Intersection over Union (mIoU) for the SemSeg task, root mean square error (rmse) for the Depth task, mean Error (mErr) for the Normal task, and optimal dataset scale F-measure (odsF) for the Bound task. Key experiments are repeated three times, reporting mean and standard deviation (i.e., std.).

PASCAL-Context dataset and metrics. PASCAL-Context training and validation contain 10103 images, while testing contains 9637 images. PASCAL-Context is usually adopted for SemSeg, PartSeg, Sal, Normal, and Bound tasks by providing annotations for the whole scene. There are five evaluation metrics to compare our model with other multi-task models: mIoU for the SemSeg and PartSeg tasks, mErr for the normal task, odsF for the bound task, and maximum F-measure (maxF) for the saliency task. 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: higher is better.

Implementation details. We conduct experiments on two publicly popular MTL datasets, NYUD-v2 [18] and PASCAL-Context [19]. For all experiments, we use CNN-based architectures (i.e., HRNetV2p-W18-Small (HRNet18) [2], hrnetv2p-w48 (HRNet48)) and transformer-based architectures (i.e., Swin-Tiny (Swin-T), Swin-Small (Swin-S), Swin-Base (Swin-B), Swin-Large (Swin-L) [10]) as our backbone for TPANet, respectively. As shown in Figure 4, our Conv & BN block in the task prior extractor and cross attention module numbers changes dynamically according to the number of tasks. For example, when we have five tasks, our method can automatically generate five Conv & BN blocks and five cross-attention modules. Our models are optimized using the AdamW policy. We use a learning rate of 0.00002 with a weight decay of 0.000001 and train the model for 40000 iterations. The dropout number (κ) in MLP is 0. We report our results for $\kappa \in \{0, 0.1, 0.2, 0.3\}$. We use the $\kappa = 0$ setting in our model.

Baselines. We adopt the standard practice of evaluating our proposed method against the single-task and multi-task baseline versions, which are based on HRNet [2] and swin transformer [10] in our case. Baselines consist of the single-task baseline and multi-task baseline. The single-task baseline network is trained using a backbone and task-specific head for a task. Furthermore, the multi-task baseline network is trained using a shared backbone and multiple task-specific heads for multiple tasks. In Tables 2 and 3, we list the single-task and multi-task performance using different backbones on multiple vision tasks.

4.2 Results

Results on 4-task NYUD-v2. In Table 2, we first report the four task results in different metrics on the NYUD-v2 dataset. We also provide a quantitative evaluation of the computational cost (GFLOPs) and parameters. Table 2 shows a comparison with the state-of-the-art approaches. Following [7], we use the same backbone and

Table 2 We report the comparison of the MTL models with the state-of-the-art on the NYUD-v2 dataset. '↓': lower is better. '↑': higher is better. Δ_m denotes the average per-task performance drop. Swin-◇ indicates that the specific swin model is uncertain. ‡ denotes results not reported in InvPT [30] but from our test.

Model	Backbone	Params	FLOPs	SemSeg (mIoU)↑		Depth (rmse)↓		Normal (mErr)↓		Bound (odsF)↑		Δ_m↑
		(M)	(G)	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	(%)
Single-task baseline	HRNet18	16.09	40.93	38.02	0.14	0.6104	0.0041	20.94	0.08	76.22	0.07	0.00
Multi-task baseline	HRNet18	4.52	17.59	36.35	0.26	0.6284	0.0034	21.02	0.06	76.36	0.05	−1.89
Cross-Stitch [3]	HRNet18	4.52	17.59	36.34	0.55	0.6290	0.0051	20.88	0.04	76.38	0.07	−1.75
Pad-Net [4]	HRNet18	5.02	25.18	36.70	0.16	0.6264	0.0021	20.85	0.03	76.50	0.06	−1.33
PAP [24]	HRNet18	4.54	53.04	36.72	0.31	0.6178	0.0065	20.82	0.03	76.42	0.07	−0.95
PSD [6]	HRNet18	4.71	21.10	36.69	0.55	0.6246	0.0036	20.87	0.07	76.42	0.13	−1.30
NDDR-CNN [5]	HRNet18	4.59	18.68	36.72	0.31	0.6288	0.0037	20.89	0.02	76.32	0.07	−1.51
MTI-Net [27]	HRNet18	5.50	32.42	36.61	0.15	0.6270	0.0048	20.85	0.03	76.38	0.07	−1.44
ATRC [7]	HRNet18	5.06	25.76	38.90	0.43	0.6010	0.0046	20.48	0.02	76.34	0.12	1.56
TPANet (ours)	HRNet18	5.18	27.02	39.43	0.31	0.5931	0.0040	20.39	0.02	76.39	0.07	2.18
Single-task baseline	Swin-T	115.08	161.25	38.02	0.21	0.6104	0.0035	20.94	0.06	76.22	0.13	0.00
Multi-task baseline	Swin-T	32.50	96.29	38.78	0.28	0.6312	0.0032	21.05	0.08	75.60	0.08	−3.74
MQTransformer [13]	Swin-T	35.35	106.02	43.61	0.32	0.5979	0.0024	20.05	0.05	76.20	0.06	0.31
InvPT [30]	Swin-T	60.14‡	162.52‡	44.27	0.35	0.5589	0.0026	20.46	0.04	76.10	0.09	2.59
TPANet (ours)	Swin-T	34.69	164.9	46.51	0.26	0.5987	0.0023	20.71	0.03	76.90	0.05	2.71
Single-task baseline	Swin-S	200.33	242.63	48.92	0.28	0.5804	0.0036	20.94	0.08	77.20	0.14	0.00
Multi-task baseline	Swin-S	53.82	116.63	47.90	0.34	0.6053	0.0025	21.17	0.05	76.90	0.09	−1.96
MQTransformer [13]	Swin-S	56.67	126.37	49.18	0.31	0.5785	0.0031	20.81	0.04	77.00	0.07	1.59
MTFormer [15]	Swin-◇	64.03	117.73	50.56	–	0.4830	–	–	–	–	–	4.12
TPANet (ours)	Swin-S	53.34	185.25	50.90	0.24	0.5603	0.0022	20.05	0.03	78.20	0.06	3.19
Single task baseline	Swin-L	789.96	819.93	56.46	0.28	0.508	0.0047	19.38	0.09	78.8	0.14	0.00
Multi-task baseline	Swin-L	204.96	316.87	54.53	0.34	0.532	0.0039	19.51	0.07	78.3	0.08	−2.36
InvPT [30]	Swin-L	292.7‡	417.27‡	51.76	0.28	0.5020	0.0028	19.39	0.05	77.60	0.06	−2.22‡
InvPT [30]	ViT-L	402.1‡	555.57‡	53.56	0.21	0.5183	0.0031	19.04	0.06	78.10	0.06	–
TPANet (ours)	Swin-L	205.61	378.58	56.42	0.17	0.5018	0.0026	19.02	0.03	79.10	0.06	0.82

training setting for a fair comparison. We find that the TPANet model outperforms InvPT in terms of multi-tasking performance (ours 46.51 vs. InvPT 44.27). When equipped with Swin-S as the backbone, the TPANet achieves comparable performance at 50.90 mIoU with a significant parameter (53.34M). Concretely, our TPANet model outperforms the previous best by +0.34 (ours 50.90 vs. MTFormer 50.56) on the SemSeg task while performing worse on the depth task. The poor depth estimation accuracy is because MTFormer only performed two tasks while we performed four. Even when compared to state-of-the-art models with a similar number of parameters, our method can yield the highest mIoU and ranks first on the Swin-S. Note that both our TPANet and InvPT [30] use Swin-L as the backbone and our approach outperforms InvPT on all tasks, especially SemSeg (ours: 56.42 mIoU vs. InvPT: 51.76 mIoU) and Bound (ours: 79.1 odsF vs. InvPT: 77.6 odsF). Furthermore, the prior art InvPT [30] using ViT-L as backbone trains the MTL model with 402.1M and 555.57G FLOPs. Our TPANet surpasses the InvPT [30] using ViT-L by a considerable margin while using fewer parameters (205.61M) and GFLOPs (378.58G). This demonstrates the strong performance of our TPANet model using different backbones across semantic segmentation, depth estimation, surface normal estimation and boundary detection tasks. As shown in Table 2, our TPANet benefits from the advantages of both task-relevant prior information and query-based transformer that shows strong performance on all the metrics. From these quantitative (Table 2) and qualitative results, TPANet demonstrates the ability to make highly accurate predictions across a wide range of tasks, while using fewer parameters and GFLOPs.

Results on 5-task PASCAL-Context. As shown in Table 3, we further evaluate our method on the PASCAL-Context dataset and then report the five task results in different metrics. To show the effectiveness and friendly compatibility of our TPANet, we conduct experiments using different backbones, e.g., HRNet18 [2], Swin-T, Swin-S, Swin-B, and Swin-L [10]. Specifically, using HRNet-18, our TPANet method outperforms the MQTransformer baseline by 1.06 mIoU on the SemSeg task. Experimental results of our method with Swin-B show significant improvements compared to the multi-task baseline. With the large transformer-based Swin-B as the backbone, our model achieves 75.56 mIoU, surpassing the much stronger MTFormer baseline by +1.41 mIoU on the SemSeg task. TPANet achieves competitive performance on other tasks as well as on PASCAL-Context. The results show that

Table 3 We report a comparison of the MTL models on the PASCAL-Context dataset. Δ_m denotes the average per-task performance drop (higher is better). Swin-◇ indicates that the specific swin model is uncertain.

Model	Backbone	SemSeg (mIoU)↑	PartSeg (mIoU)↑	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 [4]	HRNet18	53.60	59.60	65.80	15.3	72.50	−4.41
ATRC [7]	HRNet18	57.89	57.33	83.77	13.99	69.74	−4.45
MQTransformer [13]	HRNet18	58.91	57.43	83.78	14.17	69.80	−4.20
TPANet (ours)	HRNet18	59.97	58.21	84.13	13.92	69.86	−3.22
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 [13]	Swin-T	68.24	57.05	83.40	14.56	71.10	1.07
TPANet (ours)	Swin-T	69.08	57.61	82.54	14.46	71.20	1.42
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 [13]	Swin-S	71.25	60.11	84.05	14.74	71.80	1.27
TPANet (ours)	Swin-S	71.59	60.38	83.20	14.65	72.00	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
MTFormer [15]	Swin-◇	74.15	64.89	67.71	–	–	2.41
TPANet (ours)	Swin-B	75.56	64.91	83.46	14.67	73.10	1.3
Single task baseline	Swin-L	79.26	68.92	83.84	14.28	74.50	0.00
Multi-task baseline	Swin-L	77.35	63.86	82.87	14.84	73.10	−3.33
TPANet (ours)	Swin-L	78.11	68.01	83.65	14.38	74.80	−0.67

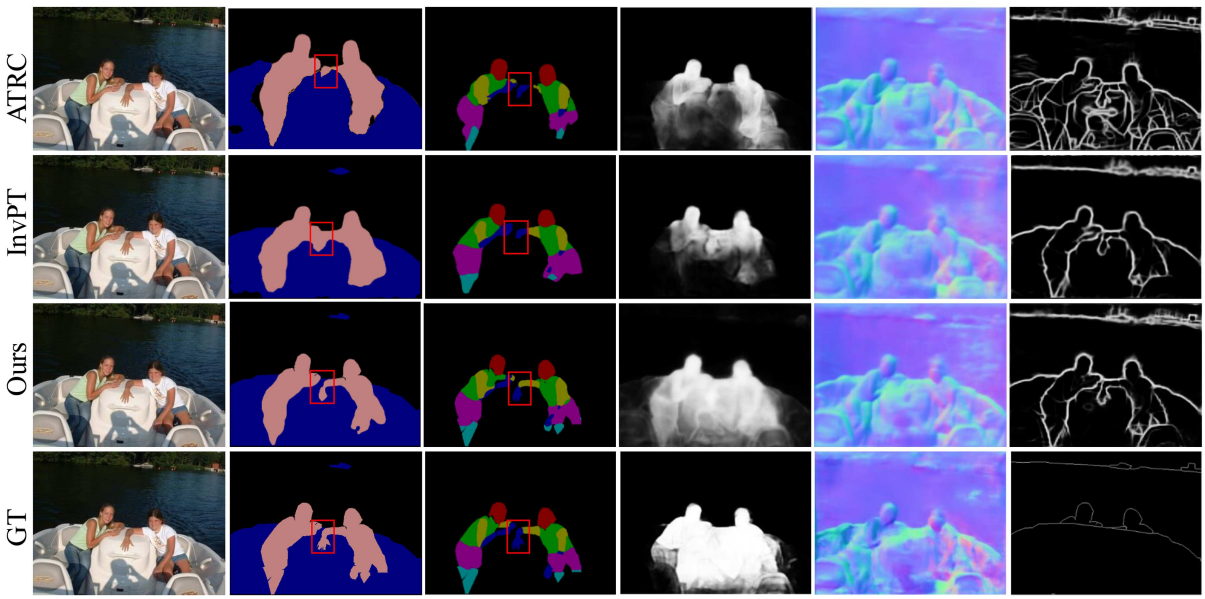

Figure 5 (Color online) Qualitative results of our TPANet compared with the previous MTL methods (i.e., ATRC and InvPT) on the PASCAL-Context dataset. The visualizations (notice the red boxes) emphasize the accuracy and efficiency of our TPANet in multiple vision tasks. From top to bottom: ATRC [7], InvPT [30], TPANet (ours) and ground truth (GT).

our TPANet is relatively robust to varying CNN-based and transformer-based backbones. Finally, we also report dense prediction results for the PASCAL-Context in Figure 5. These results show the good generalization ability of our TPANet models.

Table 4 Ablation studies on NYUD-v2 dataset using a Swin-T backbone. Task prior extractor (TPE), adaptive task mixing (ATM), and cross attention (CA) modules are the parts of our model. 'S Only' means ATM with spatial mixing only. 'w/' indicates "with". The best results are in bold.

(a) Ablation on modules

Model	SemSeg (mIoU)↑	Depth (rmse)↓	Normal (mErr)↓	Bound (odsF)↑
Multi-task baseline	38.78	0.6312	21.05	75.6
w/TPE	43.44	0.6124	20.83	76.4
w/TPE+ATM(S Only)	44.03	0.6092	20.82	76.4
w/TPE+ATM(Full)	44.21	0.6080	20.97	76.6
w/TPE+ATM+CA	**46.49**	**0.5987**	**20.71**	**76.9**

(b) Ablation on four-scale features

Scale	SemSeg (mIoU)↑	Depth (rmse)↓	Normal (mErr)↓	Bound (odsF)↑
1/32	36.89	0.6175	22.85	75.9
1/16, 1/32	41.64	0.6177	22.75	76.4
1/8, 1/16, 1/32	42.10	0.6163	22.78	76.4
1/4, 1/8, 1/16, 1/32	**46.49**	**0.5987**	**20.71**	**76.9**

4.3 Ablation studies

Ablation on the proposed modules. Our ablation studies explore the utility of using different modules in our method. We refer to our full method as TPANet and consider the following ablations: (1) w/ TPE: with the task prior extractor module; (2) w/ TPE+ATM: with the task prior extractor and adaptive task mixing modules; (3) w/ TPE+ATM+CA: with the task prior, adaptive task mixing and cross attention modules. We perform ablations to investigate how it benefits from the task-relevant prior information. As shown in Table 4(a), our model achieves strong accuracy performance when equipped with the task prior extractor module. We find that qualitative results using TPE can gain 4.6 mIoU on SemSeg task compared to multi-task baseline. These results demonstrate that introducing task-relevant prior information might be an effective way to facilitate local visual modeling and improve task performance. It can be observed that, with ATM and CA modules, TPANet achieves better performance when compared with the baseline. Thus, the qualitative results show that ATM can effectively adapt to task interactions along spatial and channel dimensions. Further, the non-shared cross attention is designed to be suitable for multiple vision scenarios. The task-relevant prior extracted by TPE guides the cross-attention module to focus on task-discriminative regions, improving feature aggregation and inter-task consistency. This interaction further enhances the model's contextual reasoning across tasks.

Ablation on the scales. The four-scale features (1/4, 1/8, 1/16 and 1/32) come from the four stages in the backbone network. Table 4(b) lists the experimental results, showing that the performance can be consistently improved with the value of the scale number. We conduct an experiment only using a 1/32 feature and achieve good performance. This indicates that 1/32 features have rich semantic information. We notice that our model achieves the best performance when using four-scale features from the backbone. This demonstrates that multi-scale features can provide more semantic information, which would be beneficial for pixel-level vision tasks.

4.4 Hyperparameter analysis

Impact of the dropout number. We test TPANet with different dropout numbers, listed in Table 5. In the cross attention module, dropout operations exist for the MLP in the cross attention module. To explore the impact of the number of dropouts in our model, we set the dropout numbers $\kappa \in \{0, 0.1, 0.2, 0.3, 0.5\}$. The default dropout number setting is 0.1 in our model.

Effectiveness of the depth of Conv & BN layers. In Table 6, we analyze the effect of Conv & BN layers depth. As shown in Table 6, the final performance gets (slightly) increased with a deeper Conv & BN layers. Note that a depth of 1 is a competitive choice compared to a 3-depth Conv & BN layers, it significantly reduces the computation cost while only marginally sacrificing the accuracy by +0.26 on the SemSeg task. As the Conv & BN layers depth increases, the overall performance shows an increasing trend. The results demonstrate that the task-relevant prior information with inductive bias provided by 3×3 convolution contributes to MTL performance improvement. To balance computational effort and performance, we choose a Conv & BN layers depth of 1 as the default setting.

Table 5 Impact of the dropout (κ). We perform this ablation using Swin-T as the backbone on the NYUD-v2 dataset.

κ	SemSeg (mIoU)↑	Depth (rmse)↓	Normal (mErr)↓	Bound (odsF)↑
0	**46.49**	0.5987	20.71	**76.90**
0.1	46.29	**0.5967**	20.82	76.80
0.2	46.42	0.6078	**20.63**	**76.90**
0.3	46.41	0.6073	20.66	**76.90**
0.5	46.26	0.6089	20.67	76.70

Table 6 Effectiveness of varying depth of Conv & BN layers using Swin-T in task prior extractor module on NYUD-v2 dataset.

Depth	SemSeg (IoU)↑	Depth (rmse)↓	Normal (mErr)↓	Bound (odsF)↑
1	46.49	0.5987	20.71	76.9
2	46.36	0.5975	**20.62**	76.9
3	**46.63**	**0.5930**	20.77	**77.0**

Table 7 NYUD-v2 performance comparison, using Swin-B and Swin-L. We compare our model with the InvPT [30].

Method	Backbone	SemSeg (mIoU)↑	Depth (rmse)↓	Normal (mErr)↓	Bound (odsF)↑
Multi-task baseline	Swin-B	51.44	0.5813	20.44	77.0
InvPT [30]	Swin-B	50.97	0.5071	19.39	77.3
TPANet (ours)	Swin-B	53.09	0.5322	19.31	77.4
Multi-task baseline	Swin-L	51.44	0.5813	20.44	77.0
InvPT [30]	Swin-L	51.76	0.5020	19.39	77.6
TPANet (ours)	Swin-L	56.34	0.5019	19.02	79.10

Table 8 Ablation on the cross attention module. We design two optional formats: a non-shared cross attention module and a shared cross attention module. We perform this ablation using Swin-T as the backbone on the NYUD-v2 dataset.

Format	SemSeg (mIoU)↑	Depth (rmse)↓	Normal (mErr)↓	Bound (odsF)↑
Non-shared	46.49	0.5987	20.71	76.9
Shared	46.29	0.5967	20.82	76.8

Table 9 Ablation studies on NYUD-v2 dataset using a Swin-T backbone. Cross attention (CA) module uses different attention mechanisms using Swin-T as the backbone on the NYUD-v2 dataset. 'w/' and 'w/o' indicates "with" and "without".

Format	SemSeg (mIoU)↑	Depth (rmse)↓	Normal (mErr)↓	Bound (odsF)↑
CA(w/ task prior)	46.49	0.5987	20.71	76.9
CA(w/o task prior)	46.19	0.6017	20.82	76.5

Comparison across different backbones. In Table 7, we further compare our TPANet against more standard multi-task baselines and InvPT [30], which are pre-trained with the image dataset. On nearly all tasks, our TPANet method outperforms the supervised baselines and the previous best method InvPT [30]. Specifically, our TPANet method further outperforms the standard multi-task baselines and InvPT [30] on both the Swin-B (2.12 mIoU improvement on SemSeg) and Swin-L (4.59 mIoU improvement on SemSeg) backbones. Moreover, performance can further be improved by adopting larger transformer-based models as backbones; our method is still effective, efficient and robust. Experimental results demonstrate that our method achieves competitive performance with existing methods, and the performance can achieve performance leadership on different backbones on the NYUD-v2 dataset.

Effectiveness of non-shared and shared cross attention module. An important design decision of our TPANet is the non-shared cross attention module. As shown in Table 8, we compare different configurations (i.e., non-shared cross attention and shared cross attention). In fact, the non-shared cross attention module achieves better accuracy while requiring +0.2 mIoU, +0.11 mErr and +0.1 odsF. Thus, we use the non-shared cross attention module in TPANet.

Effectiveness of the cross attention module using different attention mechanisms. Table 9 shows

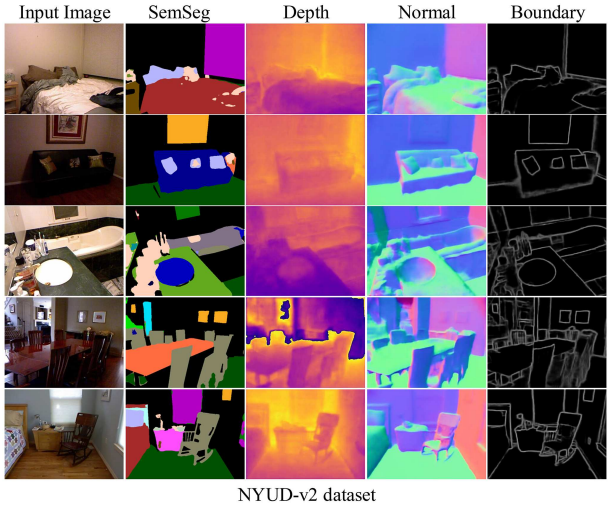

Figure 6 (Color online) Visualization of our TPANet for semantic segmentation (second column), depth estimation (third column), surface normal estimation (fourth column), and boundary detection (fifth column) on the NYUD-v2 dataset.

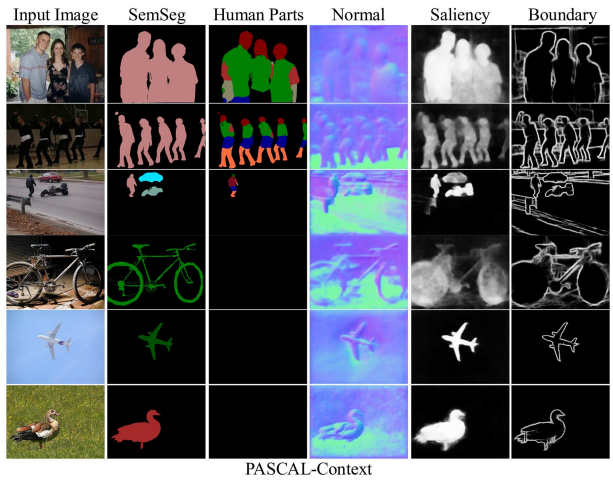

Figure 7 (Color online) Visualization of our TPANet for semantic segmentation (second column), human parts segmentation (third column), surface normal estimation (fourth column), saliency estimation (fifth column), and boundary detection (sixth column) on PASCAL-Context dataset.

an ablation study isolating the effect of the attention formulation within the cross attention module of TPANet. Two variants are compared: query-based attention with task-relevant prior information and conventional self-attention without task-relevant prior information. Across all tasks, the query-based attention with task priors consistently delivers performance gains, improving segmentation accuracy (+0.30 mIoU), reducing depth (+0.003 rmse) and normal (+0.11 mErr), and enhancing boundary detection (+0.4 odsF). These consistent improvements across multiple metrics demonstrate that task-relevant prior query-based attention provides a more effective fusion strategy for multi-task dense prediction than standard self-attention.

4.5 Visualization and analysis

To demonstrate the capability of our TPANet in an intuitive perspective, we visualize the six task predictions of the selected images, shown in Figures 5–7.

Visual comparisons. To gain insights into the learned representations, we conduct the visual comparisons among tasks on the PASCAL-Context dataset. Figure 5 shows the visual comparison of different visualization methods. We observe that our TPANet gives overall better visualizations than the baseline model, including the whole tasks, as shown in Figure 5. For the segmentation task, in Figure 5, we observe that TPANet obtains more precise semantic segmentation and human parts segmentation, where the red box highlights the semantic information in some subfigures. Specifically, comparing ATRC [7] and InvPT [30] with our TPANet in the first and second columns, we can see that ATRC [7] and InvPT [30] fail to distinguish the arms and hands of the two people. We use red boxes to mark the exact locations and quickly find the semantic segmentation differences between the three methods. While our TPANet successfully differentiates the two objects, suggesting that ours learn more semantic features. In addition, the saliency (fifth column in Figure 5) and boundary (sixth column in Figure 5) predictions using our model also focus on the region of the fertile semantic. These visual comparisons demonstrate that our model can capture more local boundaries and textures and further prove that our TPANet introduces the merit of CNN for capturing task-relevant prior information to transformer.

Visualization on 4-task NYUD-v2 dataset. To further analyze the property of our method, we show the four vision task predictions (i.e., semantic segmentation, depth estimation, surface normal estimation, and boundary detection) for qualitative results in Figure 6 on NYUD-v2. Each pixel was assigned a semantic label, such as wall, floor, furniture, or object category. The visualization of these maps offered an understanding of the variety and distribution of semantic classes within the scenes. We choose four samples from the dataset. As we can see, the visualizations clearly show that our method with a Swin-B backbone is able to obtain good predictions on multiple vision tasks. These findings facilitate a deeper comprehension of the dataset's complexities and the TPANet's ability to capture relevant visual cues. Moreover, the insights gained from the visualization analysis are shown in the context of improving scene understanding, depth estimation, and related tasks on similar indoor environments.

Visualization on 5-task PASCAL-Context dataset. As shown in Figure 7, we show five vision task predictions (i.e., semantic segmentation, human parts segmentation, surface normal estimation, saliency estimation, and boundary detection) from different types of images on the PASCAL-Context test set. The first and second rows of images show many people scenes that can also be segmented very well. The third row of images shows that our model can correctly distinguish between people and objects. The fourth, fifth, and sixth rows of images show that our model recognizes the objects. Differences can be seen between semantic segmentation and human parts segmentation. Figure 7 (the third row) shows a picture of a man walking towards a motorcycle and a car. On the human parts segmentation task, our model only segments the human, ignoring the motorcycle and car. For some qualitative segmentation examples, the fifth and sixth rows of images show the black on human parts segmentation (the third column). Moreover, in other tasks, qualitative results also demonstrate good visual performance. These qualitative results demonstrate that our model can capture more semantic features.

5 Discussion and limitation analysis

Discussion. TPANet introduces task-relevant prior information with inductive biases into the transformer, enabling more precise and adaptive task-specific feature extraction. This combination allows the transformer and convolution layers to focus on their respective strengths, namely modeling long-range dependencies and capturing local spatial details. Compared to existing methods, TPANet effectively addresses the transformer's limitations in local spatial modeling by guiding attention through task-relevant prior information, thereby enhancing both performance and balance in MTL across diverse dense prediction tasks. Moreover, TPANet excels in maintaining task balance. By leveraging task-specific attention guided by task-relevant prior information, it mitigates overfitting to dominant tasks, which is common in shared-encoder models, while still facilitating beneficial cross-task improvements in semantic segmentation, human part segmentation, saliency estimation, surface normal estimation, and boundary detection.

Limitation analysis. As shown in Figure 4, the use of task-specific cross-attention modules increases computational overhead, particularly when scaling to a large number of vision tasks. This is because TPANet, similar to previous studies, relies on individualized cross-attention mechanisms for task interaction. To address this efficiency challenge, model miniaturization through knowledge distillation is a promising direction and will be explored in future work.

6 Conclusion

In this paper, we explore the inductive biases effect in transformer-based MTL architecture, named TPANet, to effectively and efficiently perform dense predictions. By embedding task-relevant priors into the Transformer framework, TPANet enhances locality modeling and strengthens cross-task representation learning. Our TPANet achieves superior performance, especially on semantic segmentation, human part segmentation, depth estimation, saliency estimation, surface normal estimation and boundary detection tasks, compared to other transformer-based MTL architectures. Extensive experiments validate its superior accuracy, efficiency, and robustness.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 62431020, 62225113) and Foundation for Innovative Research Groups of Hubei Province (Grant No. 2024AFA017).

References

1 Vandenhende S, Georgoulis S, van Gansbeke W, et al. Multi-task learning for dense prediction tasks: a survey. IEEE Trans Pattern Anal Mach Intell, 2022, 44: 3614–3633
2 Sun K, Xiao B, Liu D, et al. Deep high-resolution representation learning for human pose estimation. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 5693–5703
3 Misra I, Shrivastava A, Gupta A, et al. Cross-stitch networks for multi-task learning. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016. 3994–4003
4 Xu D, Ouyang W, Wang X, et al. PAD-Net: multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018. 675–684
5 Gao Y, Ma J, Zhao M, et al. Nddr-cnn: layerwise feature fusing in multi-task CNNs by neural discriminative dimensionality reduction. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 3205–3214
6 Zhou L, Cui Z, Xu C, et al. Pattern-structure diffusion for multi-task learning. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 4513–4522
7 Bruggemann D, Kanakis M, Obukhov A, et al. Exploring relational context for multi-task dense prediction. In: Proceedings of IEEE/CVF International Conference on Computer Vision, 2021. 15849–15858
8 Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16×16 words: transformers for image recognition at scale. In: Proceedings of the 9th International Conference on Learning Representations, 2021
9 Wang W, Cao Y, Zhang J, et al. FP-DETR: detection transformer advanced by fully pre-training. In: Proceedings of the 10th International Conference on Learning Representations, 2022

10 Liu Z, Lin Y, Cao Y, et al. Swin Transformer: hierarchical vision transformer using shifted windows. In: Proceedings of IEEE/CVF International Conference on Computer Vision, 2021. 9992–10002
11 Bhattacharjee D, Zhang T, Süsstrunk S, et al. Mult: an end-to-end multitask learning transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 12031–12041
12 Liu S, Johns E, Davison A J. End-to-end multi-task learning with attention. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 1871–1880
13 Xu Y, Li X, Yuan H, et al. Multi-task learning with multi-query transformer for dense prediction. IEEE Trans Circ Syst Video Technol, 2023, 4: 1228–1240
14 Raychaudhuri D S, Suh Y, Schulter S, et al. Controllable dynamic multi-task architectures. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 10945–10954
15 Xu X, Zhao H, Vineet V, et al. MTFormer: multi-task learning via transformer and cross-task reasoning. In: Proceedings of European Conference on Computer Vision, 2022. 304–321
16 Xie E, Wang W, Yu Z, et al. SegFormer: simple and efficient design for semantic segmentation with transformers. In: Proceedings of Advances in Neural Information Processing Systems, 2021. 12077–12090
17 Chen Z, Duan Y, Wang W, et al. Vision transformer adapter for dense predictions. In: Proceedings of the 11th International Conference on Learning Representations, 2023
18 Silberman N, Hoiem D, Kohli P, et al. Indoor segmentation and support inference from RGBD images. In: Proceedings of European Conference on Computer Vision, 2012. 746–760
19 Chen X, Mottaghi R, Liu X, et al. Detect what you can: detecting and representing objects using holistic models and body parts. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2014. 1979–1986
20 Kendall A, Gal Y, Cipolla R. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018. 7482–7491
21 Chen Z, Badrinarayanan V, Lee C, et al. GradNorm: gradient normalization for adaptive loss balancing in deep multitask networks. In: Proceedings of the 35th International Conference on Machine Learning, 2018. 793–802
22 Sener O, Koltun V. Multi-task learning as multi-objective optimization. In: Proceedings of Advances in Neural Information Processing Systems, 2018. 525–536
23 Teichmann M, Weber M, Zoellner M, et al. Multinet: real-time joint semantic reasoning for autonomous driving. In: Proceedings of IEEE Intelligent Vehicles Symposium, 2018. 1013–1020
24 Zhang Z, Cui Z, Xu C, et al. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 4106–4115
25 Jiang J, Li Y, Chen L, et al. Multitask deep learning-based multiuser hybrid beamforming for mm-wave orthogonal frequency division multiple access systems. Sci China Inf Sci, 2020, 63: 180303
26 Gao Y, Bai H, Jie Z, et al. MTL-NAS: task-agnostic neural architecture search towards general-purpose multi-task learning. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 11540–11549
27 Vandenhende S, Georgoulis S, Gool LV. MTI-Net: multi-scale task interaction networks for multi-task learning. In: Proceedings of European Conference on Computer Vision, 2020. 527–543
28 Xu Y, Zhang L. DGMLP: deformable gating MLP sharing for multi-task learning. In: Proceedings of CAAI International Conference on Artificial Intelligence, 2022. 117–128
29 Xu Y, Yang Y, Zhang L. DeMT: deformable mixer transformer for multi-task learning of dense prediction. AAAI, 2023, 37: 3072–3080
30 Ye H, Xu D. Inverted pyramid multi-task transformer for dense scene understanding. In: Proceedings of European Conference on Computer Vision, 2022. 514–530
31 Carion N, Massa F, Synnaeve G, et al. End-to-End object detection with transformers. In: Proceedings of European Conference on Computer Vision, 2020. 213–229
32 d'Ascoli S, Touvron H, Leavitt M L, et al. ConViT: improving vision transformers with soft convolutional inductive biases. In: Proceedings of the 38th International Conference on Machine Learning, 2021. 2286–2296
33 Hussain S, Xi X M, Ullah I, et al. Difficulty-aware prior-guided hierarchical network for adaptive segmentation of breast tumors. Sci China Inf Sci, 2023, 66: 122104
34 Kim B, Lee J, Kang J, et al. HOTR: end-to-end human-object interaction detection with transformers. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 74–83
35 Yuan H, Li X, Yang Y, et al. PolyphonicFormer: unified query learning for depth-aware video panoptic segmentation. In: Proceedings of European Conference on Computer Vision, 2022. 582–599
36 Lan M, Zhang J, He F, et al. Siamese network with interactive transformer for video object segmentation. In: Proceedings of the 36th AAAI Conference on Artificial Intelligence, 2022. 1228–1236
37 Ru L, Zhan Y, Yu B, et al. Learning affinity from attention: end-to-end weakly-supervised semantic segmentation with transformers. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 16825–16834
38 Peng Z, Huang W, Gu S, et al. Conformer: local features coupling global representations for visual recognition. In: Proceedings of IEEE/CVF International Conference on Computer Vision, 2021. 357–366
39 Chen Y, Dai X, Chen D, et al. Mobile-former: bridging mobileNet and transformer. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 5260–5269
40 Huang M, Zhang L. Atrous pyramid transformer with spectral convolution for image inpainting. In: Proceedings of ACM International Conference on Multimedia, 2022. 4674–4683
41 Wu H, Xiao B, Codella N, et al. CvT: introducing convolutions to vision transformers. In: Proceedings of IEEE/CVF International Conference on Computer Vision, 2021. 22–31
42 Zhang L, Zhang L. Artificial intelligence for remote sensing data analysis: a review of challenges and opportunities. IEEE Geosci Remote Sens Mag, 2022, 10: 270–294
43 Song Y N, Gao L, Li X Y, et al. A novel vision-based multi-task robotic grasp detection method for multi-object scenes. Sci China Inf Sci, 2022, 65: 222104
44 Zhang K J, Zhang R, Wu Y L, et al. Few-shot font style transfer with multiple style encoders. Sci China Inf Sci, 2022, 65: 160109
45 Graham B, El-Nouby A, Touvron H, et al. LeViT: a vision transformer in convNet's clothing for faster inference. In: Proceedings of 2021 IEEE/CVF International Conference on Computer Vision, 2021. 12239–12249
46 Dai Z, Liu H, Le QV, et al. Coatnet: marrying convolution and attention for all data sizes. In: Proceedings of Advances in Neural Information Processing Systems, 2021. 34: 3965–3977
47 Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. In: Proceedings of Advances in Neural Information Processing Systems, 2017. 5998–6008
48 Zhang Q, Xu Y, Zhang J, et al. ViTAEv2: vision transformer advanced by exploring inductive bias for image recognition and beyond. Int J Comput Vis, 2023, 131: 1141–1162
49 Wang J, Li W, Zhang M, et al. Remote-sensing scene classification via multistage self-guided separation network. IEEE Trans Geosci Remote Sens, 2023, 61: 1–12
50 Zhang M, Li W, Zhang Y, et al. Hyperspectral and LiDAR data classification based on structural optimization transmission. IEEE Trans Cybern, 2022, 53: 3153–3164