Task Prior Attention Network for Multi-Task Learning of Dense Prediction

Anonymous submission
Paper ID 2279

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 end this issue, 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 MLPs to capture the task interaction. Third, the proposed cross attention module is applied to query task-specific feature maps with task-relevant prior information via self-attention. Our method allows compatibility with different backbones. Our TPANet (with Swin-L) performance surpasses the previous state-of-the-art by a large margin of +4.5 mIoU on NYUD-v2 dataset, and +1.4 mIoU on PASCAL-Context dataset, demonstrating the potential of our method as a robust MTL model. The code and models will be available.

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) [Vandenhende *et al.*, 2021] 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 works that apply them in various computer vision tasks.

Convolutional Neural Networks (CNNs) [Sun *et al.*, 2019] 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 [Misra *et al.*, 2016; Xu *et al.*, 2018; Gao *et al.*, 2019; Ling *et al.*, 2020; Bruggemann *et al.*, 2021]. However, these CNN-based MTL methods tend to only focus on the locality visual information, neglecting the global information. Recently, the Transformer-based methods [Dosovitskiy *et al.*, 2021; Wang *et al.*, 2022; Liu *et al.*, 2021; Jack *et al.*, 2021; Bhattacharjee *et al.*, 2022] 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 [Liu *et al.*, 2019; Xu *et al.*, 2022b; Bhattacharjee *et al.*, 2022; Raychaudhuri *et al.*, 2022] capture the long-range dependency and global relationships of all tasks by stacking self-attention blocks. The typical Transformer-based MTL models, MulT [Bhattacharjee *et al.*, 2022] and MTFormer [Xu *et al.*, 2022a] develop a self-task attention framework via plain multi-head self-attention to learn effective feature maps for multiple vision task predictions. Adopting Swin Transformer [Liu *et al.*, 2021] as the backbone to generate multi-scale features, MulT [Bhattacharjee *et al.*, 2022] designs a decoder via a shared self-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 [Enze *et al.*, 2021; Liu *et al.*, 2021; Chen *et al.*, 2022b]. 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. In this paper, we aim 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 Fig. 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 Fig. 1 (b). Then, the task-relevant prior information is leveraged in Transformer via self-attention. **Second**, as shown in Fig. 1 (b), we design a non-shared decoder for each individual 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 (Fig. 2(a)), adaptive task mixing (Fig. 2(b)), and cross attention (Fig. 2(c)). Task prior extractor is proposed to focus on producing task-relevant prior information with inductive bias into Transformer architecture for each individual task. Such task-relevant prior information can facilitate task-dependent local features. Adaptive task interaction consists of spatial MLP and channel MLP for adaptive task interaction. The task-relevant prior information with the introduced inductive biases can be adopted to promote locality visual information for individual task. Adaptive task mixing is employed to learn 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 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 [Silberman *et al.*, 2012] and Pascal-Context [Chen *et al.*, 2014]. 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

Multi-task learning of dense prediction. The multi-task learning (MTL) [Bruggemann *et al.*, 2021] approaches can precisely 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 [Kendall *et al.*, 2018; Chen *et al.*, 2018; Sener and Koltun, 2018; Teichmann *et al.*, 2018] directly use the shared representation to perform all dense predictions simultaneously. Follow-up papers have improved how to perform the task interaction in MTL of dense prediction. [Xu *et al.*, 2018; Zhang *et al.*, 2019; Liu *et al.*, 2019; Gao *et al.*, 2019; Vandenhende *et al.*, 2020;

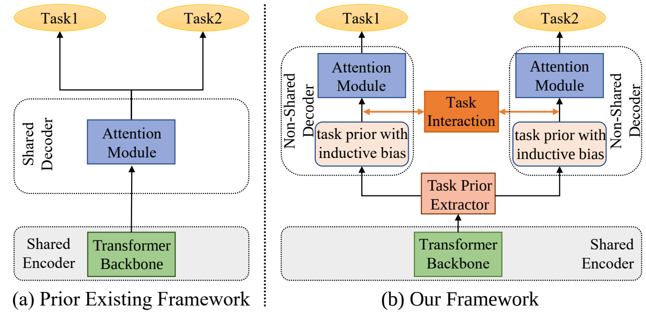

Figure 1: Previous Transformer-based MTL framework *v.s* Our Framework. (a) Previous Transformer-based MTL framework (*e.g.,* MTFormer) is designed by stacking plain self-attention. (b) We propose a Transformer-based MTL method to boost the MTL performance by introducing task-relevant prior information into the 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.

Gao *et al.*, 2020; Ling *et al.*, 2020; Bruggemann *et al.*, 2021; Xu *et al.*, 2022a; Bhattacharjee *et al.*, 2022; Xu *et al.*, 2022b; Raychaudhuri *et al.*, 2022] aim to use the interaction information between task to promote MTL performance. More recently, [Xu *et al.*, 2022a] a Transformer-based method proposed cross-task reasoning via a cross-task attention mechanism [Vaswani *et al.*, 2017] for further boosting the MTL results. Although the transformer-based frameworks have achieved the best performance in the multiple computer vision domain 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.

CNNs and Transformers. The inductive biases are hard-coded into the architecture of CNNs [Sun *et al.*, 2019] in the form of strong constraints on the locality and weight sharing [d'Ascoli *et al.*, 2021]. Vision Transformer [Dosovitskiy *et al.*, 2021] 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 [Liu *et al.*, 2021; Dosovitskiy *et al.*, 2021; Chen *et al.*, 2022b; Jack *et al.*, 2021], object detection [Wang *et al.*, 2022; Bumsoo *et al.*, 2021], semantic segmentation [Yuan *et al.*, 2022; Lan *et al.*, 2022; Ru *et al.*, 2022], *etc.* To jointly model global and local information, the methods [Peng *et al.*, 2021; Chen *et al.*, 2022a] employ the parallel individual convolution and transformer branches, while inductive biases from convolutions are introduced into Transformers [Graham *et al.*, 2021; Dai *et al.*, 2021; Wu *et al.*, 2021]. 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 and task extractor modules to encode task-relevant prior information with the convolutional inductive bias into task-specific feature maps. Experimental results confirm that introducing task-relevant prior information with inductive biases can

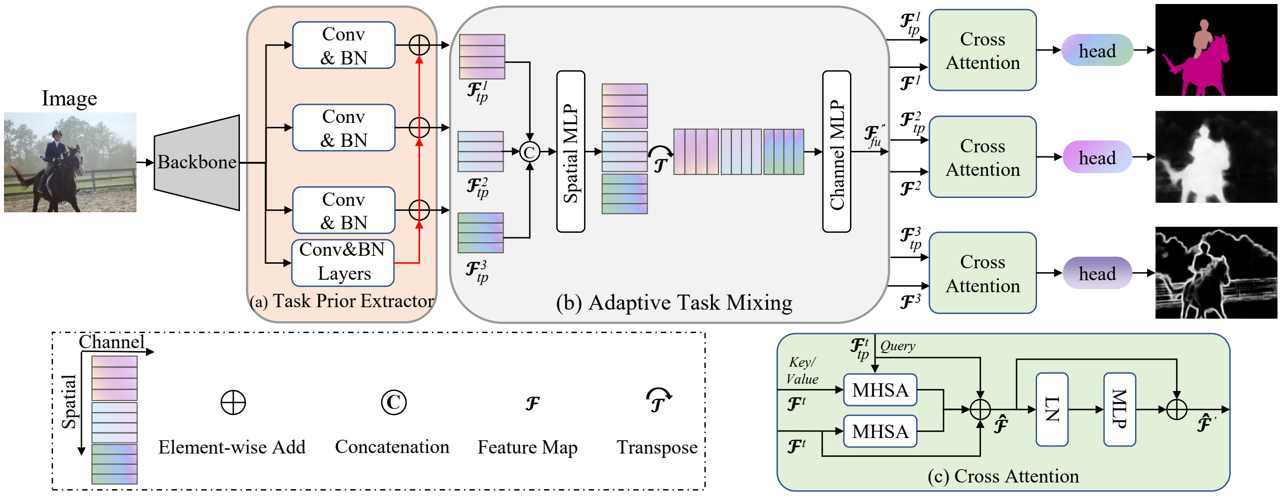

Figure 2: Illustration of the TPANet framework. Our TPANet consists of three key designs: (a) task prior extractor, (b) adaptive task mixing and (c) cross attention. We first process an image by backbone to generate feature maps. (a) Task prior extractor provides task-relevant prior information from convolution. The outputs of the task prior extractor are concatenated along channel dimension before passing them through (b) adaptive task mixing. We adapt the adaptive task mixing via spatial and channel MLPs 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.

reach higher performance in MTL of dense prediction.

3 Approach

3.1 Overall Architecture

The framework of our TPANet is summarized in Fig. 2. In the following, we first introduce how we capture the task-relevant prior information with inductive biases and their respective characteristics (Sec. 3.2, Fig. 2(a)). Then, we introduce our spatial MLP and channel MLP in the adaptive task mixing module for adaptive task interaction (Sec. 3.3, Fig. 2(b)). Finally, we show how we leverage a cross attention module for querying task-specific feature maps (Sec. 3.4, Fig. 2(c)).

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 then use the four-stage features are up-sampled to the same resolution and then concatenated along the channel dimension, with the resolution kept at $\mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times C}$ ($i.e.$, $\mathbf{x}$), where H, W, and C are the height, width, and channel of the image feature, respectively. The image feature from the backbone is then used by the task-specific heads to perform the dense predictions for every task.

We use the baseline and proposed modules for our TPANet multi-task method. Our TPANet contains three tailored designs, including (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) cross attention module for querying the task-specific feature map with task-relevant information. Finally, we obtain multiple feature maps according to task number, which can be used to conduct dense prediction tasks.

3.2 Task Prior Extractor Module

We design the task prior extractor to produce the task-relevant prior information with inductive bias from convolution.
Conv & BN block. The feature map $\mathbf{x}$ is fed into the task prior extractor module. As shown in Fig 2 (a), the number of the *Conv & BN* block is according to the task numbers in the task prior extractor module. We leverage a 1×1 convolution with batch normalization (Norm) to obtain a task-specific feature map $\mathcal{F}_{te}^t$ ($t \in [1, T]$, t indicates task number) with task-relevant prior information for each task. This procedure can be written:

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

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 feature map channel dimension.
Conv & BN Layers block. We design the *Conv & BN Layers* block to introduce the task-relevant prior information and locality inductive biases from multiple convolution layers into TPANet. Specifically, 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 $\mathbf{x}$ is fed into a *Conv & BN Layers* block to extract the inductive biases ((*Layer* = 1 in practice)), $i.e.$,

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

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

3.3 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 Fig. 2(b). The adaptive task mixing module consists of spatial MLP and channel MLP, 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 layer norm operation. The σ is a nonlinearity function (GELU).

Spatial MLP As shown in Fig. 2 (b), we first perform the spatial MLP. Spatial MLP 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 MLP 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; the $\mathcal{F}'_{fu} \in \mathbb{R}^{S \times TD}$.

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

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

where the $\mathcal{F}''_{fu} \in \mathbb{R}^{S \times TD}$.

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

3.4 Cross Attention Module

We follow [Vaswani *et al.*, 2017] to multi-head self-attention (MHSA) in computing similarity:

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

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.

The $\mathcal{F}^t_{tp}$ and $\mathcal{F}^T$ are then processed by a cross attention module to generate the task-specific feature map. As shown in Fig. 2(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 & value* from Eq. 7. We then develop query-based self-attention:

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

in which the $\mathcal{F}^t_{tp}$ is applied as $query$ from Eq. 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 Eq. 9 and 10 in the cross attention module. We use Element-wise adds, represented as:

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

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 Fig. 2, 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.

3.5 Overall Loss Functions

The overall TPANet loss $\mathcal{L}$ 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 the supplementary material subsection A.2 for loss details and hyper-parameters.

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 semantic segmentation (SemSeg), depth estimation (Depth), surface normal estimation (Normal) and boundary detection (Bound) tasks by providing dense labels for every image. There are four evaluation metrics to evaluate our model with other prior multi-task models: 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.

PASCAL-Context Dataset and Metrics. 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. There are five evaluation metrics to compare our model with other multi-task models: mean Intersection over Union (mIoU) for the SemSeg and PartSeg tasks, mean Error (mErr) for the Normal task, optimal dataset scale F-measure (odsF) for the Bound task, and maximum F-measure (maxF) for the Sal 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: the higher is the better.

Implementation Details. We conduct experiments on two publicly popular MTL datasets, NYUD-v2 [Silberman *et al.*, 2012] and PASCAL-Context [Chen *et al.*, 2014]. For all experiments, we use CNN-based architectures (*i.e.,* HRNetV2p-W18-Small (HRNet18) [Sun *et al.*, 2019], hrnetv2p-w48

Table 1: We report the comparison of the MTL models with the state-of-the-art on NYUD-v2 dataset. '↓': lower is better. '↑': higher is better. Δ_m denotes the average per-task performance drop. Swin-$\diamond$ indicates that the specific Swin model is uncertain.

Model	Backbone	Params (M)	GFLOPs (G)	SemSeg (mIoU)↑	Depth (rmse)↓	Normal (mErr)↓	Bound (odsF)↑	Δ_m[%]↑
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
TPANet (Ours)	HRNet18	5.18	27.02	39.31	0.5937	20.41	76.39	2.17
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
InvPT[Ye and Xu, 2022]	Swin-T	-	-	44.27	0.5589	20.46	76.10	2.59
TPANet (Ours)	Swin-T	32.02	113.03	46.49	0.5987	20.71	76.90	2.7
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
MTFormer[Xu *et al.*, 2022a]	Swin-$\diamond$	64.03	117.73	50.56	0.4830	-	-	4.12
TPANet (Ours)	Swin-S	53.34	133.38	50.87	0.5608	20.06	78.20	3.18

(HRNet48)) and Transformer-based architectures (*i.e.,* Swin-Tiny (Swin-T), Swin-Small (Swin-S), Swin-Base (Swin-B), Swin-Large (Swin-L) [Liu *et al.*, 2021]) as our backbone for TPANet, respectively. Our models are optimized using 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 [Sun *et al.*, 2019] and Swin Transformer [Liu *et al.*, 2021] in our case. 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 Tab. 1 and 2, we list the single-task and multi-task performance using different backbones on multiple vision tasks. See the supplementary material subsection A.1 for additional details.

4.2 Results

Results on 4-task NYUD-v2. In Tab. 1, we first report the four task results in different metrics on NYUD-v2 dataset. We also provide a quantitative evaluation of the computational cost (GFLOPs) and parameters. Tab. 1 shows a comparison with the state-of-the-art approaches. Following [Bruggemann *et al.*, 2021], we use the same backbone and training setting for a fair comparison. We find that TPANet model outperforms InvPT in terms of multi-tasking performance (Ours 2.7 *v.s* InvPT 2.59). When equipped with Swin-S as the backbone, the TPANet achieves comparable performance at 50.87 mIoU with a significant parameter (53.34M). Concretely, our TPANet model outperforms the previous best by +0.31 (Ours 50.87 v.s 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. In addition, as shown in Tab. 5, our TPANet method with Swin-L is superior to the state-of-the-art Transformer-based MTL method [Ye and Xu, 2022] in all vision tasks. Our method achieves the highest performance 56.35 mIoU on the SemSeg task on NYUD-v2 in Tab. 5. 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 Tab. 1 and 5, our TPANet benefits from the advantages of both task-relevant prior information and query-based Transformer that shows strong performance on all the metrics.

Results on 5-task PASCAL-Context. As shown in Tab. 2, we further evaluate our method on 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 [Sun *et al.*, 2019], Swin-T, Swin-S and Swin-B [Liu *et al.*, 2021]. 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 on PASCAL-Context. The results show that our TPANet is relatively robust to vary-

Table 2: We report a comparison of the MTL models with the state-of-the-art on PASCAL-Context dataset. '↓': lower is better. '↑': higher is better. 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 [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
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[Xu *et al.*, 2022b]	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[Xu *et al.*, 2022b]	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[Xu *et al.*, 2022a]	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

Table 3: Ablation studies and analysis 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. '↓': lower is better. '↑': higher is better. '**w/**' indicates **"with"**.

(a) Ablation on modules

Model	SemSeg (mIoU)↑	Depth (rmse)↓	Normal (mErr)↓	Bound (odsF)↑
baseline	38.78	0.6312	21.05	75.6
w/TPE	43.44	0.6124	20.83	76.4
w/TPE+ATM	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

Table 4: Ablation on the dropout (κ). We perform this ablation using Swin-T as the backbone on 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

Table 5: NYUD-v2 performance comparison, using Swin-B/L. We compare our model with the InvPT [Ye and Xu, 2022].

Method	Backbone	SemSeg (mIoU)↑	Depth (rmse)↓	Normal (mErr)↓	Bound (odsF)↑
Multi-task baseline	Swin-B	51.44	0.5813	20.44	77.0
InvPT[Ye and Xu, 2022]	Swin-B	50.97	0.5071	19.39	77.3
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[Ye and Xu, 2022]	Swin-L	51.76	0.5020	19.39	77.6
Ours	Swin-L	56.35	0.5019	19.02	77.9

ing CNN-based and Transformer-based backbones.

4.3 Ablation Studies

To investigate any likelihood type benefits from our framework, we conduct the 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 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 Tab. 3a, our model achieves strong accuracy performance when equipped with the task prior extractor module. We find 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 ATM can effectively adapt to task interactions along spatial and chan-

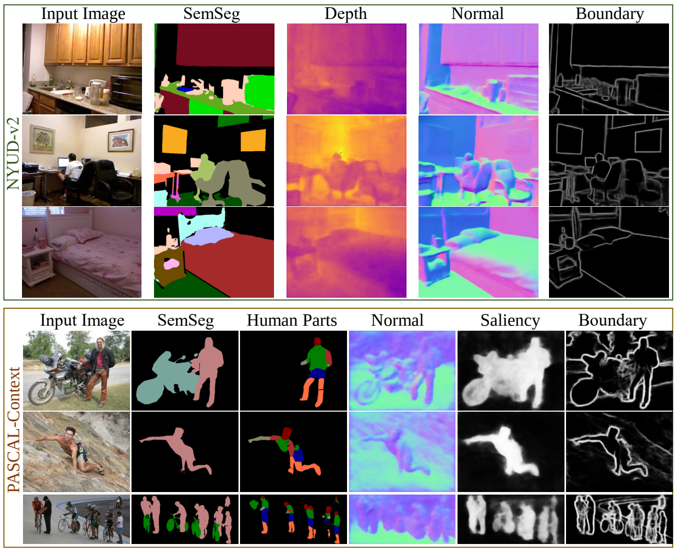

Figure 3: Qualitative results of our TPANet using Swin-S as the backbone on two datasets. The first two rows of the visualization illustrate three examples from the NYUD-v2 dataset. The visualizations in the last box also illustrate three examples from the PASCAL-Context dataset.

nel dimensions. Further, the non-shared cross attention is designed to be suitable for multiple vision scenarios.

Ablation on the scales. Tab. 3b lists the experimental results, showing that the performance can be consistently improved with the value of the scale number. 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.

Ablation on the dropout number. We test TPANet with different dropout numbers, listed in Tab. 4. In the cross attention module, dropout operations exist for MLP in cross attention module. To explore the impact of the number of dropouts in our model, we set the dropout number $\kappa \in \{0, 0.1, 0.2, 0.3\}$.

Ablation on different backbones. In Tab. 5, we further compare our TPANet against more standard multi-task baselines and InvPT [Ye and Xu, 2022], 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 [Ye and Xu, 2022]. Specifically, our TPANet method further outperforms the standard multi-task baselines and InvPT [Ye and Xu, 2022] 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 be achieved performance leadership on different backbones on NYUD-v2 dataset.

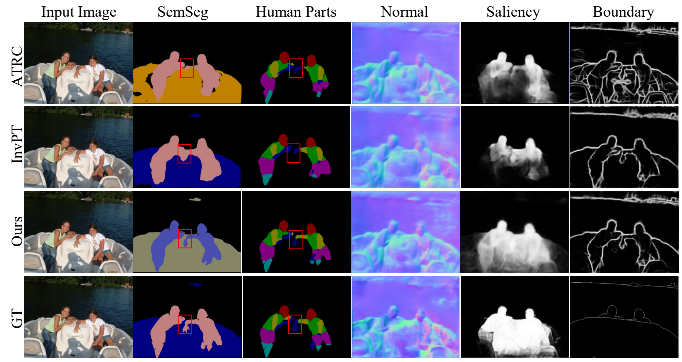

Figure 4: Qualitative results of our TPANet compare with the previous MTL methods (*i.e.,* ATRC and InvPT) on PASCAL-Context dataset. The first two rows of the visualization illustrate. The visualizations (notice the red boxes) emphasize the accuracy and efficiency of our TPANet in multiple vision tasks. From top to bottom: ATRC [Bruggemann *et al.*, 2021], InvPT [Ye and Xu, 2022], TPANet (Ours) and ground truth (GT).

4.4 Visualization

To further analyze the property of our method, we show the visualizations for qualitative comparison in Fig. 3 on two datasets. We observe that our TPANet gives overall better visualizations than the baseline model, including the whole tasks, as shown in Fig. 4. For the segmentation task, in Fig. 4, we observe that TPANet obtains more precise semantic segmentation and human parts segmentation. Specifically, comparing ATRC and InvPT with our TPANet in the first and second columns, we can see that ATRC and InvPT fail to distinguish the arms and hands of the two people. We use red boxes to mark the exact locations and quickly find the segmentation differences between the three methods. While our TPANet successfully differentiates the two objects, suggesting ours learn more semantic features. Fig. 3 and Fig. 4 show that the effective, efficient and robust of our TPANet model allows for predicting multiple tasks with strong expressive power, successfully conducting the MTL of dense prediction. See the supplementary material for more visualizations.

5 Conclusion

In this paper, we explore the inductive biases effect in Transformer-based MTL architecture, named TPANet, to effectively and efficiently perform dense predictions. To boost the MTL performance, we introduce task-relevant prior information with inductive biases to Transformer-based architecture to increase locality information for dense prediction. 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 demonstrate the effectiveness, efficiency, and robustness of our method.

Limitation and future work. One limitation of our method: We observe that some tasks do not require more parameters to achieve good results.A crucial future exploration is to develop a learnable gate to plan the parameter for each task.

References

[Bhattacharjee *et al.*, 2022] Deblina Bhattacharjee, Tong Zhang, Sabine Süsstrunk, and Mathieu Salzmann. Mult: An end-to-end multitask learning transformer. In *CVPR*, pages 12031–12041, 2022.

[Bruggemann *et al.*, 2021] David Bruggemann, Menelaos Kanakis, Anton Obukhov, Stamatios Georgoulis, and Luc Van Gool. Exploring relational context for multi-task dense prediction. In *ICCV*, pages 15869–15878, 2021.

[Bumsoo *et al.*, 2021] Kim Bumsoo, Lee Junhyun, Kang Jaewoo, Kim Eun-Sol, and Kim Hyunwoo J. Hotr: End-to-end human-object interaction detection with transformers. *CVPR*, 2021.

[Chen *et al.*, 2014] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *CVPR*, pages 1971–1978, 2014.

[Chen *et al.*, 2018] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *ICML*, pages 794–803, 2018.

[Chen *et al.*, 2022a] Yinpeng Chen, Xiyang Dai, Dongdong Chen, Mengchen Liu, Xiaoyi Dong, Lu Yuan, and Zicheng Liu. Mobile-former: Bridging mobilenet and transformer. In *CVPR*, pages 5270–5279, 2022.

[Chen *et al.*, 2022b] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*, 2022.

[Dai *et al.*, 2021] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *NeurIPS*, 34:3965–3977, 2021.

[Dosovitskiy *et al.*, 2021] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[d'Ascoli *et al.*, 2021] Stéphane d'Ascoli, Hugo Touvron, Matthew L Leavitt, Ari S Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In *ICML*, pages 2286–2296, 2021.

[Enze *et al.*, 2021] Xie Enze, Wang Wenhai, Yu Zhiding, Anandkumar Anima, M Alvarez Jose, and Luo. Ping. Segformer: Simple and efficient design for semantic segmentation with transformers. *NeurIPS*, 34, 2021.

[Gao *et al.*, 2019] Yuan Gao, Jiayi Ma, Mingbo Zhao, Wei Liu, and Alan L Yuille. Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction. In *CVPR*, pages 3205–3214, 2019.

[Gao *et al.*, 2020] Yuan Gao, Haoping Bai, Zequn Jie, Jiayi Ma, Kui Jia, and Wei Liu. Mtl-nas: Task-agnostic neural architecture search towards general-purpose multi-task learning. In *CVPR*, pages 11543–11552, 2020.

[Graham *et al.*, 2021] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. In *ICCV*, pages 12259–12269, 2021.

[Jack *et al.*, 2021] Lanchantin Jack, Wang Tianlu, Ordonez Vicente, and Qi Yanjun. General multi-label image classification with transformers. *CVPR*, 2021.

[Kendall *et al.*, 2018] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, pages 7482–7491, 2018.

[Lan *et al.*, 2022] Meng Lan, Jing Zhang, Fengxiang He, and Lefei Zhang. Siamese network with interactive transformer for video object segmentation. *AAAI*, 2022.

[Ling *et al.*, 2020] Zhou Ling, Cui Zhen, Xu Chunyan, Zhang Zhenyu, Wang Chaoqun, Zhang Tong, and Yang Jian. Pattern-structure diffusion for multi-task learning. In *CVPR*, pages 4514–4523, 2020.

[Liu *et al.*, 2019] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *CVPR*, pages 1871–1880, 2019.

[Liu *et al.*, 2021] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021.

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

[Peng *et al.*, 2021] Zhiliang Peng, Wei Huang, Shanzhi Gu, Lingxi Xie, Yaowei Wang, Jianbin Jiao, and Qixiang Ye Conformer. Local features coupling global representations for visual recognition. in 2021 ieee. In *ICCV*, pages 357–366, 2021.

[Raychaudhuri *et al.*, 2022] Dripta S Raychaudhuri, Yumin Suh, Samuel Schulter, Xiang Yu, Masoud Faraki, Amit K Roy-Chowdhury, and Manmohan Chandraker. Controllable dynamic multi-task architectures. In *CVPR*, 2022.

[Ru *et al.*, 2022] Lixiang Ru, Yibing Zhan, Baosheng Yu, and Bo Du. Learning affinity from attention: End-to-end weakly-supervised semantic segmentation with transformers. In *CVPR*, pages 16846–16855, 2022.

[Sener and Koltun, 2018] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *NeurIPS*, 31, 2018.

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

[Sun *et al.*, 2019] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, pages 5693–5703, 2019.

[Teichmann *et al.*, 2018] Marvin Teichmann, Michael Weber, Marius Zoellner, Roberto Cipolla, and Raquel Urtasun. Multinet: Real-time joint semantic reasoning for autonomous driving. In *IV*, pages 1013–1020, 2018.

[Vandenhende *et al.*, 2020] Simon Vandenhende, Stamatios Georgoulis, Luc Van Gool, and Luc Van Gool. Mti-net: Multi-scale task interaction networks for multi-task learning. In *ECCV*, pages 527–543, 2020.

[Vandenhende *et al.*, 2021] S. Vandenhende, S. Georgoulis, W. Van Gansbeke, M. Proesmans, D. Dai, and L. Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE TPAMI*, 2021.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017.

[Wang *et al.*, 2022] Wen Wang, Yang Cao, Jing Zhang, and Dacheng Tao. FP-DETR: Detection transformer advanced by fully pre-training. In *ICLR*, 2022.

[Wu *et al.*, 2021] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *ICCV*, pages 22–31, 2021.

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

[Xu *et al.*, 2022a] Xiaogang Xu, Hengshuang Zhao, Vibhav Vineet, Ser-Nam Lim, and Antonio Torralba. Mtformer: Multi-task learning via transformer and cross-task reasoning. In *ECCV*, pages 304–321, 2022.

[Xu *et al.*, 2022b] Yangyang Xu, Xiangtai Li, Haobo Yuan, Yibo Yang, Jing Zhang, Yunhai Tong, Lefei Zhang, and Dacheng Tao. Multi-task learning with multi-query transformer for dense prediction. *arXiv preprint arXiv:2205.14354*, 2022.

[Ye and Xu, 2022] Hanrong Ye and Dan Xu. Inverted pyramid multi-task transformer for dense scene understanding. In *ECCV*, 2022.

[Yuan *et al.*, 2022] Haobo Yuan, Xiangtai Li, Yibo Yang, Guangliang Cheng, Jing Zhang, Yunhai Tong, Lefei Zhang, and Dacheng Tao. Polyphonicformer: Unified query learning for depth-aware video panoptic segmentation. In *ECCV*, 2022.

[Zhang *et al.*, 2019] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan, Nicu Sebe, and Jian Yang. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *CVPR*, pages 4106–4115, 2019.