BridgeNet: Comprehensive and Effective Feature Interactions via Bridge Feature for Multi-task Dense Predictions

Jingdong Zhang, Jiayuan Fan*, Peng Ye, Bo Zhang, Hancheng Ye, Baopu Li, Yancheng Cai, *Tao Chen, Senior Member, IEEE*

Abstract—Multi-task dense prediction aims at handling multiple pixel-wise prediction tasks within a unified network simultaneously for visual scene understanding. However, cross-task feature interactions of current methods are still suffering from incomplete levels of representations, less discriminative semantics in feature participants, and inefficient pair-wise task interaction processes. To tackle these under-explored issues, we propose a novel BridgeNet framework, which extracts comprehensive and discriminative intermediate Bridge Features, and conducts interactions based on them. Specifically, a Task Pattern Propagation (TPP) module is firstly applied to ensure highly semantic task-specific feature participants are prepared for subsequent interactions, and a Bridge Feature Extractor (BFE) is specially designed to selectively integrate both high-level and low-level representations to generate the comprehensive bridge features. Then, instead of conducting heavy pair-wise cross-task interactions, a Task-Feature Refiner (TFR) is developed to efficiently take guidance from bridge features and form final task predictions. To the best of our knowledge, this is the first work considering the completeness and quality of feature participants in cross-task interactions. Extensive experiments are conducted on NYUD-v2, Cityscapes and PASCAL Context benchmarks, and the superior performance shows the proposed architecture is effective and powerful in promoting different dense prediction tasks simultaneously.

Index Terms—Multi-task Learning, Dense Prediction, Representation Learning.

1 Introduction

Dense prediction tasks that predict the pixel-wise label for an image have received much attention in many fields such as self-driving [22], [25], surveillance [4], [20], *etc*. With the aid of Convolution Neural Networks (CNNs), a large number of dense prediction works have made great progress recently, including pose estimation [8], [31], [36], [45], semantic segmentation [1], [6], [23], [27], [33], [56], and depth estimation [12], [13], [19], [50]. However, these works mainly focus on one specific task and the high relevance between different dense prediction tasks is under-explored.

- *Jingdong Zhang is with the School of Information Science and Technology, Fudan University, Shanghai 200433, China, and also with the Department of Computer Science, Texas A&M University, College Station, TX 77843 USA (e-mail: jdzhang@tamu.edu).*
- *Jiayuan Fan is with the Academy for Engineering and Technology, and Peng Ye, Hancheng Ye, Yancheng Cai, Tao Chen are with the School of Information Science and Technology, Fudan University, Shanghai 200433, China (Corresponding author: Jiayuan Fan, E-mail: jyfan@fudan.edu.cn).*
- *Bo Zhang is with the Shanghai AI Laboratory, Shanghai 200232, China (e-mail: bo.zhangzx@gmail.com).*
- *Baopu Li is an independent researcher, 100 Oracle Pkwy, Redwood City, CA 94065 (E-mail: Bpli.cuhk@Gmail.com).*

- *This work is supported by National Key Research and Development Program of China (No. 2022ZD0160101), Shanghai Natural Science Foundation (No. 23ZR1402900), Shanghai Science and Technology Commission Explorer Program Project (24TS1401300), Shanghai Municipal Science and Technology Major Project (No.2021SHZDZX0103). The computations in this research were performed using the CFFF platform of Fudan University.*

Multi-task Learning (MTL) aims to learn a model that handles multiple different tasks simultaneously. Recently, deep learning-based MTL methods [15], [18], [24], [39], [51], [55] have achieved great progress in multiple dense prediction tasks. By sharing parameters of the heavy image encoders, MTL methods can also achieve joint training and inference of different tasks with high efficiency. Besides, the high relevance among different dense prediction tasks can be further explored by designing a task-interactive module, which leads to mutual boosting of multi-task performance.

According to where task interactions happen, existing multi-task works are roughly divided into encoder-focused and decoder-focused methods [38]. In particular, the encoder-focused methods conduct interactions of different tasks in the encoding stage, as shown in Fig. 1 (a), task-specific features interact directly with task-generic features produced by the shared backbone, to gain knowledge from multi-task shared representations and thus achieve indirect cross-task interactions. Since the image backbone/encoder is usually initialized with pre-trained parameters that contain abundant visual prior knowledge, and are supervised by gradients from all tasks during the training, it can produce stronger representation with rich **low-level representations**, i.e. fundamental image details such as edges, corners, textures, etc., as shown in the right legend of Fig. 1. For pixel-level dense prediction tasks like semantic segmentation, these representations are essential in gaining basic pixel relations and forming accurate semantic masks. However, for joint dense predictions among multiple tasks, these low-level representations are task-generic, which means they only contain general image structures,

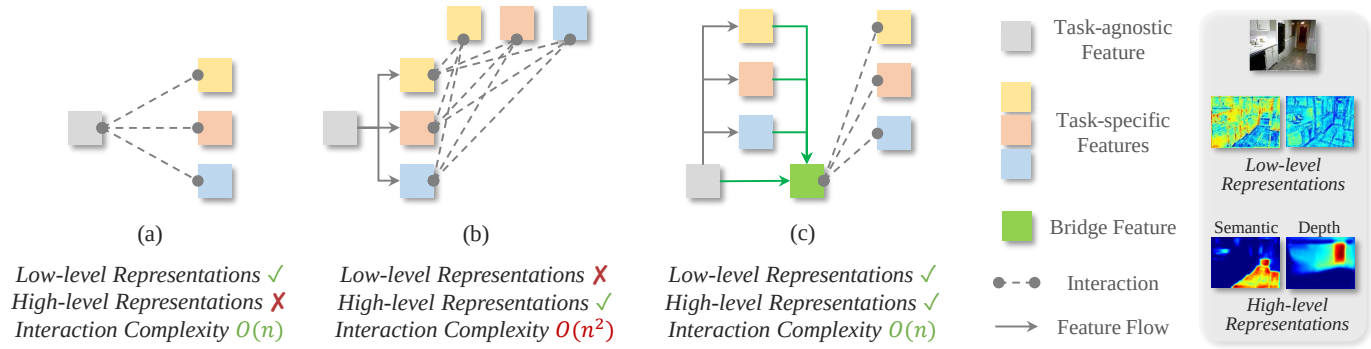


Fig. 1. Illustration of multi-task interaction strategies. (a) Encoder-focused [2], [15], [16], [24], [28], [30], [37]: the task-specific features directly interact with the task-generic features from the common backbone. The task-generic features contain rich low-level representations but lack high-level representations. The interaction complexity is $O(n)$. (b) Decoder-focused [3], [10], [49], [51], [55], [57]: the task-specific features are firstly produced by deep supervision, and then interactions are conducted only based on task-specific features, which has discriminative high-level representations, but low-level representations are absent. The pair-wise interaction complexity is $O(n^2)$. (c) BridgeNet (ours): the bridge features are produced from both task-generic and task-specific features and have both rich high-level and low-level representations. The subsequent interactions based on bridge features only have $O(n)$ complexity. On the right part of the legend, we show some examples of representations from different levels in the feature map: the **low-level representations** contain less discriminative task information but are rich in details like boundaries, corners and textures, on the contrary, the **high-level representations** are smooth with less image details, but highly correlated to the task properties, like the highlighted floor with semantic, or areas with larger distance from the depth sensor.

but lack distinguishable semantics that relate to each specific task. Directly conducting interactions based on task-generic features is sub-optimal, since the mining of discriminative task-specific features is insufficient and consequently limits the role of cross-task interaction [38].

To alleviate the above issue, the decoder-focused methods conducting interactions in the decoding stage have been developed, where discriminative **high-level representations**, i.e. abstract and semantic representations derived from images such as key-points, objects, geometries, etc., is perceived from ground-truth labels to facilitate feature interactions, as shown in the right legend of Fig. 1. Specifically, the initial task predictions are firstly produced by preliminary decoders with deep supervision in the early decoding stage [39], [49], [51], [55], [57], aiming to decouple the generic encoder features and discover task-specific high-level representations according to the corresponding labels. Subsequently, the cross-task interactions are conducted based on high-level representations by modules such as multi-modal distillation [49] or the ATRC [3]. These methods try to model task-pair relations, as shown in Fig. 1 (b), by extracting and transferring useful task knowledge from one task to another based on task-specific features, most of them achieve better performance compared with encoder-focused methods.

Despite the promising performance improvement, these previous methods still face challenges:

i) Firstly, the cross-task interactions are suffering from incomplete levels of representations in the participants, i.e. either task-generic features with low-level representations or task-specific features with high-level representations are exploited in prior works, which leads to incomplete participants for feature interaction. Though decoder-focused methods have validated that the interactions with discriminative task-specific features are more effective, in which high-level representations are gained in the initial decoding stage. However, the essential low-level representations with abundant basic image details gradually disappear since they are not directly required by supervision signals, and subsequently, in the following task interaction stage, only high-level representations are involved while low-level representations that produce essential dense prediction priors among different tasks are absent, leading to incomplete task interactions and limiting the model performance.

ii) Secondly, the cross-task interactions are also suffering from less discriminative semantics in the participants. We observe that the task-specific features produced by the preliminary decoders are usually of low quality with less discriminative representations (as shown in Fig 3 right), which negatively affects the subsequent interactions based on them. By analyzing the **task patterns**, which are defined as the attention distributions or feature structures specifically related to a certain task (as shown in Fig. 3 left), we discovered that task patterns significantly differ on different tasks. While in the task-generic features produced by the shared encoder, patterns are usually highly entangled, which makes it difficult to decompose each specific task pattern, and simply imposing supervision on preliminary decoders is not able to fundamentally solve this task-pattern entanglement issue. Thus a more effective way of disentangling meaningful and semantically high-level representations from the shared task-generic features is required for producing high-quality interaction participants.

iii) The cross-task interaction way is inefficient with high extension costs. Here exemplified by Fig. 1 (b). Since previous decoder-focused models need to consider pair-wise task relations [3], [39], [49], [51], with the task number increasing, the complexity of task interactions will increase in $O(n^2)$, which is costly and limits the extension of previous methods on the settings with more tasks.

To this end, we focus on improving the interaction quality for superior Multi-task Learning from the above three aspects, and propose a novel BridgeNet framework, which is based on the interaction with a comprehensive intermediate feature, namely the **Bridge Feature**. As shown in Fig. 1 (c), to solve the first challenge, we propose to

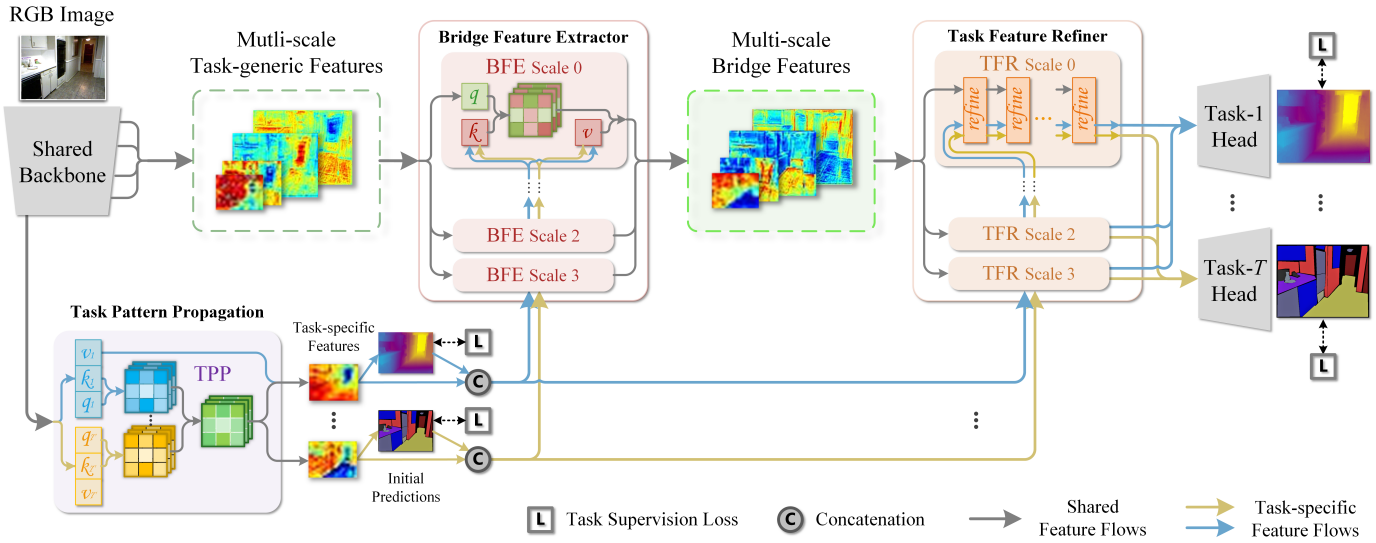


Fig. 2. The overview of our proposed method. We take depth estimation and semantic segmentation as examples. We use a shared image backbone to encode task-generic features, and a set of preliminary decoders with deep supervision are used for task-specific feature extraction. Different from decoder-focused methods, a TPP is added before the initial predictions are formed to tackle the task-pattern-entanglement issue. The produced high-quality task-specific features are not used for interaction directly, they are firstly processed by BFE at each scale to gain low-level representations and form the bridge features. Then, the multi-scale bridge features are used to gradually refine task-specific features by TFR. The outputs of TFR are aggregated and fed into task-specific heads for final predictions. The detailed structures of TPP (Sec. 3.2, Fig. 4 (a)), BFE (Sec. 3.3, Fig. 5) and TFR (Sec. 3.4, Fig. 4 (b)) will be given later.

extract and take advantage of the bridge feature which contains both rich low-level and high-level representations to ensure the integrity of feature interactions. To solve the second challenge, we propose to disentangle task patterns by jointly learning and distributing task patterns to corresponding tasks, in order to produce high-quality task-specific features serving as interaction participants. And to solve the third challenge, we propose to conduct interactions directly between the bridge features and each task-specific feature, which only involves $O(n)$ complexity of task interactions. Specifically, our proposed method consists of a shared encoder backbone for the task-generic features production, the preliminary decoders with Task Pattern Propagation (TPP) in the early decoding stage producing high-quality task-specific features and handling the task-pattern-entanglement issue. The specially designed Bridge Feature Extractor (BFE), which has a transformer-based structure with cross-attention. By globally querying task-specific features with the task-generic features at each scale, the BFE selects high-level representations correlated to every task and injects them into the task-generic features with rich low-level representations. Thus the extracted powerful bridge features contain comprehensive task representations in both high-level and low-level, and fertilize the formation of final predictions by the Task Feature Refiner (TFR). The overall bridge-feature-based multi-task framework is named as BridgeNet.

The main contributions of our work are three-fold:

1) We discover that incomplete, low-quality interaction participants and inefficient interaction processes exist in current cross-task interaction designs of multi-task dense prediction frameworks. To tackle these issues, we propose the novel BridgeNet framework, which performs a comprehensive interaction based on the bridge features containing both low-level and high-level representations. To the best of our knowledge, bridge features with comprehensive types of representations are involved in the multi-task dense prediction for the first time, which are exploited to transfer cross-task knowledge and fertilize the final task predictions.
2) A Task Pattern Propagation (TPP) module is firstly applied to avoid the task-pattern-entanglement issue and prepare high-quality participants for the subsequent interactions. A transformer-based Bridge Feature Extractor (BFE) is designed to extract bridge features from both task-generic and task-specific features for comprehensive interactions. Finally, a Task Feature Refiner (TFR) is applied to take advantage of bridge features and refine the final task predictions.
3) Our proposed method is extensively evaluated on various dense prediction tasks, including semantic segmentation, depth estimation, surface normal estimation, saliency estimation, and edge detection. The experimental results and insightful analyses on NYUD-v2 and PASCAL Context datasets demonstrate that the proposed architecture achieves superior performance over the state-of-the-art works. Our code is available at https://github.com/Evergreen0929/EEMTL.

2 Related Works

Multi-Task Learning Architectures: With the rapid development of deep CNNs, a lot of MTL works have achieved promising results [2], [15], [16], [18], [21], [24], [28], [30], [34], [37], [39], [47], [49], [51], [52], [54], [55], [57]. As mentioned in Sec. 1, existing MTL works are roughly divided

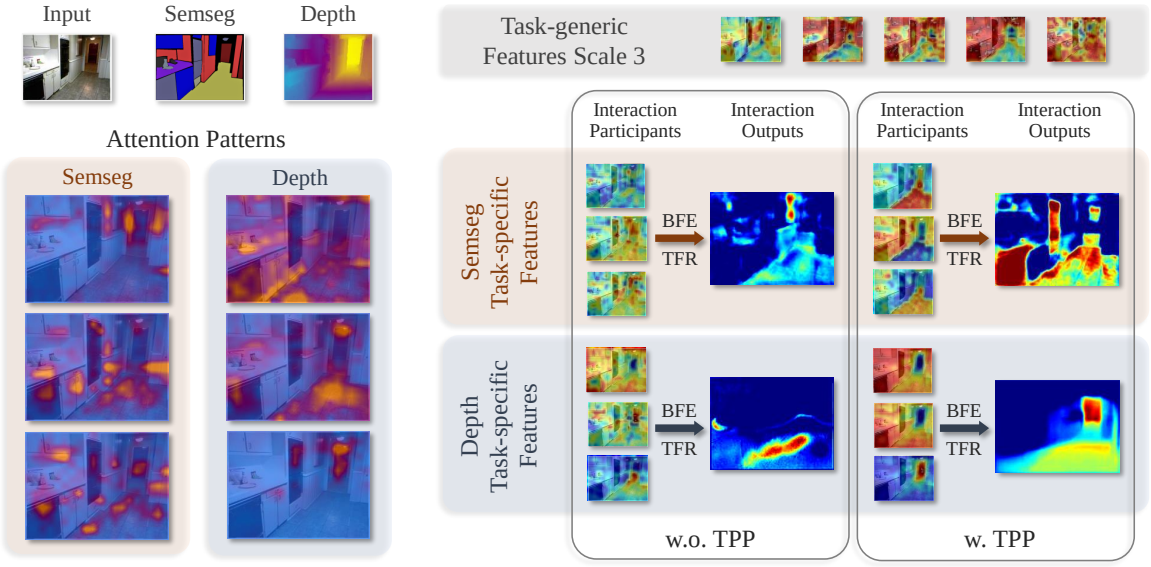


Fig. 3. **Left:** Visualization of different patterns of two tasks (semantic segmentation and depth estimation). The semantic feature focuses on objects with various semantic information, like cabinets and floors, while depth attention focuses more on the surface and edge features with geometry information. **Right:** Visualization of the task-specific features with (w.) and without (w.o.) TPP. The task-generic features produced by the shared encoder show the task-pattern-entanglement issue, which is significantly different from the distributions of task labels, and the contained patterns are implicit and ambiguous. Without TPP, the decomposed task-specific features serving as the interaction participants are still struggling with the lack of discriminative semantics, which negatively affects the subsequent interaction process (BFE, TFR) and eventually produces low-quality interaction outputs. However, with TPP, the task-specific features can obtain well-decomposed representations that have more similar distributions to their ground-truth labels (like the highlighted floor area in semantic segmentation and door area in depth estimation), and thus boosts the subsequent interaction process to produce high-quality discriminative features.

into encoder-focused and decoder-focused architectures. In addition, the sharing ways of network parameters are categorized into hard and soft parameter sharing [10], [38], which correspond to directly shared parameters or indirectly constrained by regularization or losses, respectively. The encoder-focused models in [15], [30], [34] have specifically designed shared structures to explore backbone task-generic features in the encoding stage, while the decoder-focused models in [38] adopt hard parameters sharing in the encoding stage to reduce redundant parameters and FLOPs, and specially focus on the interaction among different task-specific features in decoding stages. In this paper, our proposed method is based on the decoder-focused architecture, however, also takes advantage of the encoder-focused models, i.e. the exploring of task-generic features with rich low-level representations, and computational-friendly multi-task interaction ways.

Task Interaction Strategies in MTL: In MTL, the learned representation from one task may be beneficial for other tasks, so it is essential to design task interaction manners to promote mutual performances, and avoid potential information inconsistencies. In the field of knowledge distillation, a multi-task student network is distilled through several single-task teacher networks, which transfer multi-task knowledge to the student network and achieve good performance [21], [52]. However, this network requires to pretrain a group of teacher networks which costs a lot by increasing the number of tasks. The typical encoder-focused models [15], [24], [30] conduct interactions at the encoding stage by sharing parts of parameter spaces, especially, [24] designs attention modules for feature extraction. While another group of works further studies the parameters sharing scheme in the encoding stage and develops the Branched MTL [2], [16], [28], [37], aiming at manually or automatically determining the task-shared and task-specific branches to minimizing task inconsistencies. In contrast, decoder-focused models, such as PAD-Net [49], employ the multi-model distillation based on initial predictions by deep supervision at the decoding stage, where features of auxiliary tasks are extracted and transferred to target tasks to improve their performance. MTI-Net [39] argues that the information of different tasks only promotes each other at certain scales, and extends the multi-model distillation to the multi-scale fashion. PAP-Net [55] and PSD [57] apply the affinity and the interactions of tasks at pixel and pattern-structure levels respectively. ATRC [3] designs automatically searching for source-to-target task relations by adaptive task-relational context heads with Neural Architecture Search (NAS) techniques. InvPT [51] is the first transformer-based joint learning architecture of global spatial interaction and simultaneous all-task interaction. Although decoder-focused models have recently developed rapidly with a large number of variants, and achieve significant improvements, the interaction of them is only based on task-specific features. Besides, the idea of directly source-to-target task-pair relation modeling is inefficient. To tackle the information loss and high computational cost of these methods, we adopt a brand new BridgeNet architecture, which absorbs the advantages of both encoder- and decoder-focused methods, in order to maintain comprehensive task interaction along with acceptable resource cost.

Attention in MTL: Recently, many MTL architectures [3], [24], [39], [49], [51], [53], [55] have employed attention modules to dynamically select information among differ-

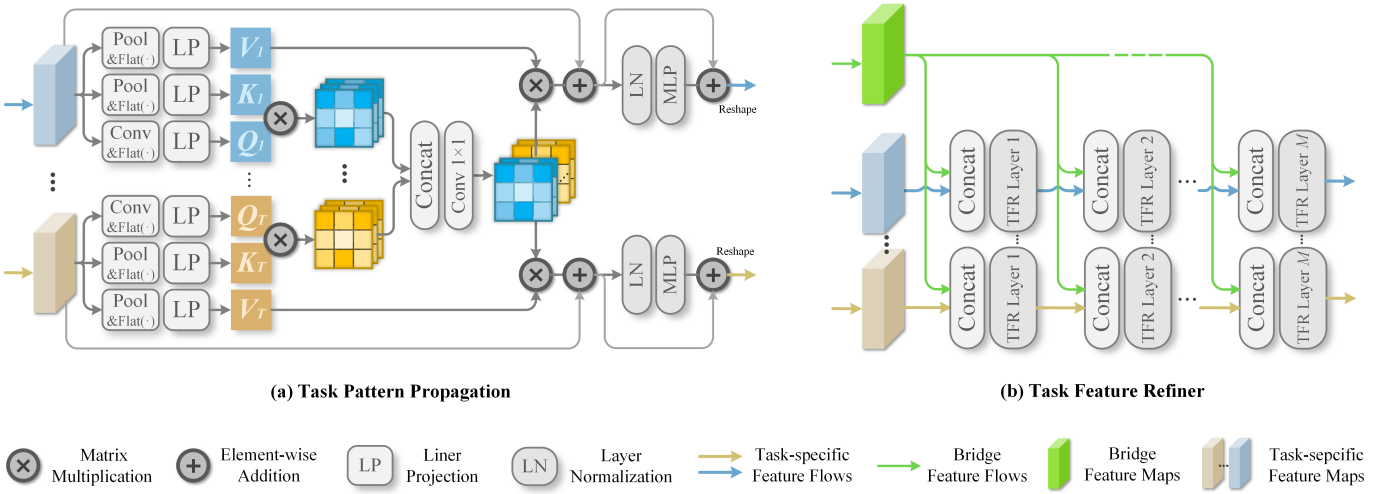


Fig. 4. (a) The structure of TPP which is applied after the last backbone layer. The propagation attention map shares the patterns for each task. (b) Task Feature Refiner (TFR), employs a cascaded structure that can be flexibly deployed within the decoding process to inject rich representations from bridge features into each task-specific feature. TFRs of different scales have similar structures and we illustrate the specific structure of one particular scale here.

ent tasks. In [24], attention is used to extract features of different tasks from the shared backbone. [39], [49] distill auxiliary tasks through the attention to extract useful features for target tasks. Since self-attention is capable of capturing global receptive fields by considering the global correlation in feature maps [14], [44], some MTL works directly introduce self-attention into networks [55]. However, these works do not further explore the distribution and interaction of attention among different tasks. With the development of Transformer [40] in vision [5], [11], [26], [43], [46], [48], Multi-head self-attention (MHSA) becomes popular for processing visual scenarios due to its long-range dependencies and context-capture capability. We further study the patterns of MHSA among different tasks and propose the Task Pattern Propagation (TPP) which produces discriminative task-specific features in the early decoding stage to solve the task-pattern-entanglement issue. The transformer-based BFE produces bridge features by global cross-attention between task-generic and task-specific features. Different from simply applying attention to MTL architectures, we focus on structural analysis of task features in the whole interaction process, and introduce priors (e.g. low-level image representations, task patterns) with special design accordingly.

3 The Proposed Method

3.1 Overview

The overview of the proposed BridgeNet is shown in Fig.2, which mainly consists of TPP for task-pattern disentanglement, BFE for bridge feature extraction, and TFR for task-specific feature refinement. In the early decoding stage, we gain multi-scale task-generic features $\mathbf{S} = \{S_i \mid i = 0, 1, 2, 3\}$ from the shared image backbone, which can be ConvNets (e.g. HRNet, ResNet) or Vision Transformers (e.g. ViT). The task-generic features are firstly partitioned into patch tokens and embedded into the decoding dimensions. Simultaneously, a set of preliminary decoders with deep supervision are implemented to produce task-specific features $\mathbf{P} = \left\{P_i^j \mid i = 0, 1, 2, 3; \quad j = 0, 1, \cdots, T\right\}$, where T represents the total number of tasks. The TPP is applied before the feature is fed into initial prediction heads to avoid task-pattern entanglement. Subsequently, both task-generic and -specific features are fed into BFE to produce the multi-scale bridge features $\mathbf{S}' = \{S_i' \mid i = 0, 1, 2, 3\}$. After the bridge features are formed, they are in turn used to fertilize task-specific features by TFR which transfers helpful representations from bridge features to task-specific features. Both BFE and TFR are applied on multiple scales, the task-specific features are up-sampled by shared up-sampling layers at each scale. Finally, the refined multi-scale task-specific features are aggregated and fed into prediction heads to make final predictions. The detailed structures will be described in the next sections.

3.2 Task Pattern Propagation

For multi-scale task-generic features, features with smaller scales contain richer high-level semantic and more contextual information and usually guide the formation of predictions in a top-down way. Meanwhile, different dense prediction tasks usually share a lot of similarities in low-level representations, but significantly differ in high-level representations. Thus, for the task-generic features produced by the final layers of the shared backbone, the high-level representations from different tasks are usually entangled severely, making it difficult to extract discriminative representations for each task and subsequently conduct interactions. We name this phenomenon the task-pattern-entanglement issue. As shown in Figure 3 (left), we visualize the regions with soft attention scores on semantic segmentation and depth estimation respectively, which show great differences that semantic attention focuses more on different objects and contexts while depth attention focuses more on spatial and geometry information. In Figure 3 (right), the patterns in the encoder shared outputs are implicit and entangled, which do not clearly reflect the distributions of either task. Thus, task-specific features directly decomposed from them

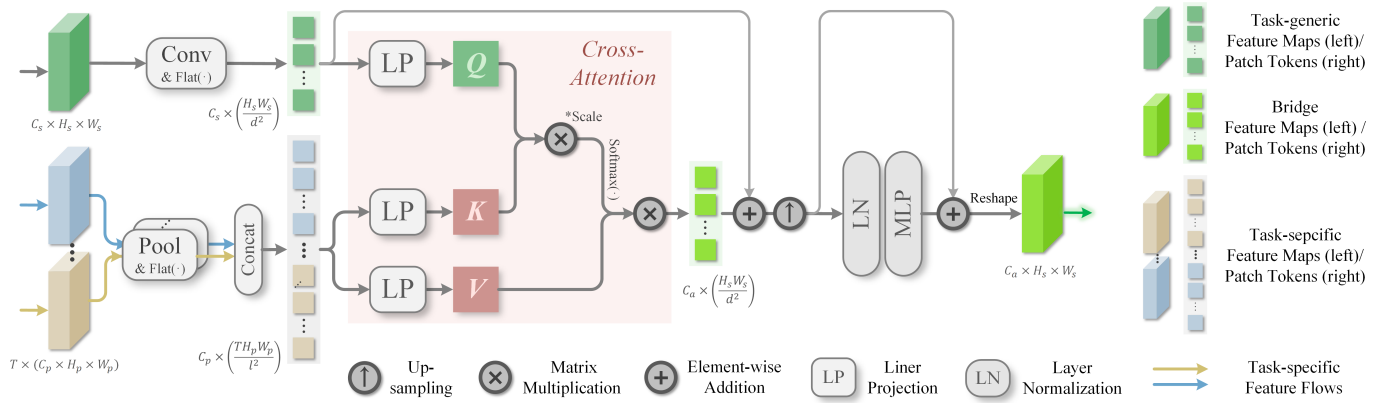


Fig. 5. A zoom-in version of one-scale BFE in Fig.2 (Different scales have similar structures). The input task-generic and task-specific features are firstly transformed into patch tokens, and then we use the task-generic tokens to query all of the task-specific tokens globally to produce a global correlation map, which helps select useful high-level information to gain high-level representations.

are struggling with the lack of discriminative semantics, which leads to low-quality interaction participants for the following interactions and eventually brings negative effects to interaction outputs.

To tackle the task-pattern-entanglement issue, we propose to conduct Task Pattern Propagation (TPP) that jointly learns and propagates different task patterns to corresponding tasks to assist in the formation of task-specific features. Although learning and propagating task patterns have been discussed in previous multi-task learning works [55], [57], the entanglement issues at the pattern level remain unexplored and unsolved, and our proposed TPP first time targeting this issue from the task pattern level. As shown in Fig. 4, firstly, for every task-specific feature maps P_3^j at scale 3, where $j = 1, 2, \cdots, T$, we correspondingly generate T sets of *query* Q_j, *key* K_j, and *value* V_j by linear projection for each task. Then task-specific attentions $A_j, j = 1, 2, \cdots, T$ are calculated by dot-product accordingly, which contains attention patterns of each task as follows:

$$A_j = \frac{Q_j \times K_j^\top}{\sqrt{C_a}}, \tag{1}$$

where C_a denotes the dimension for attention space. Subsequently, attention maps from all tasks A_j are concatenated and then squeezed by 1×1 convolution in a common space to align the dimension and share task patterns. Later the task patterns in attention maps are propagated by dot-product with all of the task *value*s:

$$\begin{aligned} \bar{A} &= \mathrm{Conv}_{1\times1}\left(\mathrm{Concat}\left(A_1, A_2, \cdots, A_T\right)\right), \\ x_j &= \mathrm{Softmax}_{\mathrm{dim=-1}}(\bar{A}) \times V_j, \quad j = 1, 2, \cdots, T. \end{aligned} \tag{2}$$

where $\mathrm{Softmax}_{\mathrm{dim=-1}}(\cdot)$ represents the softmax function applied to the last feature dimension.

Finally, x_j are processed by an FFN which has a similar structure in BFE to enhance the learned task-specific high-level representations. Since different tasks have various pattern distributions, it is necessary to perform a pattern propagation to share the attention space for multiple tasks. Empirically, the irrelevant pixels will be repressed during the matching to avoid the negative feature from transferring, which avoids the unexpected entanglement caused by task inconsistencies. With TPP, we extract explicit task patterns by self-attention and conduct propagation to avoid potential entanglements and achieve a clear decoupling effect of task-specific features, which also benefits the subsequent feature interactions of BFE and TFR. As shown in Fig. 3 (right), TPP can significantly strengthen the task patterns by producing more discriminative high-level semantics, like the highlighted floor area in semantic segmentation and door area in depth estimation, which boosts the subsequent interactions by providing high-quality participants.

3.3 Bridge Feature Extraction

In this subsection, we discuss the detailed structure of our key component, i.e. the Bridge Feature Extractor (BFE). The BFE is a transformer-based module, that aims at extracting useful high-level representations from all task-specific features and producing bridge features. The core part of BFE is the global modeling of correlations between task-generic features and all of the task-specific features, which selects important high-level information and transfers it to the generic features. The BFE is consecutively stacked at each scale to produce multi-scale bridge features, and the BFE block at each scale has the same structure. As illustrated in Fig 5, assuming that the input task-generic and -specific features are at scale i, i.e. $S_i \in \mathbb{R}^{C_s \times H_s \times W_s}$ and $P_i^j \in \mathbb{R}^{C_p \times H_p \times W_p}, j = 0, 1, \cdots, T$, they are firstly transformed into patch tokens by depth-wise convolution or average pooling with $stride = d$ and $stride = l$ respectively:

$$\begin{aligned} \widetilde{S}_i &= \mathrm{Flat}(\mathrm{Conv}(S_i)), \\ \widetilde{P_i^j} &= \mathrm{Flat}(\mathrm{Pool}(P_i^j)), \quad j = 0, 1, \cdots, T, \end{aligned} \tag{3}$$

where $\widetilde{S}_i \in \mathbb{R}^{C_s \times \left(\frac{H_s W_s}{d^2}\right)}$ and $\widetilde{P_i^j} \in \mathbb{R}^{C_p \times \left(\frac{H_p W_p}{l^2}\right)}$, $\mathrm{Flat}(\cdot)$ represents the flatten operation for tensors in the spatial dimensions. The patch tokens will receive a spatial reduction which controls the computation cost of the following attention calculation. Afterward, all of the task-specific patch tokens are concatenated at the spatial dimension. To conduct the cross-attention, the *query* (Q) is projected from the transformed task-generic patch tokens, the *key* (K) and *value* (V) is projected from the transformed task-specific patch tokens. The production of Q, K and V can be described as:

$$\begin{aligned} Q &= \mathrm{LP}(\widetilde{S}_i)^\top, \\ K &= \mathrm{LP}(\mathrm{Concat}(\widetilde{P_i^0}, \widetilde{P_i^1}, \cdots, \widetilde{P_i^T})^\top, \\ V &= \mathrm{LP}(\mathrm{Concat}(\widetilde{P_i^0}, \widetilde{P_i^1}, \cdots, \widetilde{P_i^T})^\top, \end{aligned} \tag{4}$$

where $Q \in \mathbb{R}^{\left(\frac{H_s W_s}{d^2}\right) \times C_a}$ and $K, V \in \mathbb{R}^{\left(\frac{T H_p W_p}{l^2}\right) \times C_a}$, C_a is the channel dimension for attention. Then, we conduct a standard cross-attention:

$$A = \frac{Q \times K^\top}{\sqrt{C_a}}, \quad A \in \mathbb{R}^{\frac{H_s W_s}{d^2} \times \frac{T H_p W_p}{l^2}}, \tag{5}$$

after the attention map is calculated, the output of cross-attention is:

$$x = \mathrm{Softmax}_{\mathrm{dim=-1}}(A) \times V, \quad x \in \mathbb{R}^{\left(\frac{H_s W_s}{d^2}\right) \times C_a}, \tag{6}$$

Then x is transposed, reshaped, and up-sampled into $C_a \times (H_s W_s)$ and fed into a feed-forward network which is composed of a layer normalization and an MLP. The final output is reshaped into $C_a \times H_s \times W_s$, which is the bridge feature we denote as S_i' at scale i. By querying all of the task-specific features globally, the task-generic feature gains discriminative high-level representations by selecting the task-specific pixels that have the highest response to the task-generic features in A, and the low-level representations are simultaneously reserved by residual paths. Thus, the extracted bridge feature meets the need as a medium for multi-task dense prediction. Different from directly conducting pixel-wise addition of task-generic and -specific features, which might cause unexpected conflicts if there is task inconsistency exists in a certain position, the global querying ensures only the highest responding task-specific pixel is selected and avoids the potential task conflicts.

3.4 Task Feature Refiner

To conduct effective feature fusion between bridge features $\mathbf{S}'$ and task-specific features $\mathbf{P}$, in order to take advantage of the rich representations in $\mathbf{S}'$, we propose a Task Feature Refiner (TFR), as shown in the right part of Fig. 4 (b). TFR utilizes bridge features to guide task-specific characteristics and achieve effective interaction between tasks. TFR offers a relatively flexible configuration, employing a cascaded structure where each layer consists of TFR layers with the same structures. Within each layer, bridge feature S_i' from a certain scale i needs to be concatenated with each task-specific feature $P_i^j, j = 0, 1, \cdots, T$, followed by fusion through the TFR layer:

$$\begin{aligned} x^{(k)} &= \mathrm{TFRlayer}^{(k)}\left(\mathrm{Concat}\left(S_i', x^{(k-1)}\right)\right), \\ j &= 0, 1, \cdots, T; \quad k = 1, 2, \cdots, M; \quad x^{(0)} = P_i^j, \end{aligned} \tag{7}$$

where M represents the total number of layers in TFR, stacking these TFR layers forms the entire TFR module. During this process, the task-specific features P_i^j undergo progressive refinement through these layers, continuously guided by the bridge features, obtaining abundant task representations. This ultimately generates high-quality task-specific features, laying a solid foundation for the generation of final task predictions.

Additionally, our TFR layers can be flexibly configured. In the simplest case, we can use just one convolutional layer to align the concatenated feature channels and generate predictions. However, this is certainly insufficient, as a single convolutional layer lacks the necessary capacity for nonlinear transformations. Alternatively, Transformer layers can be employed for global relationship modeling, such as InvPT [51]. Yet, experimental findings reveal that the complex quadratic relationship modeling of the Transformer does not significantly improve the quality of task features. This is partly because the scales of the feature maps are not large at this stage, and global modeling doesn't offer significant advantages. Furthermore, spatial local similarity serves as a more crucial prior for dense predictions. Hence, in our method, we utilize depth-wise separable dilated convolutions to form our fundamental layer. We stack dilated convolutions of varying sizes to ensure sufficient receptive fields and avoid the grid effect associated with dilation. Simultaneously, the local connections brought by convolutions are suitable for extracting vital local information from the bridge features to generate the final predictions.

4 Experiment

4.1 Experimental Setup

Datasets: We conduct experiments on three benchmark datasets: NYUD-v2 [35], Cityscapes [9] and PASCAL Context [7]. NYUD-v2 dataset is mainly used for indoor scene segmentation and depth estimation in MTL works. The dataset contains 1449 images. Following the standard settings in [38], we use 795 images for training which are randomly selected, and the rest are for testing. Cityscapes [9] contains 2975 and 500 high-resolution street-view images captured from different European countries for training and testing respectively. These images are annotated with fine 19-class semantic labels for segmentation and disparity map for depth estimation. To speed up the convergence and train all models in comparisons with controllable computational cost overhead, we down-sample the images with their annotations to 384×768 resolution. PASCAL is a popular benchmark for many dense prediction tasks and we use the split from PASCAL Context, which contains 10103 images, where 4998 images are randomly selected for training and the rest are for testing [38]. We choose several subsets of tasks on both datasets to make a more explicit comparison, including semantic segmentation (Seg.), human parts segmentation (H.Parts), depth eatimation (Dep.), surface normal estimation (Norm.), saliency estimation (Sal.) and edge detection (Edge.).

Metrics: We consider multiple metrics for different tasks respectively and conduct extensive experiments to further validate the effectiveness of our model. The metric notations are listed as follows:

1) *mIoU*: mean intersection over union.
2) *rmse*: root mean square error. (For surface normal estimation, we calculate the rmse of normal angle.)
3) *mErr*: mean of angle error.
4) *max-F*: maximum of $F_1 - measure$.
5) *odsF*: optimal dataset scale F-measure

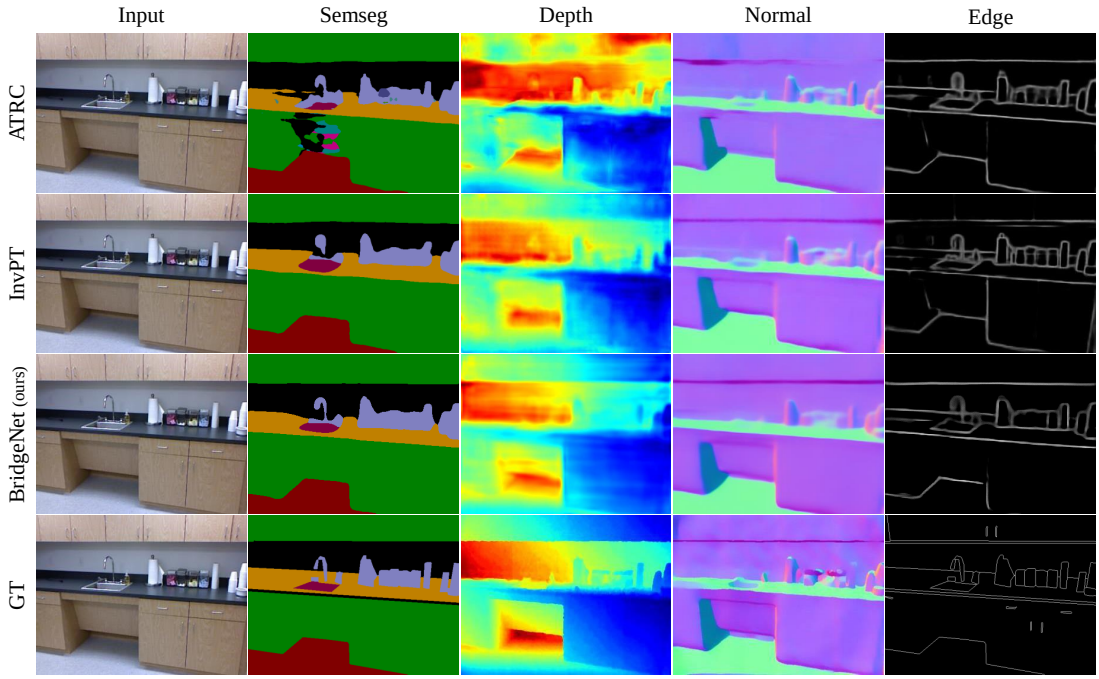


Fig. 6. As visualized above, we compare with SOTA works (ATRC [3] and InvPT [51]) on NYUD-v2 4 tasks. Our BridgeNet clearly produces more precise predictions on each task. To avoid cherry-picking, the input image is selected from the sample examples provided in [51] paper.

TABLE 1
Comparison with SOTA works on NYUD-v2 and PASCAL Context.

NYUD-v2					PASCAL Context					
Models	Seg. *mIoU*↑	Dep. *rmse*↓	Norm. *mErr*↓	Edge. *odsF*↑	Models	Seg. *mIoU*↑	H.Parts *mIoU*↑	Sal. *maxF*↑	Norm. *mErr*↓	Edge. *odsF*↑
Cross-Stitch [30]	36.34	0.6290	20.88	76.38	ASTMT [29]	68.00	61.10	65.70	14.70	72.40
PAP [55]	36.72	0.6178	20.82	76.42	PAD-Net [49]	53.60	56.60	65.80	15.30	72.50
PSD [57]	36.69	0.6246	20.87	76.42	MTI-Net [39]	61.70	60.18	84.78	14.23	70.80
PAD-Net [49]	36.61	0.6270	20.85	76.38	ATRC [3]	62.69	59.42	84.70	14.20	70.96
MTI-Net [39]	45.97	0.5365	20.27	77.84	ATRC-ASPP [3]	63.60	60.23	83.91	14.30	70.86
ATRC [3]	46.33	0.5363	20.18	77.94	ATRC-BMTAS [3]	67.67	62.93	82.29	14.24	72.42
InvPT [51]	53.56	0.5183	19.04	78.10	InvPT [51]	79.03	67.61	84.81	14.15	73.00
BridgeNet (*ours*)	**56.57**	**0.4655**	**17.29**	**80.02**	BridgeNet (*ours*)	**79.89**	**71.33**	**85.64**	**13.38**	**73.24**

TABLE 2
Comparison with SOTA works on Cityscapes.

Cityscapes		
Models	Seg. *mIoU*↑	Dep. *rmse*↓
Cross-Stitch [30]	85.31	3.422
NDDR-CNN [15]	84.73	3.415
MTAN [24]	90.20	3.442
PAD-Net [49]	90.13	3.496
MTI-Net [39]	91.50	2.680
InvPT [51]	91.19	2.631
BridgeNet (*ours*)	**92.61**	**2.606**

Additionally, to better evaluate the proposed method, we consider the relative gain in each task, for task τ_j, $j = 1, \ldots, N$, the relative gain Δ_{τ_j} can be designed as:

$$\Delta_{\tau_j} = (-1)^{l_j} \left(M_{m,j} - M_{s,j}\right) / M_{s,j}, \tag{8}$$

where $l_j = 1$ if a lower value means better for performance measure M_j of task j, and $l_j = 0$ if higher is better.

Also, we use the *multi-task performance* Δ_{MTL} from [38] to evaluate the mutual promotion in all tasks, defined as:

$$\Delta_{MTL} = \frac{1}{N} \sum_{j=1}^{N} \Delta_{\tau_j}, \tag{9}$$

Implementation Details: We conduct our experiments on Pytorch [32] with one NVIDIA Tesla V100 GPU. The models used for different evaluation experiments were trained for $40,000$ iterations on both datasets with a batch size of 6. We employed various backbone networks to comprehensively validate our approach. These include classic single-scale CNN encoders like the ResNet [17] series with dilated convolutions, single-scale Transformer encoders like the ViT [11] series, multi-scale dense prediction network HRNet [36] series, and multi-scale deformable convolution series InternImage [42]. For models using ViT and ResNet-50 backbones, we utilized the Adam optimizer with a learning rate of 2×10^{-5} and a weight decay rate of 1×10^{-6}.

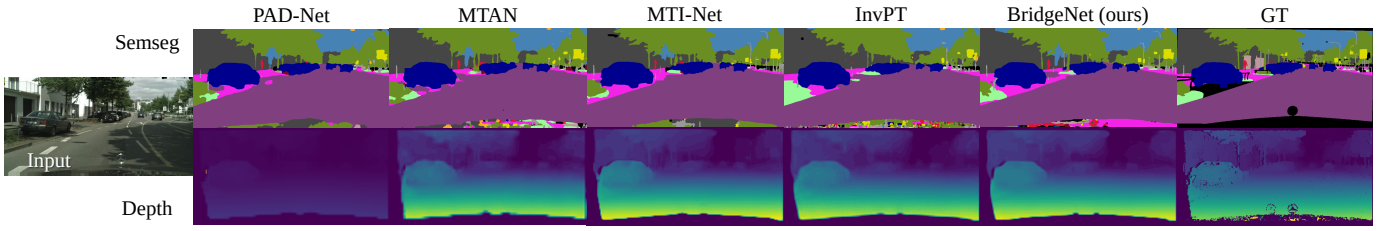


Fig. 7. As visualized above, we compare with SOTA works (including PAD-Net [49], MTAN [24], MTI-Net [39] and InvPT [51]) on Cityscapes.

TABLE 3
Comparison with SOTA works on different backbones.

NYUD-v2						PASCAL Context						
Models		Seg. *mIoU*↑	Dep. *rmse*↓	Norm. *mErr*↓	Edge. *odsF*↑	Models		Seg. *mIoU*↑	H.Parts *mIoU*↑	Sal. *maxF*↑	Norm. *mErr*↓	Edge. *odsF*↑
HRNet-w18	MTI-Net	39.89	0.5824	20.57	76.60	ResNet-50d	ASTMT	66.80	61.10	66.10	14.70	70.90
	ATRC	**40.80**	0.5826	20.51	76.50		MTI-Net	66.60	**63.30**	66.60	14.60	**74.90**
	BridgeNet	40.74	**0.5662**	**19.95**	**77.24**		ATRC	62.69	59.42	**84.70**	**14.20**	70.96
HRNet-w48	MTI-Net	45.97	0.5365	20.27	77.86		ATRC-ASPP	63.60	60.23	83.91	14.30	70.86
	ATRC	**46.33**	0.5363	20.18	77.94		ATRC-BMTAS	67.67	62.93	82.29	14.24	72.42
	BridgeNet	46.28	**0.5292**	**19.76**	**78.24**		BridgeNet	**69.79**	62.49	84.28	14.27	72.30
ViT-base	InvPT	50.30	0.5367	19.00	77.86	ViT-base	InvPT	77.33	66.62	**85.14**	13.78	**73.20**
	BridgeNet	**51.14**	**0.5186**	**18.92**	**77.98**		BridgeNet	**77.98**	**68.19**	85.06	**13.48**	72.98
ViT-large	InvPT	53.56	0.5183	19.04	78.10	ViT-large	InvPT	79.03	67.61	**84.81**	14.15	**73.00**
	BridgeNet	**55.51**	**0.4930**	**18.47**	**78.22**		BridgeNet	**80.64**	**70.06**	84.64	**13.82**	72.96

TABLE 4
Analysis of model sizes and cost. **Bold** and underlined fonts represent the first and second best results respectively.

Models	FLOPs (GB)	Params (M)	Latency (s)	Memory (GB)	Seg. *mIoU*↑	H.Parts *mIoU*↑	Sal. *maxF*↑	Norm. *mErr*↓	Edge. *odsF*↑
PAD-Net [49] *w.* ViT-large	773	330	0.043	3.096	78.01	67.12	79.21	14.37	72.60
MTI-Net [39] *w.* ViT-large	774	851	0.087	7.642	78.31	67.40	84.75	14.67	73.00
ATRC [3] *w.* ViT-large	871	340	0.098	7.933	77.11	66.84	81.20	14.23	72.10
InvPT [51] *w.* ViT-large	669	423	0.060	3.510	79.03	67.61	84.81	14.15	73.00
BridgeNet *w.* ViT-base	412	230	0.090	2.710	77.98	68.19	85.06	13.48	72.70
BridgeNet *w.* ViT-large	645	477	0.144	3.718	**80.64**	<u>70.06</u>	84.64	13.82	72.96
BridgeNet *w.* InternImage-B	406	248	0.130	2.774	78.07	69.03	<u>85.29</u>	<u>13.44</u>	<u>73.10</u>
BridgeNet *w.* InternImage-L	527	420	0.171	3.494	<u>79.89</u>	**71.33**	**85.64**	**13.38**	**73.24**

For models using HRNet backbones, we utilized the SGD optimizer with a learning rate of 0.01 and a weight decay rate of 5×10^{-4}, and the momentum weight is 0.9. And for models using InternImage backbones, we utilized the AdamW optimizer with a learning rate of $6 \times 10^{-5}$and 2×10^{-5} for the *base* and the *large* model respectively, and a weight decay rate of 0.05, the β for the optimizers are $(0.9, 0.999)$. A polynomial learning rate decay scheduler was used.

We maintained alignment with the settings from [51]. For single-scale image backbones, we first extract the intermediate backbone features at specified depths, then convert their spatial and channel dimension by convolution layers to build a multi-scale feature pyramid. The last three layers were taken as the multi-scale input for BridgeNet. Correspondingly, BFE and TFR were applied only to features from these last three scales, ensuring a favorable trade-off between performance and inference speed. For multi-scale image backbones, they directly produce a feature pyramid which is acceptable for BridgeNet.

For ViT-base and ViT-large, the initial output channel numbers of the preliminary decoders were 768 and 1024, respectively. The number of heads in the multi-head attention of BFE and TPP was set to 2. The downsampling ratios for query vectors were 2, and for key and value vectors, they were $[2, 4, 8]$. Regarding the HRNet series models, following the settings of previous multi-scale multi-task methods [3], [39], we utilized features from all four scales. The downsampling ratios for key and value vectors were $[2, 4, 8, 16]$. For HRNet-w18 and HRNet-w48, the initial output channel numbers of the preliminary decoders were set at 144 and 384, respectively. For the InterImage-B and InterImage-L backbones, the initial output channel numbers of the preliminary decoders were 896 and 1280 respectively. Regarding the scale of TFR, we established three different sizes: *base* with 2 layers, *large* with 4 layers, and *huge* with

6 layers. Each TFR layer followed the standard design of Hybrid Dilated Convolution (HDC) [41], employing three layers of depth-wise separable dilated convolutions with dilation rates of $[1, 2, 5]$. This was done to avoid the grid effect brought about by multiple layers of dilated convolutions and to achieve as large a receptive field as possible.

Baselines: Following the previous works, we use the same baseline setting by employing a simple multi-task baseline with a shared encoder and multiple decoders. The features from the backbone network's output are directly fed into task-specific decoders or prediction heads, each receiving label supervision from its respective task.

4.2 Comparison with SOTA Methods

4.2.1 Overall Comparisons

We compared BridgeNet with various state-of-the-art multi-task methods on the NYUD-v2, Cityscapes and PASCAL Context datasets. We employed InternImage-L as the backbone for training on NYUD-v2 and PASCAL Context, and ViT-large as the backbone for Cityscapes. As shown in Table 1, across all the compared tasks, our method surpasses previous works significantly in all of the tasks. On the NYUD-v2 dataset, the tasks include semantic segmentation (Seg.), depth estimation (Dep.), surface normal estimation (Norm.), and edge detection (Edge.). For the NYUD-v2 comparisons, we used TFR-*huge*, and BridgeNet outperforms previous methods on all tasks. Particularly significant improvements are achieved in semantic segmentation and depth estimation. Compared to InvPT [51], our method enhances by $+3.01$ *mIoU* and -0.0528 *rmse*, respectively. Compared to the previous state-of-the-art multi-scale interaction MTI-Net [39], we achieve a significant improvement of $+10.60$ *mIoU* and -0.0710 *rmse*. This improvement is substantial. Noticeable enhancements are also seen in the other two tasks. Qualitative comparisons are shown in Fig. 6, which further prove that BridgeNet yields better prediction quality.

On the Cityscapes, the tasks include semantic segmentation (Seg.) and depth estimation (Dep.), and we use TFR-*huge*. Compared with encoder-focused methods which only utilize low-level representations in task-generic features for task interactions, our method significantly achieves better performance on both tasks. Also, our BridgeNet surpasses the best performing previous decoder-focused InvPT [51] by $+1.42$ *mIoU* and -0.025 *rmse*. We also make qualitative comparisons in Fig. 7 to further validate the effectiveness of our method.

On the PASCAL Context dataset, the tasks include semantic segmentation (Seg.), human body part segmentation (H.Parts), surface normal estimation (Norm.), saliency estimation (Sal.), and edge detection (Edge.). We used TFR-*large* for BridgeNet. Due to the increased number of tasks, achieving a balance in performance across multiple tasks becomes challenging, and maintaining superiority across all tasks over previous works is not easy to achieve. Nevertheless, even under these conditions, we achieved balanced promotions in all tasks. More significant improvement occurred in human body part segmentation (H.Parts) and saliency estimation (Sal.), resulting in $+3.72$ *mIoU* and -0.77 *mErr*. Besides, we also conduct qualitative comparisons in Fig. 8. Furthermore, we visualized the task-specific features refined through TFR, as shown in Fig. 9. In comparison to works lacking the TFR structure, such as MTI-Net [39], the task-specific features generated by our method exhibit clearer high-level semantics, highlighted regions possess sharper boundaries, and overall coherence is improved.

4.2.2 Comparisons on Different Backbones

Different from InvPT [51] which is specially designed for ViT backbones, our method is adaptive to different types of backbones. We also employed different backbones to compare with various established methods, ensuring the effectiveness of our method across diverse conditions. We utilized two different sizes of ViTs [11], ViT-base and ViT-large, two different sizes of HRNet [36], HRNet-w18 and HRNet-w48, as well as the commonly used ResNet-50d with dilated convolutions [6], [17]. As observed, our BridgeNet demonstrates robust performance across various types of backbone networks.

When using the HRNet series backbone networks, our method exhibits comparable performance to other methods in the Semseg. task, while showcasing significant improvements in other spatial geometric estimation tasks. On ResNet-50d, though our method achieves competitive performance, the performance among different tasks is more balanced, which means our method does not tend to sacrifice performance on one task to enhance another. When employing larger parameter backbone networks such as the ViT series, our method shows significant enhancements in high-level semantic understanding tasks like Semseg. and H.Parts. The reason behind this phenomenon is that when the backbone network has fewer parameters, its feature extraction capacity might be limited. Simpler geometric estimation tasks (Norm., Edge.) might not have reached saturation, allowing the model to learn useful representations from richer semantic tasks (Semseg.) and thereby enhance the performance. Conversely, when the backbone network has more parameters, it can capture precise image representations, reaching saturation in geometric estimation tasks. At this point, further improvements become challenging. However, larger feature space and richer representations become advantageous for tasks that require advanced information (Semseg. and H.Parts). This phenomenon is also observed in [51].

4.2.3 Comparisons on Model Size and Cost

To further show the efficiency of our BridgeNet, we compare our method with some of the previous SOTA works with the consideration of computation cost, parameter amount, latency, and memory cost regarding a single image input. We test all models on a single NVIDIA 3090 GPU under the same condition. As shown in Table 4, our method achieves competitive performance with ViT-base compared with other methods applied with ViT-large, and under the same ViT-large backbones, our method shows clear superiority over previous works on the majority of tasks, especially on Seg., Sal. and Norm. Benefits from the high-efficiency interactions brought by BFE, our BridgeNet with the configuration of TFR-*large* reduces over 100 GFLOPs compared with [3], [39], [49], and 24 GFLOPs compared with InvPT [51]. Considering the memory cost for single image

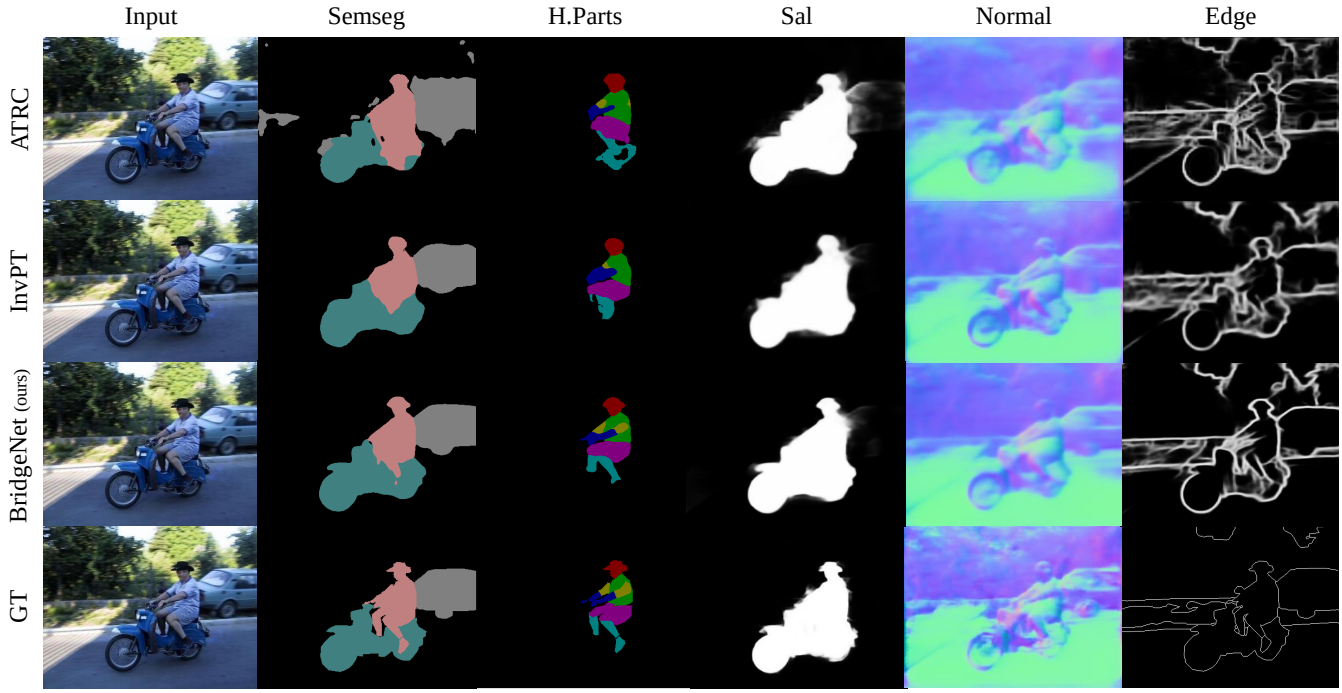


Fig. 8. As visualized above, we compare with SOTA works (ATRC [3] and InvPT [51]) on PASCAL Context 5 tasks. Our BridgeNet clearly produces more precise predictions on each task. We also use the sample example image provided in [51] paper for comparisons.

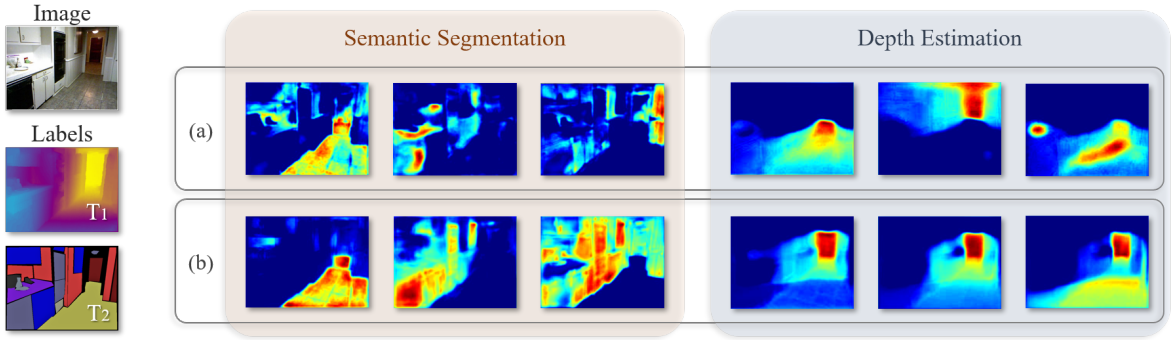


Fig. 9. The visualization of feature maps at scale 0 by MTI-Net **(a)** and our BridgeNet **(b)**. In semantic segmentation **(left)**, our method presents more consistency with clear edges for the same semantic region. The brighter area means more attention is paid in those areas. In depth estimation **(right)**, our method also produces more accurate edges for areas with different depths, such as the door in the faraway distance which is supposed to be the brightest area. T_1 and T_2 represent semantic segmentation and depth estimation labels respectively.

inference, our BridgeNet significantly surpasses methods with heavy Multi-scale Feature Propagation Module (FPM) like [3], [39], with using less than half of the memory. Though our method is slightly slower when comparing the inference latency with other methods, which is due to the two-step extraction and refinement involving the bridge feature, however the advantages in model size, FLOPs, memory consumption, etc. prove that our BridgeNet saves a lot of additional computational overhead for multi-task interactions.

For different backbone configurations, our method shows better performance with less computation costs on InternImage backbones. Compared with ViT-base, our method performs better with compatible FLOPs and Params on InternImage-B. When compared with ViT-large, our method achieves better performance on InternImage-L even with fewer parameters (-57 MParams) and less computation costs (-118 GFLOPs).

TABLE 5
Ablation analysis of the components of BridgeNet. The experiments are conducted on NYUD-v2.

Models	Seg. *mIoU* ↑	Dep. *rmse* ↓	Norm. *mErr* ↓	Edge. *odsF* ↑	Δ_{MTL} (%) ↑
STL	50.95	0.5698	19.08	78.28	-
MTL basline	48.30	0.5605	19.08	77.42	-1.17
+BFE	50.30	0.5367	19.00	77.40	0.96
+BFE+TPP	50.22	0.5312	18.97	77.68	1.29
+BFE+TPP+TFR-*huge*	51.14	0.5186	18.92	77.98	2.45
$\Delta_{\tau_j}/(\%)$ ↑	0.37	8.99	0.84	-0.38	-

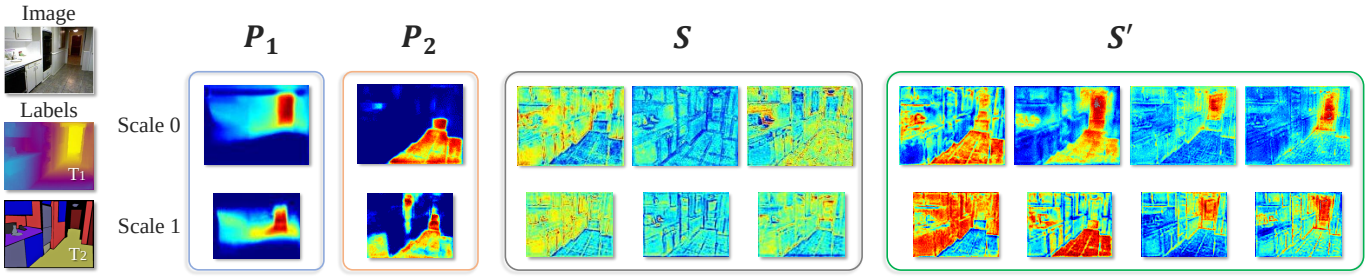


Fig. 10. We randomly visualized feature maps from Bridge Features ($\mathbf{S}'$), task-generic features ($\mathbf{S}$), and task-specific features ($\mathbf{P}$) at scales 0 and 1. In comparison to $\mathbf{S}$, $\mathbf{S}'$ exhibits more high-level representations, highlighted areas on different distances, and semantic objects. Additionally, $\mathbf{S}'$ retains the rich low-level features obtained in $\mathbf{S}$, such as edges and textures.

TABLE 6
Ablation analysis of refine strategies. The experiments are conducted on NYUD-v2.

Strategies	Params (M)	Seg. *mIoU* ↑	Dep. *rmse* ↓	Norm. *mErr* ↓	Edge. *odsF* ↑	Δ_{MTL} (%) ↑
STL	-	50.95	0.5698	19.08	78.28	-
MTL basline	-	48.30	0.5605	19.08	77.42	-1.17
Add	+0	49.94	0.5354	18.89	77.34	0.96
Concatenate	+7.3	50.20	0.5303	18.95	77.10	1.16
Add + InvPT	+30.7	50.33	0.5386	18.92	77.32	0.97
TFR-*base*	+15.9	50.36	0.5267	19.04	77.86	1.52
TFR-*large*	+31.7	50.86	0.5247	18.95	77.84	1.96
TFR-*huge*	+47.6	51.14	0.5186	18.92	77.98	2.45

4.3 Ablation Study

The ablation experiments are divided into three sections. In section 4.3.1, Model Analysis, we delve into the significance of the three core modules that constitute BridgeNet: BFE, TPP, and TFR. We provide comprehensive visual analysis in this regard. In section 4.3.2, Refine Strategies Analysis, we explore how different methods of feature refinement impact the final predictive performance, including varying sizes of TFR and other techniques. In section 4.3.3, Task Set Analysis, we investigate how the model's performance changes when dealing with different combinations of tasks in multi-task learning. For all the experiments, we employ ViT-base as the backbone network.

4.3.1 Model Analysis

These experiments were conducted on the NYUD-v2 dataset. Initially, we trained and tested the performance of single-task learning (STL) on each individual task. Then, we calculated the performance of the multi-task learning baseline (MTL baseline) and progressively incorporated BFE, TPP, and TFR. The comparative results are presented in Table 5. As evident, due to the varying number of tasks and their differences, each task pursues a distinct optimization objective. Consequently, MTL baseline exhibits a noticeable performance decline in comparison to STL. Upon introducing BFE over the baseline, performance improvement is observed, nearing the level of advanced models like InvPT [51] shown in Table 3. However, at this stage, without TPP's assistance in obtaining more discriminative task-specific features, the quality of the Bridge Features generated by BFE is not optimal. Hence, with the introduction of TPP, further performance enhancement is achieved.

To better harness the bridge features, simple pixel-wise addition of feature maps is insufficient. Therefore, after adding TFR, task-specific features experience superior optimization. From Table 3, it can be observed that the performance improvement is more significant for Dep. among the four tasks. With each of our designed modules incorporated, the model's average multi-task performance Δ_{MTL} shows improvement. Despite encountering negative transfer among tasks, which prevents surpassing STL performance in Edge., the overall average performance across multiple tasks still improves. This underscores the effectiveness of our model design.

We also conducted a comprehensive visual analysis. As shown in Fig. 10. We selected bridge features ($\mathbf{S}'$) and compared them with task-generic features ($\mathbf{S}$) and task-specific features ($\mathbf{P}$) at different scales. These results are presented in Fig. 10. The distribution of task-specific features $\mathbf{P}$ is similar to their corresponding labels and tends to emphasize the relevant regions under supervision. For example, they highlight regions with significant depth variations, as well as distinct semantic areas like floors, walls, and cabinets. This suggests that they possess distinct task-specific perception capabilities. On the other hand, task-generic features $\mathbf{S}$ exhibit rich low-level representations, such as edges and textures. Bridge features $\mathbf{S}'$ exhibit both task-specific perception capabilities and low-level representations. This demonstrates that $\mathbf{S}'$ has the potential to serve as an intermediate medium for multi-task interaction, ensuring the integrity of feature interactions.

4.3.2 Refine Strategies Analysis

As discussed in Section 3.4, there are several ways to perform task refinement. Here, we compare the performance of different refinement methods and Task Feature Refiners (TFR) of varying sizes. The results are displayed in Table 4. Simply pixel-wise addition of task-specific features and bridge features, or concatenating and processing them with a single convolutional layer, yields sub-optimal results. Applying InvPT [51] for global spatial and task modeling on the added features also does not significantly improve performance. As analyzed in Section 3.4, bridge features need to effectively transfer learned representations to task-specific features, where local correlations are more critical, rendering global modeling unnecessary and even introduc-

TABLE 7
Comparison of different task sets on NYUD-v2.

Models	Seg. *mIoU*↑	$\Delta_{\tau_{seg}}$ (%)	Dep. *rmse*↓	$\Delta_{\tau_{dep}}$ (%)	Norm. *mErr*↓	$\Delta_{\tau_{norm}}$ (%)	Edge. *odsF*↑	$\Delta_{\tau_{edge}}$ (%)	Δ_{MTL} (%)
STL	50.95	-	0.5698	-	19.08	-	78.28	-	-
BridgeNet (TRF-*base*)	52.73	+3.49	0.5247	+7.92	-	-	-	-	5.70
	50.30	-1.28	-	-	19.03	+0.26	-	-	-0.51
	-	-	0.5416	+4.95	18.90	+0.94	-	-	2.94
	50.20	-1.47	0.5305	+6.90	19.03	+0.26	-	-	1.90
	-	-	0.5351	+6.09	18.89	+1.00	77.60	-0.87	2.07
	50.36	-1.16	0.5267	+7.56	19.04	+0.21	77.86	-0.54	1.52

TABLE 8
Comparison of different task sets, we use the TFR-*base* in decoding stage.

Models	Seg. *mIoU*↑	$\Delta_{\tau_{seg}}$ (%)	H.Parts *mIoU*↑	$\Delta_{\tau_{hpart}}$ (%)	Sal. *maxF*↑	$\Delta_{\tau_{sal}}$ (%)	Norm. *mErr*↓	$\Delta_{\tau_{norm}}$ (%)	Edge. *odsF*↑	$\Delta_{\tau_{edge}}$ (%)	Δ_{MTL} (%)
STL	79.69	-	71.15	-	84.77	-	13.26	-	73.00	-	-
BridgeNet (TRF-*base*)	80.70	+1.27	71.95	+1.12	-	-	-	-	-	-	1.20
	78.93	-0.95	-	-	85.20	+0.51	-	-	-	-	-0.22
	79.68	-0.01	70.74	-0.58	85.19	+0.50	-	-	-	-	-0.03
	-	-	-	-	85.12	+0.41	13.35	-0.68	2.50	-0.68	-0.31
	76.71	-3.74	67.33	-5.37	84.79	+0.02	13.49	-1.73	-	-	-2.70
	77.98	-2.15	68.19	-4.16	85.06	+0.34	13.48	-1.65	72.96	-0.05	-1.54

ing redundant parameters and computations. Our TFR-*base*, with only half the parameter count, outperforms the effect of refinement using InvPT ($\Delta_{MTL} 1.52\% > 0.97\%$). All three sizes of TFR consistently achieve improvements among all tasks, effectively striking a balance between performance and computational cost.

4.3.3 Task Set Analysis

The intrinsic characteristics of tasks in MTL significantly influences the effectiveness of collaborating. Tasks with close relationships might mutually enhance performance, while tasks with substantial differences could lead to negative transfer and performance degradation. In the domain of multi-task dense prediction, there is a relatively limited number of studies focusing on task composition through ablation experiments. Here, we conducted comprehensive experiments on NYUD-v2 and PASCAL Context datasets to analyze task composition effects.

Table 7 illustrates different combinations of tasks on the NYUD-v2 dataset. We considered 6 distinct task sets, including cases with 2, 3, and 4 tasks. It can be observed that when fewer tasks are involved, the occurrence of mutual enhancement between tasks becomes more likely. Notably, the most significant mutual enhancement happens between the Seg. and Dep. tasks (Seg. $+3.49\%$ $\Delta_{\tau_{seg}}$, Dep. $+7.92\%$ $\Delta_{\tau_{dep}}$), resulting in an average performance improvement of $+5.70\%\Delta_{MTL}$. This aligns with findings from several studies, indicating a strong latent correlation between depth estimation and semantic segmentation. However, the correlation between the Seg. and Norm. tasks are relatively weak, leading to considerable negative transfer effects (Seg. -1.28% $\Delta_{\tau_{seg}}$, Norm. $+0.26\%$ $\Delta_{\tau_{norm}}$), resulting in an average performance decline of $-0.51\%\Delta_{MTL}$. The surface normals are aligned with the gradients of depth in the space, establishing a strong correlation between these two tasks, facilitating mutual enhancement (Dep. $+4.95\%$ $\Delta_{\tau_{dep}}$, Norm. $+0.94\%$ $\Delta_{\tau_{norm}}$), and achieving an average performance improvement of $+2.94\%\Delta_{MTL}$. As the number of tasks increases, due to inherent inconsistencies between various task pairs, such as Seg. and Norm., performance tends to decrease. For example, the combination of Seg., Dep., and Norm. results in $+1.90\%\Delta_{MTL}$, with a more noticeable decline in Seg.'s performance. In contrast, the combination of Dep., Norm., and Edge. exhibits good performance, with an average improvement of $+2.07\%\Delta_{MTL}$, as all three tasks are designed to learn spatial geometry and lack evident conflicts.

Overall, among the four tasks, Dep. displays the most pronounced improvement due to its minimal conflicts with the other three tasks, allowing it to extract sufficient information from other tasks. In contrast, Seg. and Edge. exhibit relatively modest enhancements. Seg.'s potential conflicts hinder its enhancement, while Edge.'s phenomenon aligns with the analysis in ATRC [3], where it benefits less from the other tasks.

Table 8 depicts different combinations of tasks on the PASCAL Context dataset. We also examined 6 different task combinations, encompassing 2, 3, 4, and 5 tasks. Among these, the Seg., H.Parts, and Sal. tasks exhibit higher correlation. Learning these tasks together or in pairs does not yield noticeable negative effects. Seg. and H.Parts are tasks involving segmentation at different granularities, and they share a substantial amount of information. This direct information sharing explicitly enhances prediction accuracy of the *person* category in Seg., boosting the Intersection over Union (*IoU*) from 86.10 to 86.67. The interaction of high-level semantic information across different granularities benefits both tasks, resulting in an average performance improvement of $+1.20\%\Delta_{MTL}$. Similarly, the Seg. and Sal. tasks share some foreground and background information,

leading to no substantial negative impact, with an average performance change of $-0.22\%\Delta_{MTL}$. Learning all three tasks together results in an average performance change of $-0.33\%\Delta_{MTL}$, which is an acceptable compromise. However, introducing Norm. in addition to these three tasks significantly reduces the gains, with an average performance decline of $-2.70\%\Delta_{MTL}$. This observation aligns with the findings on the NYUD-v2 dataset. Furthermore, the addition of Edge. results in a performance rebound, enhancing the relative gains for nearly every task. This underscores the importance of Edge. for most dense prediction tasks. When learning Sal., Norm., and Edge., the three relatively simpler tasks together, the average performance is $+0.31\%\Delta_{MTL}$.

5 Conclusion

In this paper, we discovered the limitations of previous multi-task dense prediction methods in terms of incomplete levels of representations, less discriminative semantics in feature participants, and inefficient pair-wise task interaction processes. To address these challenges, we introduced a novel framework called BridgeNet. Specifically, our method adopts a shared encoder backbone to generate task-generic features. In the early decoding stage, a preliminary decoder with Task Pattern Propagation (TPP) is used to generate high-quality task-specific features. We introduced the Bridge Feature Extractor (BFE) to interact between globally queried task-specific features and backbone task-generic features, selecting task-specific perception information and generating multi-scale bridge features. The Task Feature Refiner (TFR) injects the representation of bridge features and iteratively optimizes the final task predictions, resulting in the ultimate task predictions. Compared to previous structures, our proposed method ensures the integrity of feature interaction, striking a good balance between performance and cost.

In our experiments, we evaluated our method on the widely used multi-task learning benchmark datasets, NYUD-v2, Cityscapes and PASCAL Context. By comparing with Single Task Learning (STL), Multi-Task Learning Baseline (MTL baseline), PAD-Net, MTI-Net, InvPT, and our BridgeNet, we observed the superior performance of BridgeNet across multiple tasks. Furthermore, we conducted a comprehensive visual analysis to demonstrate the advantage of bridge features over the other two types (task-generic and task-specific features). Compared to models generating and utilizing task-generic and task-specific features, such as BMTAS and InvPT, BridgeNet dives deeper into feature mining from important dense prediction priors, driving significant performance improvements. We analyzed the reasons for our method's superiority at the feature level. Moreover, through ablation experiments on model components, we analyzed the effectiveness of each component and demonstrated the improvements of our method tailored for dense prediction tasks. Additionally, our ablation experiments on task sets started from the nature of each task to analyze task relationships and studied the impact of different task combinations on multi-task learning performance. By conducting these ablation experiments on task sets, we gained a better understanding of the interactions between tasks and provided guidance for constructing effective multi-task learning models.

In summary, our research addressed the limitations of previous multi-task dense prediction methods through the introduction of the BridgeNet approach. Experimental results show that BridgeNet achieves significant performance improvements in multi-task learning, which were further validated through visual analysis and ablation experiments. Our work provides valuable insights and guidance for research and practice in the field of multi-task dense prediction. Future research can further explore the application and optimization of the BridgeNet method to meet the broader demands of dense prediction tasks.

References

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

[2] David Bruggemann, Menelaos Kanakis, Stamatios Georgoulis, and Luc Van Gool. Automated search for resource-efficient branched multi-task networks. *arXiv preprint arXiv:2008.10292*, 2020.

[3] David Brüggemann, Menelaos Kanakis, Anton Obukhov, Stamatios Georgoulis, and Luc Van Gool. Exploring relational context for multi-task dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15869–15878, 2021.

[4] Yancheng Cai, Bo Zhang, Baopu Li, Tao Chen, Hongliang Yan, and Jingdong Zhang. Rethinking cross-domain pedestrian detection: a background-focused distribution alignment framework for instance-free one-stage detectors. *IEEE transactions on image processing*, 2023.

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

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

[7] 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 *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1971–1978, 2014.

[8] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7103–7112, 2018.

[9] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016.

[10] Michael Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[12] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2650–2658, 2015.

[13] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in Neural Information Processing Systems*, 27, 2014.

[14] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019.

[15] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3205–3214, 2019.

[16] Pengsheng Guo, Chen-Yu Lee, and Daniel Ulbricht. Learning to branch for multi-task learning. In *International Conference on Machine Learning*, pages 3854–3863. PMLR, 2020.
[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
[18] Falk Heuer, Sven Mantowsky, Saqib Bukhari, and Georg Schneider. Multitask-centernet (mcn): Efficient and diverse multitask learning using an anchor free approach. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 997–1005, 2021.
[19] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth international conference on 3D vision (3DV)*, pages 239–248. IEEE, 2016.
[20] Wenbo Lan, Jianwu Dang, Yangping Wang, and Song Wang. Pedestrian detection based on yolo network model. In *2018 IEEE international conference on mechatronics and automation (ICMA)*, pages 1547–1551. IEEE, 2018.
[21] Wei-Hong Li and Hakan Bilen. Knowledge distillation for multi-task learning. In *European Conference on Computer Vision*, pages 163–176. Springer, 2020.
[22] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *European conference on computer vision*, pages 1–18. Springer, 2022.
[23] Yongqing Liang, Xin Li, Navid Jafari, and Jim Chen. Video object segmentation with adaptive feature bank and uncertain-region refinement. *Advances in Neural Information Processing Systems*, 33:3430–3441, 2020.
[24] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1871–1880, 2019.
[25] Yingfei Liu, Junjie Yan, Fan Jia, Shuailin Li, Aqi Gao, Tiancai Wang, and Xiangyu Zhang. Petrv2: A unified framework for 3d perception from multi-camera images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3262–3272, 2023.
[26] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.
[27] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.
[28] Yongxi Lu, Abhishek Kumar, Shuangfei Zhai, Yu Cheng, Tara Javidi, and Rogerio Feris. Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5334–5343, 2017.
[29] Kevis-Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos. Attentive single-tasking of multiple tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1851–1860, 2019.
[30] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 3994–4003, 2016.
[31] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016.
[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural information Processing Systems*, 32:8026–8037, 2019.
[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
[34] Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. Latent multi-task architecture learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4822–4829, 2019.
[35] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760. Springer, 2012.
[36] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5693–5703, 2019.
[37] Simon Vandenhende, Stamatios Georgoulis, Bert De Brabandere, and Luc Van Gool. Branched multi-task networks: Deciding what layers to share. *Proceedings British Machine Vision Conference 2020*, 2019.
[38] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
[39] Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Mti-net: Multi-scale task interaction networks for multi-task learning. In *European Conference on Computer Vision*, pages 527–543. Springer, 2020.
[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural information Processing Systems*, pages 5998–6008, 2017.
[41] Panqu Wang, Pengfei Chen, Ye Yuan, Ding Liu, Zehua Huang, Xiaodi Hou, and Garrison Cottrell. Understanding convolution for semantic segmentation. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 1451–1460. Ieee, 2018.
[42] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14408–14419, 2023.
[43] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 568–578, 2021.
[44] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.
[45] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016.
[46] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22–31, 2021.
[47] Runmin Wu, Mengyang Feng, Wenlong Guan, Dong Wang, Huchuan Lu, and Errui Ding. A mutual learning method for salient object detection with intertwined multi-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8150–8159, 2019.
[48] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021.
[49] Dan Xu, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 675–684, 2018.
[50] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1281–1292, 2020.
[51] Hanrong Ye and Dan Xu. Inverted pyramid multi-task transformer for dense scene understanding. *ECCV*, 2022.
[52] Jingwen Ye, Yixin Ji, Xinchao Wang, Kairi Ou, Dapeng Tao, and Mingli Song. Student becoming the master: Knowledge amalgamation for joint scene parsing, depth estimation, and more. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2829–2838, 2019.
[53] Jingdong Zhang, Hanrong Ye, Xin Li, Wenping Wang, and Dan Xu. Multi-task label discovery via hierarchical task tokens for partially annotated dense predictions. In *Processing*, 2024.
[54] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Zequn Jie, Xiang Li, and Jian Yang. Joint task-recursive learning for semantic segmentation and depth estimation. In *Proceedings of the European Conference on*

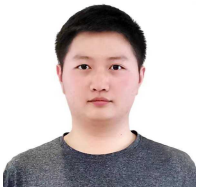
Computer Vision, pages 235–251, 2018.
[55] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan, Nicu Sebe, and Jian Yang. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4106–4115, 2019.
[56] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2881–2890, 2017.
[57] Ling Zhou, Zhen Cui, Chunyan Xu, Zhenyu Zhang, Chaoqun Wang, Tong Zhang, and Jian Yang. Pattern-structure diffusion for multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4514–4523, 2020.

Jingdong Zhang is a second-year Ph.D. student in Computer Science and Engineering at Texas A&M University, specializing in computer graphics and vision. He received his bachelor's degree in Intelligent Science and Technology from Fudan University. His research interests include multi-task learning, dense scene parsing, neural rendering, 3D reconstruction and generation. He serves as a reviewer for various conferences, including CVPR, ECCV, ICRA, and PG.

Jiayuan Fan received the Ph.D. degree in information engineering from Nanyang Technological University, Singapore, in 2015. After her graduation, she worked as a Research Scientist at the Institute for Infocomm Research, A*STAR, Singapore. She is currently an associate professor with Academy for Engineering and Technology in Fudan University, Shanghai, China. Her main research interests include computer vision, and image forensic analysis and application.

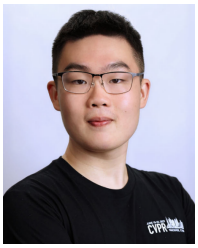
Peng Ye is currently pursuing the Ph.D. degree with Fudan University, Shanghai, China. He has published papers in leading journals and conferences, including PAMI, IJCV, CVPR Oral, NeurIPS, ACM MM Oral, ICME Best Student Paper, TGRS, TCSVT, and ICASSP Oral. His research interests include computer vision, network design, and network optimization. He serves as a Reviewer for various journals and conferences, including PAMI, IJCV, TCSVT, CVPR, ECCV, ICCV, and ICLR.

Bo Zhang is currently a Research Scientist at the Shanghai Artificial Intelligence Laboratory. He has published 22 papers in top-tier international conferences and journals such as CVPR, NeurIPS, ICML, ICLR, etc. Professionally, he led the development of the 3DTrans open-source project, which won the championship of the Waymo International Challenge and accumulated over 5k stars. Additionally, he is committed to promoting the rapid application of multimodal large language models in various scenarios, such as scientific document understanding, scientific research surveys, mathematical reasoning, and autonomous driving.



Hancheng Ye received his B.S. and M.S. degrees in Electronic Engineering at the School of Information Science and Technology from Fudan University. He is currently pursuing a Ph.D. degree at Duke University. His primary research interests focus on efficient machine learning, model compression, and multimodal learning.

Baopu Li obtained his Ph.D degree in the field of computer vision and robotics from the Chinese University of Hong Kong in the year 2008. After that, he was in the academic field for another 8 years. Since 2016, he changed his career path to industry in the USA. His major research and development interests include Auto machine learning (ML), low-level image processing, video understanding and so on together with their applications on cloud and edge devices.



Yancheng Cai is a second-year PhD student in Computer Science and Technology at the University of Cambridge, specializing in artificial intelligence, computer vision, computer graphics, and robotics. He completed his undergraduate studies at Fudan University in the Excellence Class of Intelligent Science and Technology, ranking first in his engineering department for four consecutive years. He has received two National Scholarships, the Fudan University Top Ten Students award, numerous individual awards for academic and research excellence, as well as the College Star and Shanghai Outstanding Graduate titles. Formerly a Research Assistant with the Fudan's Embedded Deep Learning Laboratory, he continued as a Research Assistant with Stanford's Computer Vision Laboratory in 2022. As the first author, he has published academic papers in CVPR 2023, SIGGRAPH Asia 2024, T-IP, and Neurocomputing. Additionally, he has served as a reviewer for top conferences such as CVPR 2022, CVPR 2023, ECCV 2022, and ECCV 2024, and as a reviewer for T-PAMI. For more information visit the link (https://caiyancheng.github.io/academic.html).

Tao Chen (Senior Member, IEEE) received the Ph.D. degree in information engineering from Nanyang Technological University, Singapore, in 2013. He was a Research Scientist with the Institute for Infocomm Research, A*STAR, Singapore, from 2013 to 2017, and a Senior Scientist with the Huawei Singapore Research Center from 2017 to 2018. He is currently a Professor with the School of Information Science and Technology, Fudan University, Shanghai, China. His main research interests include computer vision and machine learning.