Pyramid Fusion Transformer for Semantic Segmentation

Zipeng Qin, Jianbo Liu, Xiaolin Zhang, Maoqing Tian, Aojun Zhou, Shuai Yi, and Hongsheng Li

Abstract—The recently proposed MaskFormer [Cheng et al. (2021)] gives a refreshed perspective on the task of semantic segmentation: it shifts from the popular pixel-level classification paradigm to a mask-level classification method. In essence, it generates paired probabilities and masks corresponding to category segments and combines them during inference for the segmentation maps. In our study, we find that the per-mask classification decoder on top of a single-scale feature is not effective enough to extract reliable probabilities or masks. To mine for rich semantic information across the feature pyramid, we propose a transformer-based Pyramid Fusion Transformer (PFT) for per-mask approach semantic segmentation with multi-scale features. The proposed transformer decoder performs cross-attention between the learnable queries and each spatial feature from the feature pyramid in parallel and uses cross-scale inter-query attention to exchange complementary information. We achieve competitive performance on three widely used semantic segmentation datasets. In particular, on ADE20K validation set, our result with Swin-B backbone surpasses that of MaskFormer with a much larger Swin-L backbone in both single-scale and multi-scale inference, achieving 54.1 mIoU and 55.7 mIoU respectively. Using a Swin-L backbone, we achieve single-scale 56.1 mIoU and multi-scale 57.4 mIoU, obtaining state-of-the-art performance on the dataset. Extensive experiments on three widely used semantic segmentation datasets verify the effectiveness of our proposed method.

Index Terms—Multi-scale features, semantic segmentation, transformer.

I. INTRODUCTION

The goal of semantic segmentation is to assign each pixel of an image with a semantic class label. Over the past decade, encoder-decoder based methods are the mainstream models to address this task [2], [3], [4], [5], [6]. They usually use convolution- [2], [7] or transformer-based [8], [9] networks to produce dense predictions from deep features generated by encoder networks. Efforts have been made to either design stronger backbone encoders [9], [10], [11], [12] or decoders [13], [14], [15], [16], [17], [18] for the dense prediction task.

Following FCN [2], most of the encoder-decoder based semantic segmentation methods [3], [4], [18], [19] follow a per-pixel classification paradigm to address the task: the outputs of the models are spatial segmentation maps with categorical assignment for each pixel. The optimization happens at each pixel location, usually through a cross-entropy loss. Recently, Cheng et al. took inspiration from DETR [13] in object detection and proposed MaskFormer [1] to advocate a path-breaking replacement for the per-pixel formulation. It adopts a novel per-mask classification method for semantic and panoptic segmentation. Instead of directly producing segmentation maps, a set of paired probabilities and masks are predicted corresponding to category segments in the input image. To this end, MaskFormer uses a transformer decoder to obtain probability-mask pairs with a set of learnable object queries using softmax activation and sigmoid activation respectively. Each object query is associated with a probability-mask pair and extracts useful information from encoder features through cross-attention [8]. The probabilities and masks are end-to-end optimized through a classification loss and a mask loss separately. During inference, the probability-mask pairs are combined to produce the segmentation maps.

Despite MaskFormer’s use of transformer module to integrate global information, we empirically observe that it still lacks the capability to produce fine details and is inclined to make contextual mistakes when decoding based on a single-scale feature map. We train MaskFormer with a Swin-T [10] backbone on top of the spatial feature maps of 1/8, 1/16, 1/32 input resolutions respectively (Table I column 1, 2, and 3). As shown in Fig. 1, it makes mistakes due to the lack of either fine-grained information or global context when decoding on a single-scale feature map. However, neither increasing the model capacity nor using multi-scale features for the decoder are trivial or necessarily lead to an increase of segmentation accuracy (see Table I column 4 and 5 and Section IV-B). A naive attempt at using multi-scale features is to perform self-attention on the patch tokens of all scales. However, such an approach does not necessarily lead to performance gain and is computationally overwhelming as the computational cost increases quadratically with the number.
of tokens (see Section IV-C). We propose a simple yet elegant multi-scale transformer decoder, which efficiently propagates and aggregates multi-scale information from a feature pyramid for more accurate segmentation. Specifically, we use a set of learnable object queries and stacked transformer layers on top of each multi-scale feature to produce probability-mask pairs parallelly and average the per-scale predictions for the final output. The cross-scale communication is achieved via the proposed cross-scale inter-query attention mechanism, which uses queries not only for information summarization within each scale but also for communication across scales. The small number of query tokens makes the cross-scale communication efficient. Such a mechanism also allows queries of different scales to be aware of semantic information of other scales without directly computing on them.

In addition, we observe that the cross-attention weights are naturally correlated with the category segments. As shown in Fig. 2, we visualize the cross-attention weights for the same object query at different training iterations and observe that they gradually concentrate more on the concerned objects as the training progresses. Therefore, we aim at enhancing the correlation and propose a novel loss to directly supervise the cross-attention weights with the ground-truth category masks to facilitate and stabilize the training process.

With the above proposed ingredients, we name our solution Pyramid Fusion Transformer (PFT), a segmentation transformer that efficaciously reasons from multi-scale feature maps with enhanced latent representations and consolidates predictions with high fidelity. To demonstrate the effectiveness of our method, we conduct extensive experiments and ablation studies on three widely-used ADE20K [20], COCO-Stuff-10K [21], and PASCAL Context [22] datasets and achieve state-of-the-art results on all the benchmarks.

In summary, our contributions are threefold.

- We propose a Pyramid Fusion Transformer that effectively aggregates information from multi-scale feature pyramid to improve segmentation accuracy. The multi-scale communication and aggregation is efficiently achieved via a novel cross-scale inter-query attention mechanism.
- Observing the strong correlation between the cross-attention map and the final segmentation map for each category, we apply a direct supervision on the cross-attention weights between class queries and spatial features.
- Our proposed PFT achieves state-of-the-art performances on ADE20K and COCO-Stuff-10K datasets with extensive ablation studies validating the effectiveness of different ingredients.

II. RELATED WORKS

Vision Transformer: With the inspiration from success stories of transformer and attention mechanism [8] in natural language processing tasks [23], the vision community enjoys a recent surge of interests in adapting the transformer structures into solving various vision tasks [11], [13], [24], [25], [26]. The pioneering work ViT [11] first proposes to divide an image input into a set of $16 \times 16$ patches, and sends them to a series of transformer layers consisting of self-attention modules and fully-connected layers with skip connections. On the task of image classification, it achieves competitive results with popular CNN networks.
proposes a sophisticated and Out-of-features to better align feature fusion at each spatial scale. It adopts as backbone encoder-decoder based per-pixel classification semantic segmentation framework has been the dominating paradigm for the task [3, 4, 16, 40, 41]. They usually encode an image into deep spatial features of different sizes using an encoder backbone network and use a decoder network to produce a spatial segmentation map of the same resolution as the input image. The encoder is usually a neural network pretrained on a large-scale dataset [42] and generates a feature pyramid rich with both fine-grained and contextual information. The decoder is another neural network that uses the multi-scale features to extract dense information and restores the original image resolution to predict a categorical label for each spatial location. During training, the encoder and decoder are end-to-end optimized, usually through a classification loss applied at each spatial location. Particularly, FCN [2] adopts as backbone encoder a CNN trained on ImageNet [42] classification and discard the final fully connected layer to keep the spatial feature map. Deconvolution/up-sampling layers are used to generate segmentation maps with the same resolution as the input image. Following the paradigm of FCN, improvements have been made to better aggregate semantic information in deep neural networks. For example, PCAA [43] proposes a sophisticated partial CAM module to obtain intra-class semantic information under different context. Ma et al. propose an adaptive attention module to enhance the semantic information across feature scales, it is beneficial to utilize information across different spatial scales, it is beneficial to utilize information across different spatial scales.

**Multi-scale feature representation:** As the semantic information at different scales can better capture different objects or structures, it is beneficial to utilize information across different feature scales. [3, 30, 31, 32]. [30] shows that effectively processing multi-scale features generated from a hierarchical backbone network can improve performance for many segmentation and detection frameworks. Its success has inspired numerous works to adopt a pyramidal feature representation design and to mine helpful ingredients from multi-scale features. [3, 7, 14, 31, 33, 34, 35, 36]. For instance, Efficient-FCN [3] encodes a set of semantic codewords by down-sampling all multi-scale features to 1/32 the input resolution to capture strong global context information from multi-scale features. It then produces assembly coefficients from concatenated features scaled to 1/8 resolution and linearly combine the codewords to produce the segmentation maps. Semantic FPN [31] adds an extra top-down path that connects all levels in the feature pyramid into a single 1/4 resolution feature map and generate segmentation through up-sampling. FaPN [36] further improves the fine-grained multi-scale representation by using deformable convolution [37] to better align feature fusion at each spatial location and achieves improved performance for segmentation tasks compared to FPN [30]. Multi-scale feature has also been proven to be beneficial to Few-Shot Learning [38] and Out-of-distribution (OOD) Detection [39]. For example, utilizing multi-scale features improves the quality of support samples in [38] by filtering out the noises in images and labels through focusing on local region representations and global image representations respectively. In [39], global and local representations are used to calculate pairwise distances so as to effectively find “unknown” data. Contrary to their motivations, our proposed method uses multi-scale features to more accurately capture the both the local details and the global context for better segmentation.

**Per-pixel classification semantic segmentation:** Starting from the seminal work Fully Convolutional Network (FCN) [2], encoder-decoder based per-pixel classification semantic segmentation framework has been the dominating paradigm for the task [3, 4, 16, 40, 41]. They usually encode an image into deep spatial features of different sizes using an encoder backbone network and use a decoder network to produce a spatial segmentation map of the same resolution as the input image. The encoder is usually a neural network pretrained on a large-scale dataset [42] and generates a feature pyramid rich with both fine-grained and contextual information. The decoder is another neural network that uses the multi-scale features to extract dense information and restores the original image resolution to predict a categorical label for each spatial location. During training, the encoder and decoder are end-to-end optimized, usually through a classification loss applied at each spatial location. Particularly, FCN [2] adopts as backbone encoder a CNN trained on ImageNet [42] classification and discard the final fully connected layer to keep the spatial feature map. Deconvolution/up-sampling layers are used to generate segmentation maps with the same resolution as the input image. Following the paradigm of FCN, improvements have been made to better aggregate semantic information in deep neural networks. For example, PCAA [43] proposes a sophisticated partial CAM module to obtain intra-class semantic information under different context. Ma et al. propose an adaptive attention module to enhance the semantic information across feature scales. Since the original ViT adopts an isotropic structure and can only produce feature maps of the same resolution, it is unsuitable for many downstream tasks that require multi-scale features. Moreover, the computational cost of patch tokens self-attention increases quadratically to the image size and quickly becomes overwhelming when ViT is applied to dense tasks that require high resolution inputs. To better adapt the transformer structure to dense vision tasks, researchers have designed various variants with stage-by-stage feature maps with shrinking resolutions [9, 10, 27, 28, 29]. The improved transformers are both more efficient and accurate than the vanilla version when evaluated on downstream vision tasks, such as object detection and segmentation. In stead of performing self-attention on the entire set of patch tokens, it uses window partitions as constraints on where the attention mechanism is applied and allows communication among windows by using a shifted configuration. Interleaved patch merging module is used to reduce the feature resolution stage-by-stage. As a result, it achieves high performance in several downstream vision tasks with reduced computational costs and memory consumption.

![Fig. 2. Cross-attention weights and binary masks. Visualizations obtained from the same model with and without our attention weight loss at different training iterations. Values are normalized to [0, 1] range. Images are from the ADE20K validation set.](image-url)
channels by learning the low and high frequency components. One of the recent works SegFormer [17] designs a transformer backbone that generates multi-scale features and uses an MLP decoder to predict segmentation results from the concatenated features. Despite the use of transformer architecture in its architecture, SegFormer still follows the per-pixel semantic segmentation paradigm that predicts and optimizes classification at each spatial location.

**Per-mask classification semantic segmentation:** MaskFormer [1] and Max-DeepLab [44] are among the pioneering works to use a mask classification paradigm in place of the end-to-end per-pixel classification segmentation approach. Contrary to FCNs that predict segmentation maps with per-pixel labelings, they produce paired results for masks and their corresponding class labels. To this end, they usually interact a set of object queries with the backbone features through stacked transformer layers and produce a probability-mask pair associated with each query. In particular, Max-DeepLab adds a transformer module to each backbone convolutional block and performs self/cross-attention to allow communications between the backbone and the decode head. MaskFormer extracts semantic information from 1/32 resolution feature through cross-attention between object queries and feature. The probabilities are generated by applying a linear layer on the queries and the masks are produced through dot-product between the queries and 1/4 resolution feature. Recently, Li et al. [34] proposes Panoptic Segformer, which shares a similar per-mask classification concept for panoptic segmentation on top of multi-scale feature maps. Specifically, it first applies a transformer decoder with sparse attention [14] on the multi-scale features and uses the output queries to predict bounding box locations. The location-aware queries are then passed to a second transformer to perform dense cross-attention [8] again with the multi-scale features. The predicted masks are then drawn from the attention weights between the queries and the spatial features and the probabilities are predicted in a separate branch. During inference, the masks and probabilities are combined to generate the segmentation maps.

**Cross-attention in Weakly Supervised Semantic Segmentation:** Weakly Supervised Semantic Segmentation (WSSS) aims to learn a segmentation model using image-level annotations only. To this end, it usually focuses on improving the localization of object categories through exploiting a learned image-level classifier. With the introduction of cross-attention mechanism, researchers devise WSSS frameworks that adopt a set of queries rich in semantic information based on the outputs from the image-level classifier. This leads to improved discriminative power to better differentiate pixels belonging to different categories [45], [46]. Particularly, [45] receives paired images with the same image-level label as the input and uses two cross-attention modules to learn cross-image common semantics and unshared semantics respectively. The attention weights from the cross-attention modules are then used to generate more accurate localization maps for better segmentation. [46] further improves the semantic information learned by the model by using dictionary learning to maintain a set of queries. With a similar spirit, Lin et al. [47] design a Query-Memory Aggregation (QMA) module to use cross-attention to aggregate semantic information for different categories among different frames and apply their novelty in Weakly-supervised Video Object Segmentation (WVOS). Contrary to the use of cross-attention weights to localize category regions, we use a cross-attention weight loss to accelerate the convergence of the cross-attention modules using groundtruth pixel-level labels. This is based on the observation from the difficulties of training the cross-attention that operates on the whole spatial feature maps. In Section I of the supplementary material, we show the the use of this loss applied on the cross-attention weights can make the attention modules focus more on the relevant regions, and therefore lead to improved segmentation quality.

### III. Method

#### A. Background: Per-Mask Classification Paradigm for Semantic Segmentation

MaskFormer [1] is one the pioneering works that adopt a per-mask classification paradigm to address segmentation tasks. The core design of such model is a DETR [13]-like architecture that interacts a set of queries with the backbone features through stacked layers of self-attention and cross-attention [8]. The interacted queries then take a dot product spatially with the backbone feature of 1/4 the input image resolution to produce a set of object masks. Another linear layer is applied on each object query to predict its category. With this novel per-mask classification paradigm and use of strong transformer-based architecture, MaskFormer achieves state-of-the-art across semantic, instance, and panoptic segmentation tasks of various public datasets with a unified architecture [1].

In our work, we follow the principal design of this per-mask classification framework with an emphasis on the semantic segmentation task. Various novel designs are introduced below to improve the performance of per-mask classification semantic segmentation framework. With the proposed improvements, our model can achieve state-of-the-art results on widely used semantic segmentation datasets, surpassing MaskFormer by healthy margins.

#### B. Overview

Our overall pipeline is shown in Fig. 3. Our proposed Pyramid Fusion Transformer (PFT) takes a feature pyramid encoded by a backbone network as input and adopts a novel multi-scale transformer decoder with a cross-scale inter-query attention mechanism to efficiently fuse multi-scale information for accurate semantic segmentation. The backbone network, which can either be a convolutional or transformer network, receives an input image of size $H \times W \times 3$ and produces a hierarchy of feature maps $\{P_4, P_8, P_{16}, P_{32}\}$ by a Feature Pyramid Network (FPN) with a uniform channel dimension $C$ and spatial dimension $\frac{H}{2} \times \frac{W}{2}$, where $s = 4, 8, 16, 32$. Our PFT is applied on top of the three scales of feature maps $\{P_8, P_{16}, P_{32}\}$ as sequences of pixel tokens. Each scale has a separate set of $K$ queries to estimate the confidences and locations of $K$ semantic categories, where each query is only responsible for capturing
semantic information of one assigned category. Within the transformer, we recurrently stack three types of attention layers: (1) an intra-scale query self-attention layer that conducts conventional self-attention between queries of the same scale, (2) a novel cross-scale inter-query attention layer to efficiently communicate scale-aware information using the limited number of $3\times$ queries of the three different scales, and (3) an intra-scale query-pixel cross-attention layer that aggregates semantic information from flattened sequences of pixel tokens.

C. Pyramid Fusion Transformer With Cross-Scale Inter-Query Attention

Multi-scale information is important for achieving accurate scene understanding. Low-resolution feature maps are able to capture global context while high-resolution ones are better at discovering fine structures such as category boundaries [31], [41], [48], [49]. It is therefore vital to propagate information across the multiple scales to capture both global and find-grained information. However, due to the high computational cost of directly applying attention on the large number of multi-scale pixel tokens, previous transformer-based semantic segmentation methods such as [15], [34] often rely on the sparse attention mechanism on the pixel tokens to model pixel-to-pixel relations. Contrary to their approaches, to avoid heavy computation, we propose to efficiently fuse the multi-scale information with our proposed cross-scale inter-query attention mechanism. Instead of extracting global and local semantic information among the pixel tokens, we propose to fuse the multi-scale information in the query embedding space. Three types of attention layers are recurrently stacked in our PFT to achieve the goal.

Intra-scale query self-attention: Within each scale, the intra-scale query self-attention layer conducts self-attention only between category queries within the scale. Specifically, for each scale, the $K$ category queries $Q_s \in \mathbb{R}^{K \times C}$ with learnable positional encodings $P_s \in \mathbb{R}^{K \times C}$ are input into the layer.

$$Q_s, K_s = \text{Projection}(Q_s + P_s), \quad V_s = \text{Projection}(Q_s),$$

where $\text{Projection}(\cdot)$ represents a linear projection layer and $\text{Attention}(\cdot)$ is the multi-head attention module introduced in [8]. For this intra-scale query attention, we use separate parameters for different scales. The category queries $Q_s$ are learnable and randomly initialized before training. In the forward pass, they are updated by the stacked attention layers, while the learnable positional encodings $P_s$ are shared and added to the category queries before being input into each stacked layer. Following [11], [13], [14], the positional encodings are only used to encode the embeddings $Q_s, K_s$ but not $V_s$ for self-attention. The self-attention is conducted only between queries from the same scale to obtain the updated queries $Q_s$. Such an intra-scale self-attention layer consists of the commonly used multi-head self-attention sub-layer and Feed-Forward Network (FFN) sub-layer with layer normalization and residual connection.

Cross-scale inter-query attention: As the above intra-scale self-attention is limited within each scale, their three sets of updated queries can only obtain information specific to each scale. To allow information propagation between the multiple scales for knowledge fusion, a novel cross-scale inter-query attention layer is introduced. As the number of such scale-aware queries in each scale is much smaller than the number of all visual tokens in each scale, we achieve information propagation across the multiple scales by conducting attention between the concatenated $3K$ category queries $Q_{all} = \text{Concat}(Q_{8}, Q_{16}, Q_{32}) \in \mathbb{R}^{3K \times C}$ of the three scales. The inter-query attention results are then sliced

We use the same notations to represent linear projection layers as $\text{Projection}(\cdot)$ and multi-head attention module as $\text{Attention}(\cdot)$ in the equations in the later parts of this paper as well.
back to the queries of the three scales, which serve as the input for the follow-up intra-scale query-pixel cross-attention layer. To distinguish queries of different scales, we use the learnable positional encodings \( \mathcal{P}_{all} = \text{Concat}(\mathcal{P}_8, \mathcal{P}_{16}, \mathcal{P}_{32}) \) from the intra-scale query self-attention layer.

\[
Q, K = \text{Projection}(Q_{all} + \mathcal{P}_{all}), \\
V = \text{Projection}(Q_{all}), \\
Q_8, Q_{16}, Q_{32} = \text{Attention}(Q, K, V)
\]  

(2)

where the attention outputs a sequence of category queries of length \( 3K \) and are sliced to sub-sequences of length \( K \) and assigned to each \( Q_s \) for \( s = 8, 16, 32 \). In this way, the cross-scale attention and information communication are efficiently achieved by using the small number of category queries. The proposed cross-scale inter-query attention layer also consists of a multi-head attention sub-layer and an FFN sub-layer with layer normalization and residual connection following the classical design of the dot-product attention. The parameters of the layer are shared across all scales, since the layer receives the concatenated category queries as the input.

**Intra-scale query-pixel cross-attention:** To aggregate dense pixel-level semantic information to the category queries, we conduct dense query-pixel cross-attention within each scale. Within each scale, the category queries \( Q_s \) perform cross-attention with the pixel tokens \( P_s \) for \( s = 8, 16, 32 \) using the multi-head attention.

\[
Q_s = \text{Projection}(Q_{s} + \mathcal{P}_s), \\
K_s = \text{Projection}(P_s + \mathcal{P}_{s}^{\text{sine}}), \\
V_s = \text{Projection}(P_s), \\
Q_s = \text{Attention}(Q_s, K_s, V_s), \text{ for } s = 8, 16, 32
\]  

(3)

where the same learnable positional encodings \( \mathcal{P}_s \) used in the above two attention layers are added to the category queries \( Q_s \), and fixed sinusoidal positional encodings \( \mathcal{P}_{s}^{\text{sine}} \) are added to the pixel tokens \( P_s \) following [13]. Separate parameters are used for different scales for this query-pixel cross-attention layer.

As illustrated by Fig. 3, in our pyramid fusion transformer, each transformer layer consists of the above three types of attention layers and is stacked for \( L \) times to form an \( L \)-layer transformer. By default, we use \( L = 6 \) layers. For intra-scale query attention and intra-scale query-pixel cross-attention, we use separate weights for the linear projection layers, layer normalization, etc. for each scale. The proposed cross-scale inter-query attention layers are placed between the intra-scale query self-attention layers and the intra-scale query-pixel cross-attention layers. The small number of category queries serve as the bridges to efficiently aggregate and propagate the pixel-level semantic information across the multiple scales. Neither intra-scale nor cross-scale pixel-to-pixel attention is used in our PFT, avoiding heavy computational cost of dense multi-scale information fusion.

**Generating Segmentation Maps:** After the multiple transformer layers, the updated three sets of category queries \( \{Q_s \in \mathbb{R}^{K \times C}\}_{s=8,16,32} \) with multi-scale information can be used to generate the probability-mask pairs for segmentation. To generate \( K \) probabilities of the \( K \) categories for the three scale, we use three separate linear projections to map the category queries \( \{Q_s\}_{s=8,16,32} \) at the output of our PFT to the three sets of \( K \) probability logits \( \{L_{s, \text{prob}} \in \mathbb{R}^K\}_{s=8,16,32} \), each of the \( K \) logits representing the probability of the corresponding semantic category. The averaged logits \( L_{s,\text{prob}} \) of the three scales are followed by a sigmoid activation layer to generate the binary probabilities \( p \in \mathbb{R}^K \), which denotes the confidence of each category existing in the input image.

\[
L_{s, \text{prob}} = \text{Linear}(Q_s), \\
L_{\text{prob}} = \frac{\sum_{s=8}^{32} L_{s, \text{prob}}}{3}, \\
p = \text{sigmoid}(L_{\text{prob}}).
\]  

(4)

To generate the binary category masks for the multiple scales, we first apply a \( 3 \times 3 \) convolutional layer on the backbone feature \( P_{l} \in \mathbb{R}^{C \times H \times W} \) to produce the mask feature \( M \in \mathbb{R}^{C \times H \times W} \) for each category. Then, they perform a matrix product with the mask feature \( M \) to produce the mask logits \( L_{\text{mask}} \in \mathbb{R}^{K \times C \times H \times W} \). The averaged logits \( L_{\text{mask}} \) of the three scales are followed by a sigmoid activation layer to obtain the mask probability maps \( m \in \mathbb{R}^{K \times \frac{H}{4} \times \frac{W}{4}} \) of each category.

\[
L_{s, \text{mask}} = \text{MLP}(Q_s) \otimes M, \\
L_{\text{mask}} = \frac{\sum_{s=8}^{32} L_{s, \text{mask}}}{3}, \\
m = \text{sigmoid}(L_{\text{mask}}),
\]  

(5)

where \( L_{s, \text{mask}}, L_{\text{mask}} \) and \( m \in \mathbb{R}^{K \times \frac{H}{4} \times \frac{W}{4}} \) and \( \otimes \) denotes matrix multiplication. The matrix multiplication \( L_{\text{mask}} = \text{MLP}(Q_s) \otimes M \) is performed between the \( K \times C \) tensor \( \text{MLP}(Q_s) \) and the \( C \times (\frac{H}{4} \times \frac{W}{4}) \) tensor \( M \), and the result \( L_{\text{mask}} \) is therefore of shape \( K \times \frac{H}{4} \times \frac{W}{4} \). Note that we use sigmoid activation here since we do not enforce the masks of different categories to be exclusive to each other, following the practice in [1].

For each category \( k \)'s probability and mask, \( \{(p_k, m_k) | p_k \in \mathbb{R}, m_k \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4}}\}_{k=1}^{K} \), \( p_k \) is the \( k \)-th entry (class) of \( p \), \( m_k \) is the \( k \)-th channel (class) of \( m \). The semantic segmentation map is obtained by probability-mask marginalization as that in MaskFormer [1], where the category prediction for pixel \((h, w)\) is computed as

\[
\text{ClassPrediction}(h, w) = \arg\max_{k \in \{1, \ldots, K\}} p_k \cdot m_k(h, w),
\]  

(6)

where the masks of different categories are weighted by the predicted probabilities of their existence.

**D. Cross-Attention Weight Loss for Stabilizing Training**

For an input image with \( N \) categories present, we decompose the groundtruth segmentation map into \( N \) groundtruth
label-mask pairs. Since |N| is usually smaller than the total number of semantic categories $K$, we pad the ground-truth set with “not-exist” category $\emptyset$. This results in a set of $K$ padded groundtruth label-mask pairs $\{(c_k, m^\mathbb{P}_k) | c_k \in \{1, \ldots, K, \emptyset\}, m^\mathbb{P}_k \in \{0, 1\}^{H \times W}\}_{k=1}^K$, where $c_k = k$ and the binary mask $m^\mathbb{P}_k$ corresponds to all pixels belonging to the category if the $k$-th category is present in the image. If the $k$-th category is absent, $c_k = \emptyset$ and we do not have a non-existing category binary mask for it.

**Cross-attention weight loss**: Recent studies \cite{50, 51, 52} suggest that transformer-based models for vision dense tasks suffer from the difficulty of optimizing cross-attention between queries and spatial features and propose various ways to optimize cross-attention. We observe that the attention weights between the queries $Q_s$ and spatial features $P_s$ for transformer-based per-mask semantic segmentation have strong correlation with the spatial arrangements of the semantic maps (see Fig. 2). To better guide the query-pixel cross-attention during the training process, we explicitly enforce a novel attention weight loss on the cross-attention maps. Specifically, we obtain the attention weights $W_s \in \mathbb{R}^{K \times H \times W}$ from the Attention operation in (3)

$$W_s = \text{softmax}\left(Q_sK_s^T / \sqrt{C}\right), \text{ for } s = 8, 16, 32 \quad (7)$$

where $Q_s$ and $K_s$ are obtained from (3). Each channel $w_k \in \mathbb{R}^{H \times W}$ of $W_s$ represents the spatial attention weight between the $k$-th query and the spatial pixel tokens. Here, we omit the multi-head attention notation for simplicity, for which we take the average of logits across all heads as the input to the softmax layer.

We normalize the groundtruth binary mask $m^\mathbb{P}_k$ to form a probability distribution (i.e., values sum to 1) as the supervision target. Additionally, we uniformly use a uniform probability (i.e., all values equal $s^2 / HW$) to supervise the non-existing category’s attention weight. A cross-entropy loss naturally follows to fit to the target distribution.

$$L_{\text{attn}} = \lambda_{\text{attn}} L_{\text{ce}}\left(w_k, \text{Norm}\left(m^\mathbb{P}_k\right)\right), \text{ for } s = 8, 16, 32 \quad (8)$$

where we abuse the notation for groundtruth binary mask $m^\mathbb{P}_k$ a bit for absent categories. Norm(·) is the normalization described above. $\lambda_{\text{attn}}$ is the loss weight. An extra 0.1 weight multiplier is applied if $c_k = \emptyset$ to balance positive and negative samples. The final attention weight loss is averaged across all categories and scales.

We empirically find the formulation for non-existing categories’ attention weights helps reduce the variance of the performance. A more detailed experiment about our attention weight loss can be found in Section IV-C.

**Overall training loss**: Similar to MaskFormer, we use a mask loss $L_{\text{mask}}$ on the predicted masks $m$. The mask loss follows \cite{11} and consists of a binary focal loss \cite{53} $L_{\text{focal}}$ and a dice loss \cite{54} $L_{\text{dice}}$, where $L_{\text{mask}} = \lambda_{\text{focal}} L_{\text{focal}} + \lambda_{\text{dice}} L_{\text{dice}}, \lambda_{\text{focal}}$ and $\lambda_{\text{dice}}$ are hyperparameters balancing the two terms. Note that the mask loss is only applied at the averaged mask logits $L_{\text{mask}}$ from the three scales and optimizes masks with groundtruth categories only. Masks corresponding to $\emptyset$ are simply discarded during training.

Our classification loss $L_{\text{cls}}$ consists of two terms: a binary cross-entropy loss $L_{\text{ce}}$ applied at the averaged probability logits $L_{\text{prob}}$ from the three scales and a focal-style \cite{53} binary cross-entropy loss $L_{\text{focal-ce}}$ at each decoder outputs $L_{\text{prob}}^s$ to adaptively reweight the hard samples following \cite{53}. The formulation of the focal-style cross-entropy is described in the supplementary. The classification loss $L_{\text{cls}}$ is a linear combination of the above two losses $L_{\text{cls}} = \lambda_{\text{ce}} L_{\text{ce}} + \lambda_{\text{focal-ce}} L_{\text{focal-ce}}$, where $\lambda_{\text{ce}}$ and $\lambda_{\text{focal-ce}}$ are hyperparameters balancing the two terms.

Our final training loss $L_{\text{train}}$ is a sum of the classification loss, the mask loss, and the attention weight loss,

$$L_{\text{train}} = L_{\text{cls}} + L_{\text{mask}} + L_{\text{attn}} \quad (9)$$

Similar to DETR \cite{13} we apply supervision to each transformer layer’s output queries. Besides, supervision is also applied to the input learnable queries $Q_s$ before any transformer layer.

**IV. EXPERIMENTS**

In this section, we demonstrate the effectiveness of our method with competitive semantic segmentation results and compare to both state-of-the-art per-pixel classification and mask-level classification frameworks on three popular segmentation datasets, ADE20K \cite{20}, COCO-Stuff-10K \cite{21}, and PASCAL-Context \cite{22}. We choose MaskFormer \cite{1} as our baseline model because of its strong performance among the mask-level classification methods \cite{1, 34, 44}. In the ablations, we further study the effectiveness of our proposed components, including usage of multi-scale features, cross-scale inter-query attention, and the design of attention weight loss. Experimental results demonstrate that our model can learn useful information from multi-scale feature maps to deliver high quality segmentation maps with our proposed multi-scale transformer decoder and optimization for query-pixel cross-attention.

**A. Datasets and Implementation Details**

**Datasets**: ADE20K \cite{20} is a semantic segmentation dataset with 150 fine-grained semantic categories, including thing and stuff. It contains 20,210 images for training, 2,000 images for validation and 3,352 images for testing. COCO-Stuff-10K \cite{21} is a scene parsing dataset with 171 categories, not counting the class “unlabeled”. We follow the official split to partition the dataset into 9k images for training and 1k images for validation. PASCAL-Context \cite{22} contains pixel-level annotations for the whole scenes with 4,998 images for training and 5,105 images for testing. We evaluate our method on the commonly used 59 classes of the dataset.

**Implementation details**: We use the open-source segmentation codebase mmsegmentation \cite{55} to implement PFT. We adopt Swin Transformer \cite{10} and ResNet \cite{12} as backbone networks for evaluation. For ResNets, we report results obtained with ResNet-50 and ResNet-101, along with its slightly modified version ResNet-101c. ResNet-101c has its $7 \times 7$ stem convolution layer replaced by three consecutive $3 \times 3$ convolutions, which is a protocol widely adopted in semantic segmentation methods \cite{4, 18, 41, 56, 57, 58}.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
**Table II**

| method | backbone | crop size | schedule | mIoU (s.s.) | mIoU (m.s.) | params. | FLOPS |
|--------|----------|-----------|----------|-------------|-------------|---------|-------|
| OCRNet [60] | R101c | 512×512 | 160k | - | - | 45.3 | - |
| GRNet [61] | R101c | 542×542 | 200k | - | - | 47.1 | - |
| DeepLabV3+ [18] | R50 | 512×512 | 160k | 44.0 | 44.9 | 44M | 177G |
| | R101c | 512×512 | 160k | 45.5 | 46.4 | 63M | 255G |
| MaskFormer [1] | R50 | 512×512 | 160k | 44.5 | 46.7 | 41M | 53G |
| | R101 | 512×512 | 160k | 45.5 | 47.2 | 60M | 73G |
| | R101c | 512×512 | 160k | 46.0 | 48.1 | 60M | 80G |
| PFT (ours) | R50 | 512×512 | 160k | 45.5 (+1.0) | 47.9 (+0.9) | 62M | 61G |
| | R101 | 512×512 | 160k | 46.5 (+1.1) | 48.4 (+1.2) | 81M | 81G |
| | R101c | 512×512 | 160k | 47.8 (+1.8) | 49.8 (+1.7) | 81M | 82G |
| BEIT [62] | ViT-L | 640×640 | 160k | 56.7 | 57.0 | 441M | - |
| SETR [16] | ViT-L | 512×512 | 160k | 48.6 | 50.3 | 308M | - |
| MaskFormer [1] | Swin-T | 512×512 | 160k | 46.7 | 48.8 | 42M | 55G |
| | Swin-S | 512×512 | 160k | 49.8 | 51.0 | 63M | 79G |
| | Swin-B | 640×640 | 160k | 52.7 | 53.9 | 102M | 195G |
| | Swin-L | 640×640 | 160k | 54.1 | 55.6 | 212M | 375G |
| PFT (ours) | Swin-T | 512×512 | 160k | 45.7 (+2.0) | 50.1 (+1.3) | 63M | 64G |
| | Swin-S | 512×512 | 160k | 51.0 (+1.2) | 52.0 (+1.0) | 84M | 87G |
| | Swin-B | 640×640 | 160k | 54.1 (+1.4) | 55.7 (+1.8) | 123M | 206G |
| | Swin-L | 640×640 | 160k | 56.1 (+2.0) | 57.4 (+1.8) | 232M | 385G |

The bold values highlight the best performance metrics within each column.

**Training settings:** Models are trained on ADE20K, COCO-Stuff-10K, and PASCAL-Context with 160k, 60k, and 40k-iteration schedules respectively. For the ADE20K dataset, 512 × 512 images are cropped after scale jittering, horizontal random flip, and color jittering. The same data augmentations are used for COCO-Stuff-10K dataset and PASCAL-Context dataset, while a crop size of 640 × 640 is used for COCO-Stuff-10K and 480 × 480 for PASCAL-Context. We use a batchsize of 16, 32, and 16 for ADE20K, COCO-Stuff-10K, and PASCAL-Context respectively. Scale jittering is set to between 0.5 and 2.0 of the crop sizes. We set $\lambda_{\text{attn}} = 0.1$ for our attention weight loss and detach the loss at 3/4 of the total training schedule (see Section IV-C for our analysis). We choose $\lambda_{\text{focal-ce}}$ from $\{1.0, 2.0\}$ for all datasets and provide the ablations of the loss weights. AdamW [59] is used as our optimizer with a linear learning rate decay schedule. For ResNet backbones, they are pretrained on ImageNet-1K and we use a learning rate of $10^{-4}$ and a weight decay of $10^{-4}$. The learning rate for Swin-Transformer backbones is set to $6 \times 10^{-5}$ and a weight decay of $10^{-2}$ is used. For Swin-T and Swin-S backbones, we use the official pretrained weights on ImageNet-1K [42] with 224 × 224 resolution. For Swin-B and Swin-L, we use the official pretrained weights on ImageNet-22K with 384 × 384 resolution. All models are trained on a single compute node with 8 NVIDIA Tesla V100 GPUs. See the supplementary for a more details of hyperparameters and experimental settings.

**Evaluation settings:** We use mean Intersection-over-Union (mIoU) as our evaluation metric for semantic segmentation performance. Both the single-scale and multi-scale inferences are reported in our experiments. For multi-scale inference, we apply horizontal flip and scales of 0.5, 0.75, 1.0, 1.25, 1.5, and 1.75.

**B. Main Results**

**Results on ADE20K dataset:** Table II summarizes our results on ADE20K validation set. We report both results from single-scale and multi-scale inferences. As shown in the table, when paired with the Swin-T backbone, PFT achieves 48.7 mIoU, improving over MaskFormer [1] by 1.6 mIoU and matching the accuracy obtained by SETR [16] with a much larger backbone. Notably, with the Swin-B backbone, we obtain an mIoU of 55.7, surpassing MaskFormer with a much larger Swin-L backbone. Our best model achieves a 57.4 mIoU, obtaining state-of-the-art performance without any bells-and-whistles [36], [63]. Consistent improvements with CNN backbones can also be observed from the results.

**Results on COCO-Stuff-10K dataset:** We report our results on COCO-Stuff-10K dataset in Table III. As shown in the table, PFT obtains consistent improvements over MaskFormer with CNN backbones. A significant 3.7 mIoU improvement over MaskFormer is attained by using the Swin-T backbone, surpassing MaskFormer by a much larger Swin-L backbone. Our best model achieves a 57.4 mIoU, obtaining state-of-the-art performance without any bells-and-whistles [36], [63]. Consistent improvements with CNN backbones can also be observed from the results.

**Results on PASCAL-Context dataset:** We present our results from PFT trained on PASCAL-Context dataset in Table IV. Our
method beats MaskFormer with different backbone networks, showing steady improvements for the per-mask classification framework for semantic segmentation. Our most significant performance gain is obtained by the ResNet-101c backbone, which achieves 55.2 mIoU with only 18M additional parameters compared to MaskFormer, outperforming it by a 2.1 mIoU margin in single-scale testing. With multi-scale inference, we obtain 57.3 mIoU with ResNet-101c. To our knowledge, we have the best semantic segmentation performance with ResNet-101c backbone on PASCAL-Context dataset.

**Compare to concurrent work:** Mask2Former [65] was recently proposed as a multi-scale variant for MaskFormer [1]. Compared to our approach, Mask2Former extracts semantic information from multi-scale feature maps by using cross-attention in a round-robin fashion, i.e., the queries attend to the spatial features one by one in the cross-attention layers. Compared to our method, it uses a more sophisticated masked attention module based on predicted mask priors, and a stronger FPN than ours in Table IV.

| method | backbone | crop size | schedule | mIoU (s.s.) | mIoU (m.s.) | #params. |
|--------|----------|-----------|----------|-------------|-------------|----------|
| OCRNet [60] | R101c | 520/H | 60k | - | 39.5 | - |
| GRAN [61] | R101c | 544/H | 100k | - | 41.9 | - |
| MaskFormer [1] | RS0 | 544 | 60k | 37.1 | 38.9 | 44M |
| | R101 | 640 | 60k | 38.1 | 39.8 | 63M |
| | R101c | 640 | 60k | 38.0 | 39.3 | 63M |
| PFT (ours) | RS0 | 640 | 60k | 39.3 (+2.2) | 40.6 (+1.7) | 62M |
| | R101 | 640 | 60k | 39.6 (+1.5) | 41.5 (+1.7) | 81M |
| | R101c | 640 | 60k | 40.9 (+2.9) | 43.0 (+3.7) | 81M |

The bold values highlight the best performance metrics within each column.

**TABLE IV**

| method | backbone | crop size | schedule | mIoU (s.s.) | mIoU (m.s.) | #params. |
|--------|----------|-----------|----------|-------------|-------------|----------|
| SFPNet [64] | R50c | 512/H | 34k | - | 50.7 | - |
| | R101c | 512/H | 34k | - | 53.8 | - |
| GRAN [61] | R101c | 544/H | 50k | - | 55.7 | - |
| MaskFormer [1] | RS0 | 468 | 40k | 52.5 | 54.1 | 44M |
| | R101 | 468 | 40k | 53.7 | 55.4 | 63M |
| | R101c | 468 | 40k | 53.1 | 55.6 | 63M |
| PFT (ours) | RS0 | 468 | 40k | 53.5 (+1.0) | 55.0 (+0.9) | 62M |
| | R101 | 468 | 40k | 54.7 (+1.0) | 56.2 (+0.8) | 81M |
| | R101c | 468 | 40k | 55.2 (+1.7) | 57.3 (+1.7) | 81M |

The bold values highlight the best performance metrics within each column.

C. Ablation Studies and Analysis

To evaluate the effectiveness of the components in our multi-scale transformer decoder for semantic segmentation, we conduct ablations on the multi-scale design and the cross-scale inter-query attention layer. Furthermore, we provide the analysis of our attention weight loss and the effects of the removal of the loss at various training iterations. Unless otherwise specified, for all the experiments, we train our model with the Swin-T backbone on ADE20K dataset and detach our attention loss at 3/4 the training schedule (120k iterations) and report the single-scale mIoU on the validation set.

**Comparison with stronger baseline models:** To further study the improvements brought by our framework, we conduct experiments with different variants of the baseline model with the same Swin-T backbone and compare our performance. Specifically, we first train MaskFormer [1] with different single-scale features of spatial shapes 1/8, 1/16, 1/32 the input image resolution respectively, where using a 1/32 scale feature map corresponds to the original MaskFormer. As shown in Table I columns 1, 2, and 3, single-scale feature is insufficient to produce higher segmentation quality for the per-mask segmentation framework. We further augment the baseline by increasing the number of transformer layers from 6 to 8 and double the hidden dimension in transformer. The augmented baseline model matches obtained by [65] to ours in Table V. PFT achieves competitive performance with Mask2Former with fewer FLOPs even if Mask2Former uses a stronger FPN based on multi-scale deformable attention [14]. Our FPN follows the conventional design with lateral connections and group convolutions. In particular, PFT with the Swin-T [10] backbone outperforms Mask2Former by a 1.0 single-scale mIoU margin, while using 10 GFLOPs fewer than Mask2Former. When we replace the FPN with the one used by Mask2Former, we obtain 56.3 single-scale mIoU and 57.8 multi-scale mIoU with the Swin-L backbone, surpassing Mask2Former's result and matches that of Mask2Former with FaPN [36], a sophisticated Feature Pyramid Network specially designed for semantic segmentation.

**Results on ADE20K test set:** To demonstrate the superior performance of our proposed PFT, we additionally report the results on the test set of ADE20K [20] dataset. We adopt the same training settings for the experiments on the validation set, including pretrained checkpoints, input resolution, hyperparameters, etc., except that we train our models on the union of the training and validation sets of ADE20K as a common practice. The results are from submitting the multi-scale inference results on the testing set to the official evaluation server. As shown in Table VI, our PFT without any specialized FPNs achieves higher segmentation score than that obtained by Mask2Former with FaPN [65] (64.84 vs. 64.76) and surpasses all previous state-of-the-art methods. Replacing FPN with FaPN [36] in our framework pushes the mIoU to 50.63 and the segmentation score to 65.00. To our knowledge, it is the only method obtaining over 65.00 segmentation score on the ADE20K testing set with the Swin-L backbone.
the parameters and FLOPs of our PFT with the Swin-T backbone. As can be seen from Tab I column 4, the performance slightly increases but still lags a considerable margin behind our multi-scale approach. However, using multi-scale features is a non-trivial task: directly inputting concatenated multi-scale features into the baseline model achieves a 47.8 mIoU (Table I column 5), which is lower than our method with the same backbone by 0.9 mIoU and consumes a larger computation (88 GFLOPs vs. 64 GFLOPs).

Effects of cross-scale inter-query attention: In our framework, we propose a novel cross-scale inter-query attention module to allow the multiple scales to propagate and aggregate useful information to other scales. To verify the benefit by such a module, we conduct experiments to remove the cross-scale query attention. After such removal, our framework can be viewed as a multi-scale variant of MaskFormer [1] with fixed-matching between queries and categories. As shown in Fig. 4, with the Swin-T backbone, we observe a slight performance drop of 0.1 mIoU without the proposed module. With the Swin-B and the Swin-L backbones, the performance drops become more prominent (0.8 and 0.4 mIoU drops respectively). Moreover, significant performance drop of 1.3 mIoU is observed when paired with the ResNet-50 backbone. Noticeable performance drops are also observed with the ResNet-101 and the ResNet-101c backbones (0.4 and 1.1 mIoU drops respectively). We deduce that the differences in performance gains might be attributed to the intrinsic characteristics of transformer and CNN backbones, where transformers can more effectively aggregate global information across different scales. The proposed cross-scale inter-query attention can benefit more the CNN backbones by exchanging the global semantic information across different scales. The comprehensive experimental results show that our proposed cross-scale inter-query attention module is beneficial for different backbones in general.

Efficiency of cross-scale inter-query attention compared to pixel tokens self-attention: In our model, we propose to use the more efficient cross-scale inter-query attention module to propagate and aggregate information across scales instead of directly computing self-attention among the pixel tokens. In Table VII, we add comparison between our proposed cross-scale inter-query attention module with 1) pixel-token self-attention, and 2) removing the inter-query attention. For 1), we replace our cross-scale inter-query attention module with 1) pixel-token self-attention, and 2) removing the inter-query attention. For 1), we replace our cross-scale inter-query attention module with 1) pixel-token self-attention, and 2) removing the inter-query attention. For 1), we replace our cross-scale inter-query attention module with 1) pixel-token self-attention, and 2) removing the inter-query attention. For 1), we replace our cross-scale inter-query attention module with 1) pixel-token self-attention, and 2) removing the inter-query attention. For 1), we replace our cross-scale inter-query attention module with 1) pixel-token self-attention, and 2) removing the inter-query attention. For 1), we replace our cross-scale inter-query attention module with 1) pixel-token self-attention, and 2) removing the inter-query attention. For 1), we replace our cross-scale inter-query attention module with 1) pixel-token self-attention, and 2) removing the inter-query attention. For 1), we replace our cross-scale inter-query attention module with 1) pixel-token self-attention, and 2) removing the inter-query attention. For 1), we replace our cross-scale inter-query attention module with 1) pixel-token self-attention, and 2) removing the inter-query attention. For 1), we replace our cross-scale inter-query attention module with 1) pixel-token self-attention, and 2) removing the inter-query attention. For 1), we replace our cross-scale inter-query attention module with 1) pixel-token self-attention, and 2) removing the inter-query attention. For 1), we replace our cross-scale inter-query attention module with 1) pixel-token self-attention, and 2) removing the inter-query attention. For 1), we replace our cross-scale inter-query attention module with 1) pixel-token self-attention, and 2) removing the inter-query attention. For 1), we replace our cross-scale inter-query attention module with 1) pixel-token self-attention, and 2) removing the inter-query attention. For 1), we replace our cross-scale inter-query attention module with 1) pixel-token self-attention, and 2) removing the inter-query attention.

![Fig. 4. Ablations for cross-scale inter-query attention. Results obtained on ADE20K validation set.](image-url)
Attention weight loss: We propose to use a novel attention weight loss to guide the cross-attention layer to focus on locations in the feature maps corresponding to the groundtruth category segments as well as optimizing the attention weights for non-existing categories with a uniform distribution. As shown in Fig. 2 row 2, using our attention weight loss helps the cross-attention to concentrate more on the regions corresponding to the categories. Besides, we calculate the pearson correlation coefficient between the attention weights and the binary masks to give a quantitatively analysis of the effects of our proposed loss. As shown in Fig. 5 (dotted lines), our proposed loss significantly improves the correlation between attention weight maps and category masks at different training iterations. The improved correlations between attention weights and groundtruth binary masks lead to higher segmentation accuracy at different training iterations (see Fig. 5 solid lines).

To further study the improvements from the proposed loss, we ablate the loss weight and verify our design through experiments. Fig. 6 shows the influences of different $\lambda_{\text{attn}}$ for attention weight loss. The x-axis is drawn on a log scale.

**Table VII**  
**Efficiency of Our Cross-Scale Inter-Query Attention.** Measured on ADE20K. Training Memory is the Maximum Memory of a Single GPU. FLOPs of $512 \times 512$ Inputs. ‘IQA’ Denotes Cross-Scale Inter-Query Attention. ‘PIXEL-SA’ Denotes Pixel-Token Self-Attention.

| Backbone | Attention Type | mIoU | Training Mem. | FLOPs |
|----------|----------------|------|---------------|-------|
| Swin-T   | Ours w/o IQA   | 48.6 | 13,762M       | 62.4G |
|          | Ours w/ IQA    | 48.7 | 13,795M       | 63.7G |
|          | Ours w/ Pixel SA | 48.5 | 26,391M       | 127.9G|
| RS50     | Ours w/o IQA   | 44.2 | 18,233M       | 60.1G |
|          | Ours w/ IQA    | 45.5 | 18,264M       | 61.5G |
|          | Ours w/ Pixel SA | 45.3 | 25,816M       | 125.7G|

**Table VIII**  
**Analysis of Query-Pixel Cross-Attention Weight Loss.** $L_{\text{attn}}$ w/o: Attention Weight Loss Without Optimizing Attention Weights From Queries Corresponding to Non-Existing Categories in the Input Images. $L_{\text{attn}'}$: Our Full Attention Weight Loss. Each Row is Three Runs of the Same Experiment.

| $L_{\text{attn}}$ w/o | Stop Iteration | mIoU                  |
|-----------------------|----------------|-----------------------|
| ✓                     | 40k            | 48.2 ± 0.1            |
| ✓                     | 80k            | 48.3 ± 0.4            |
| ✓                     | 120k           | 48.7 ± 0.1            |
| ✓                     | 160k           | 48.7 ± 0.6            |
| ✓                     | ✓              | 48.4 ± 0.7            |
| ✓                     | ✓              | ✓                     |

Fig. 5. Correlation between attention weights and category masks. Left y-axis dashed lines: pearson correlation coefficient between attention weights and category masks. Right y-axis solid lines: mIoU at each training iteration.

Fig. 6. Effects of different $\lambda_{\text{attn}}$ for attention weight loss. The x-axis is drawn on a log scale.
We conduct an ablation to remove this extra mechanism that fuses multi-scale information. Table VIII rows 3, 4, 5, and 6 show the effect of removing the loss at different stages of training. We notice that applying the loss throughout the entire training phase slightly degrades the performance and removing it at 120k iteration shows the best performance and variance (Table VIII rows 3&6). The attention weight loss is used to modulate the optimization of the query-pixel cross-attention layers. Intuitively, if the loss is removed too early during training, such a modulation might show marginal effects on the final performance. On the other hand, applying this loss throughout the entire training phase might constrain too much the cross-attention that it fails to obtain beneficial semantic information outside the category regions. We therefore choose 120k as the number of iterations optimizing the loss for ADE20K dataset and extend it to other datasets’ training settings for our main experiments by matching the proportion of total training iterations (3/4 of the total iterations for each training schedule). In Fig. 5, the slight decrease in correlation between the attention weights and category masks indicates that the queries indeed attend more to the regions outside the category masks after the loss removal. In the supplementary, we provide more visualizations and analysis of the proposed attention weight loss and the effect of the removal of the loss.

Supervision added to the learnable queries before any transformer layer: We conduct an ablation to remove this extra supervision. The performance drops slightly from 48.7 to 48.5 mIoU. We deduce that, even though the learnable queries haven’t attained any information from backbone features before the transformer layers, adding this extra supervision can make them marginally work better with the network parameters, backbone features and prediction targets, since there are linear layers and spatial features involved when producing the predicted probability-mask pairs from the learnable queries (see (4) and (5)).

The choice of multi-scale features: We use multi-scale features of 1/8, 1/16, and 1/32 the input resolution in our segmentation head. In Table IX, we conduct experiments where we gradually add more multi-scale features to the models. Using three scales of features achieves the best trade-off between computation and performance. Reducing the number of multi-scale features noticeably degrades the performance, while adding an extra feature of 1/4 the input resolution brings too much computation overhead with marginal performance improvement.

### V. Advantages and Disadvantages

Our Pyramid Fusion Transformer achieves state-of-the-art performance in several semantic segmentation benchmarks. We summarize the advantages of our proposed method as follows. (1) It effectively and efficiently draws benefits from using multi-scale features and achieves uniform performance gains compared to previous methods. (2) We propose a cross-scale inter-query attention mechanism that fuses multi-scale information with small computational overhead. (3) The proposed cross-attention weight loss further improves the performance without adding any computation in the inference phase. As a result, our method can infer more fine-grained and accurate details (see Fig. 1). For numerical comparisons, our PFT with a Swin-B backbone achieves 55.7 mIoU, compared with MaskFormer using Swin-L backbone obtaining 55.6 mIoU. In this setting, PFT uses 123M parameters and 206 GFLOPs computation, while MaskFormer uses 212M parameters and 375 GFLOPs computation. More examples can be drawn from Tables. II, III, and IV.

However, there is still some room for further improvement. Our ablation studies show that using an additional multi-scale feature with 1/4 the image input resolution can only marginally improve the performance, while adding much more computational costs (see Table IX). Designing a more effective and efficient mechanism to mine for useful information from larger resolution feature might be a future direction to further improve the performance.

### VI. Conclusion

We have presented Pyramid Fusion Transformer that aims to improve the segmentation performance in the per-mask classification semantic segmentation paradigm with multi-scale feature inputs. Using a transformer-based decoder with a novel cross-scale inter-query attention and optimization for query-pixel cross-attention, PFT shows steady improvements over MaskFormer on various datasets and achieves state-of-the-art results. We hope that our approach will inspire the community to further the research in improving per-mask classification segmentation framework.

### References

[1] B. Cheng, A. G. Schwing, and A. Kirillov, “Per-pixel classification is not all you need for semantic segmentation,” in Proc. Adv. Neural Inf. Process. Syst., 2021, pp. 17864–17875.

[2] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., 2015, pp. 3431–3440.

[3] J. Liu, J. He, J. Zhang, J. S. Ren, and H. Li, “Efficient: Holistically-guided decoding for semantic segmentation,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 1–17.

[4] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking atrous convolution for semantic image segmentation,” 2017, arXiv:1706.05587.

[5] Z. Gu, L. Niu, H. Zhao, and L. Zhang, “Hard pixel mining for depth privileged semantic segmentation,” IEEE Trans. Multimedia, vol. 23, pp. 3738–3751, 2021.
[58] T. He et al., “Bag of tricks for image classification with convolutional neural networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 558–567.

[59] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in Proc. Int. Conf. Learn. Representations, 2019.

[60] Y. Yuan, X. Chen, and J. Wang, “Object-contextual representations for semantic segmentation,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 173–190.

[61] H. Ding et al., “Interaction via Bi-directional graph of semantic region affinity for scene parsing,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 15848–15858.

[62] H. Bao, L. Dong, S. Piao, and F. Wei, “BEit: BERT pre-training of image transformers,” in Proc. Int. Conf. Learn. Representations, 2022. [Online]. Available: https://openreview.net/forum?id=p-BhZS5z9o4

[63] J. Jain et al., “SeMask: Semantically masked transformers for semantic segmentation,” in Proc. Int. Conf. Comput. Vis. Workshops, 2023, p. 8.

[64] X. Li et al., “Semantic flow for fast and accurate scene parsing,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 775–793.

[65] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, “Masked-attention mask transformer for universal image segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 1290–1299.

Zipeng Qin received the B.A. degree in computer science and applied mathematics from the University of California, Berkeley, Berkeley, CA, USA, in 2020. He is currently working toward the Ph.D. degree with the Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong. His research interests include deep learning and computer vision.

Jianbo Liu received the B.S. degree from Lanzhou University, Lanzhou, China, in 2012, and the M.S. degree from the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Beijing, China, in 2015, and the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 2023. His research interests include computer vision, deep learning, and scene understanding.

Xiaolin Zhang received the Ph.D. degree from the University of Technology Sydney, Sydney, NSW, Australia, in 2021. He was a Visiting Scholar with the University of Illinois Urbana-Champaign, Champaign, IL, USA, in 2018. From 2021 to 2024, he was a Research Scientist with SenseTime and MooreThreads. He is currently a Professor with the Shandong University of Science and Technology, Qingdao, China. His primary research interests include the fields of computer vision and 3D digital human modeling.

Maoqing Tian received the B.S. and M.S. degrees from Beihang University, Beijing, China, in 2017 and 2020 respectively. He is currently a Senior Researcher with SenseTime Research, Hong Kong. His research interests include deep learning and computer vision.

Aojun Zhou received the M.S. degree from the National Laboratory of Pattern Recognition, Institute of Automation Chinese Academy of Science, Beijing, China, in 2019. He is currently working toward the Ph.D. degree with the Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong. His research interests include deep learning and computer vision.

Shuai Yi received the B.Eng. degree in electronic engineering from Tsinghua University, Beijing, China, in 2012, and the Ph.D. degree in electronic engineering from The Chinese University of Hong Kong, Hong Kong, in 2016. His research interests include computer vision and machine learning.

Hongsheng Li received the bachelor’s degree in automation from the East China University of Science and Technology, Shanghai, China, in 2006, and the Doctorate degree in computer science from Lehigh University, Bethlehem, PA, USA, in 2012. He is currently an Associate Professor with the Department of Electronic Engineering, Chinese University of Hong Kong, Hong Kong. From 2013 to 2015, he was an Associate Professor with the School of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu, China. He has authored or coauthored more than 70 papers in premier conferences on computer vision and machine learning, including CVPR, ICCV, ECCV, NeurIPS, ICLR, and AAAI. His research interests include computer vision, machine learning, and medical image analysis. He was the recipient of the 2020 IEEE CAS Society Outstanding Young Author Award. He won the first place in Object Detection from Videos (VID) track of ImageNet challenge 2016 as the team leader and 2015 as a team co-leader. He is an Associate Editor for Neurocomputing and is an Area Chair of NeurIPS 2021.