# PolyphonicFormer: Unified Query Learning for Depth-aware Video Panoptic Segmentation

Haobo Yuan[1⋆], Xiangtai Li[3⋆], Yibo Yang[4⋆⋆], Guangliang Cheng[5], Jing Zhang[6], Yunhai Tong[3], Lefei Zhang[1,2⋆⋆], and Dacheng Tao[4]

[1] Institute of Artificial Intelligence and School of Computer Science,
Wuhan University, Wuhan, 430072, P. R. China
[2] Hubei Luojia Laboratory, Wuhan, 430072, P. R. China
[3] Key Laboratory of Machine Perception, MOE, School of Artificial Intelligence,
Peking University, Beijing, 100871, P. R. China
[4] JD Explore Academy, Beijing, P. R. China
[5] SenseTime Research, Beijing, P. R. China
[6] The University of Sydney, Sydney, Australia
`yuanhaobo@whu.edu.cn, lxtpku@pku.edu.cu`

**Abstract.** The Depth-aware Video Panoptic Segmentation (DVPS) is a new challenging vision problem that aims to predict panoptic segmentation and depth in a video simultaneously. The previous work solves this task by extending the existing panoptic segmentation method with an extra dense depth prediction and instance tracking head. However, the relationship between the depth and panoptic segmentation is not well explored – simply combining existing methods leads to competition and needs carefully weight balancing. In this paper, we present Polyphonic-Former, a vision transformer to unify these sub-tasks under the DVPS task and lead to more robust results. Our principal insight is that the depth can be harmonized with the panoptic segmentation with our proposed new paradigm of predicting instance level depth maps with object queries. Then the relationship between the two tasks via query-based learning is explored. From the experiments, we demonstrate the benefits of our design from both depth estimation and panoptic segmentation aspects. Since each thing query also encodes the instance-wise information, it is natural to perform tracking directly with appearance learning. Our method achieves state-of-the-art results on two DVPS datasets (Semantic KITTI, Cityscapes), and ranks **1st** on the ICCV-2021 BMTT Challenge video + depth track. Code is available here.

**Keywords:** Depth-aware Video Panoptic Segmentation, Scene Understanding, Multi-Task Learning

## 1 Introduction

To enable machines to perceive the 3D world from 2D video clips, Depth-aware Video Panoptic Segmentation (DVPS) [43] is proposed by extending the video

---

⋆ The first two authors contribute equally.
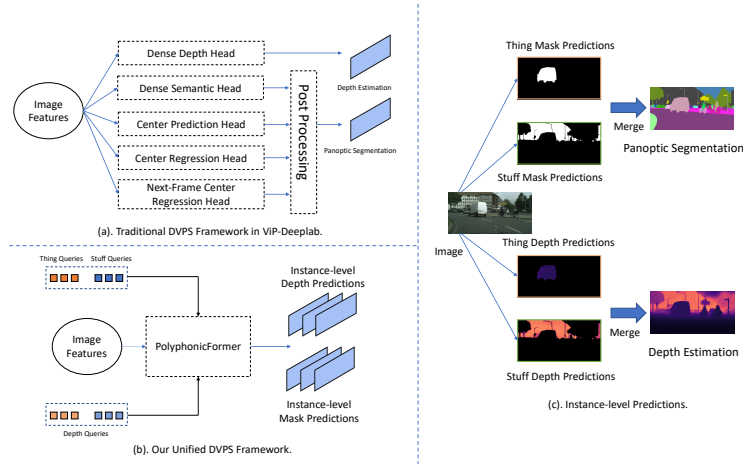⋆⋆ Corresponding Author. Email: zhanglefei@whu.edu.cn, ibo@pku.edu.cn.

Fig. 1: The principal idea of PolyphonicFormer. (a) Previous work's pipeline [43]. (b), Illustration of our unified framework. (c). Key insights of our PolyphonicFormer: The queries are extracted from instance-level information and used to predict instance-level mask and depth results in a unified manner. Best view it on screen.

panoptic segmentation [21,61] with monocular depth estimation [46] which is a challenging and not trivial task for scene understanding. It predicts temporally consistent instance-level semantic results along with per-pixel depth prediction. Achieving accurate and robust DVPS methods in real-world scenarios can greatly promote the development of autonomous driving [15,10].

To tackle this problem, previous work [43] builds two DVPS datasets, including Cityscapes-DVPS and SemKITTI-DVPS by extending two existing datasets Cityscape-VPS [21] and Semantic-KITTI [1]. It also offers a strong baseline ViP-Deeplab [43] by adding two separate task heads based on Panoptic-Deeplab [8] including one dense depth estimation prediction head and one center prediction head for offline tracking as shown in Figure 1(a). However, the separate-head design will damage the performance on both the depth estimation and panoptic segmentation since both fight for suitable backbone features and conflict with each other as observed in previous work [43]. Besides, the model is sensitive to the loss weights of depth estimation and panoptic segmentation, and thus requires a careful balance between the two tasks to enable satisfying performances for both. Therefore, a unified framework that stably facilitates the mutual benefit of the two tasks remains to be developed.

Recently, transformers [11,34] make great progress on vision tasks. Except for the global perception property of vision transformers, the query-based set prediction design in Detection Transformer (DETR) [3] has been proved to be effective in modeling the interactions between instance-level and global contexts. Following the DETR formulation, the query-based design has been applied with good performances on various tasks [13,73,9,53]. In our task, the instance-level semantic contexts have a strong connection with depth information, since the

boundary of an instance in the image corresponds to the region where sharp changes occur in the depth map. So it is natural to utilize query-based reasoning to fulfill the mutual benefit between semantic context and geometry information.

Motivated by such analysis, in this paper, we develop a novel unified query-based representation for things, stuff, and depth, and propose **Polyphonic-Former** (Polyphonic Transformer) where different queries come from different sources (depth or panoptic) but can benefit each other. This is just like the **polyphony** used in the music field. As shown in Figure 1(b), the Polyphonic-Former builds a unified query learning framework to enable the interaction between panoptic context and depth information and make them benefit from each other iteratively. We further link both panoptic and depth queries to facilitate mutual benefit. Our insight is that both depth prediction and panoptic prediction can be linked via the corresponding *instance-wised masks and queries*, which are shown in Figure 1(c), since one unique mask corresponds to one unique query. As a result, the joint modeling can inject semantic information from thing and stuff queries into depth prediction. Moreover, the query-based depth prediction can be seen as regularization items to force semantic consistency during training, which also improves the segmentation results. In terms of temporal modeling, our model directly links to the thing query, which avoids the complex offline post-processing that is used in ViP-Deeplab [43].

Adopting the query-based method can avoid noises and reduce the resource battle from two tasks because the query can be seen as a disentanglement from the feature. This makes our method more robust and less sensitive to the weight of loss for different tasks. Our experiment shows that our unified query learning can enhance both results of panoptic segmentation and depth estimation on a strong baseline. Besides, the experimental results show that the Polyphonic-Former has a robustness of weight choice for the losses in the depth and panoptic predictions. In general, our contributions can be summarized as follows:

- We propose PolyphonicFormer, which unifies the sub-tasks of the depth-aware video panoptic segmentation. To the best of our knowledge, our work is the first unified and end-to-end model in this field. The unified design can leverage information from both the panoptic and depth features to promote the other, and thus makes PolyphonicFormer more stable and insensitive to the loss weights choices that need to be carefully tuned in the previous study.
- Based on the observation that the depth map of each instance has a relatively simple structure than the whole scene, we propose a novel query-based depth estimation scheme to predict the depth map of each instance and merge them into a complete depth map. The query-based depth estimation scheme takes full advantages of the instance boundaries and has better accuracy.
- We conduct extensive experiments on Cityscapes-DVPS, Cityscapes-VPS, and SemKITTI-DVPS to verify the effectiveness of PolyphonicFormer. Our method achieves 64.6 DSTQ on the SemKITTI-DVPS Challenge and won the ICCV-2021 BMTT Challenge video + depth track. Ablation studies show that our framework can boost the *mutual benefit* between depth and panoptic predictions as well as has *robustness on weight choice for losses*.

## 2   Related Work

**Panoptic Segmentation.** The earlier work [23] directly combines predictions of things and stuff from different models. To alleviate computation cost, many works [22,30,7,26,41,70,64] are proposed to model both stuff segmentation and thing segmentation in one model with different task heads. Detection based methods [65,22,42,20,31] usually represent thing within the box prediction, while several bottom-up models [8,69,14,54] perform grouping instance via the pixel-level affinities or instance centers from semantic segmentation results. Recently, several works [3,53,73,9] use transformer-like architecture and use object queries to encode thing and stuff masks in an end-to-end manner. Our method is inspired by these works and takes a further step by jointly modeling depth estimation into panoptic segmentation to build a unified framework.

**Joint Depth and Semantic Modeling.** Several works [6,36,44,45] model the depth estimation and semantic (panoptic) segmentation as a naive multi-task learning procedure and bond them with a series of consistent losses. Some other works [17,27,55] utilize semantic segmentation to guide depth estimation learning. From the perspective of 3D geometry, previous works [24,25,4] use semantic or instance segmentation to tackle the object motion problem left in the self-supervised monocular depth estimation [16]. Recently, ViP-Deeplab [43] extends Panoptic-Deeplab [8] by adding extra dense depth prediction heads, and builds a holistic perception framework. Different from the previous works, where depth and segmentation are tackled separately, we link both parts via object queries in a unified manner, so enjoy the mutual benefit of both tasks.

**Segmentation and Tracking in Video.** Several tasks including Video Object Segmentation (VOS) [40,51,37], Video Instance Segmentation (VIS) [68], Multi-Object Tracking and Segmentation (MOTS) [52], and Video Panoptic Segmentation (VPS) [21] are very close to DVPS. Previous works [21,2,76,63,29] mainly model the temporal consistency for these tasks. VisTR [58] solves VIS problem using transformer [50] in an end-to-end manner. However, this method is limited by the complex scene and long video clip inputs. Recently, STEP [61] proposes a new metric named Segmentation Tracking Quality (STQ) for better evaluating video panoptic segmentation. There are also several tracking methods [47,75,38,60,59] for MOT in terms of transformer architecture and appearance learning. Our PolyphonicFormer simply applies an appearance-based method, Quasi-Dense [38] to extract tracking embeddings as the tracking head.

**Vision Transformer.** Many researchers have adopted Transformer [50] to solve computer vision problems [54,11,49,34,67,72,57,56]. With the help of vision transformers, DETR [3] reformulates the object detection task to output set predictions with an end-to-end framework. The key component is the object query design, which serves as a learnable anchor to detect each instance. This novelty has been also used in panoptic segmentation [53,73,66,9,32,28,74,71] where the pure mask classification can work well on panoptic segmentation. In this paper, inspired by DETR-like segmentation methods [3,9], we borrow the merits of query-based reasoning and into joint depth and panoptic prediction.
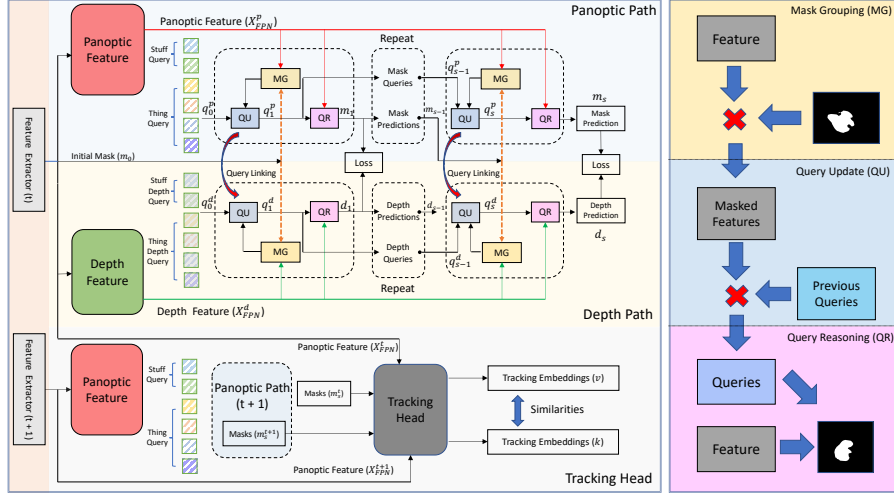
Fig. 2: Illustration of our proposed PolyphonicFormer (left). Our method contains three parts: (1) Feature Extractor to obtain two parallel features for depth estimation and panoptic segmentation. The extracted panoptic (red) and depth (green) features are directly used to predict the final output. (2) Polyphonic Head containing the panoptic path and depth path to refine all the queries at the same time (middle). We use the instance masks to group (orange) the panoptic and depth features for query-based learning. (3) Tracking Head to learn the feature embedding among the frames (bottom). The Mask Grouping (MG), Query Update (QU), and Query Reasoning (QR) (right).

## 3    Method

We explain the architecture design of PolyphonicFormer in detail. As shown in Figure 2, our PolyphonicFormer is composed of three parts: (1) A feature extractor to extract features for each frame. (2) a unified query learning sub-network, which takes three different types of queries and backbone features as inputs and outputs both panoptic segmentation and depth estimation predictions. It contains two paths: depth path and panoptic path, which are interacted with each other iteratively. (3) A lightweight tracking head with several convolution layers to learn the instance-wise tracking embeddings for each thing query.

### 3.1    Feature Extractor

We use a shared backbone network (Residual Network [19] or Swin Transformer [34]) along with Feature Pyramid Network as the feature extractor. We adopt the semantic FPN [22] design to simply fuse the multiscale information since both the depth and panoptic predictions need the high-resolution feature to find the fine structures as well as high-level semantic information. In particular, leveraging a shared backbone, the neck generates different feature maps for panoptic and depth predictions in parallel. These feature maps output from

FPN are denoted as $\boldsymbol{X}^p_{\text{FPN}}$ and $\boldsymbol{X}^d_{\text{FPN}}$, respectively. They are directly used to generate the final and intermediate predictions of depth estimation and panoptic segmentation with the queries in each stage, as described in Section 3.2.

### 3.2   Unified Query Learning

The unified query learning sub-network is composed of two parallel paths for panoptic and depth predictions, respectively. Each path takes the panoptic or depth feature as input and uses *mask grouping* and *query update* to iteratively update the queries. The updated queries are adopted to refine the corresponding mask and depth predictions by the *query reasoning* module.

**Mask Grouping.** At each stage $s$, we first use the mask results generated at the former stage $s - 1$ to group the $\boldsymbol{X}^p_{\text{FPN}}$ and $\boldsymbol{X}^d_{\text{FPN}}$. The key motivation has two folds: The first is that using grouping can avoid noise and select the most salient part for query learning. The second is that adopting grouping or pooling can speed up the learning process of DETR-like models [48,73,29]. Thus, we formulate the process as follows:

$$\boldsymbol{X}^{(n,c)}_s = \sum_u^W \sum_v^H \boldsymbol{m}^n_{s-1}(u,v) \cdot \boldsymbol{X}^c_{\text{FPN}}(u,v), \qquad (1)$$

where $\boldsymbol{m}_{s-1}$ is the mask predictions from former stage. The $\boldsymbol{X}^p_{\text{FPN}}$ is with a $C \times H \times W$ shape, $\boldsymbol{m}$ is with a $N \times H \times W$ shape, if we use $C$ and $N$ to represent number of channels and instances respectively. In PolyphonicFormer, we set $C = 256$ and $N = 100$, respectively. For each mask, we generate a corresponding feature to represent the instance-level information. The instance-level features ($N \times C$) are used to update the queries better focusing on each instance. This process is done in both panoptic path and depth path, with the same former stage mask $\boldsymbol{m}_{s-1}$, as shown in Figure 2 with orange line. With the help of previous stage masks, both the depth and panoptic predictions can be refined with instance-level features.

For the first stage, since the former stage is not available, we use an initial stage (denoted as stage 0), to generate the initial depth and panotpic results. Specifically, we use different convolution module to generate dense predictions on instance segmentation, semantic segmentation, and depth estimation, which are all guided with the ground truth. The initial panoptic segmentation results are delivered to the first stage for mask grouping. The convolution kernels are also transferred to the first stage as the initial queries. Mask Grouping operation is conducted on both panoptic path and depth path separately.

**Query Update.** The masked instance-level features are then delivered to update panoptic and depth queries. We describe the panoptic path first. Inspired by K-Net [73], we adopt an adaptive query update process. The query update process avoids the heavy pixel-wised affinity calculation that is used in DETR-like models [9,3]. The query update process forces the model to learn the instance-aware feature to speed up the training procedure. Specifically, we use a gated

fusion design to capture the information from queries and features. We first calculate the gate features $F_G$ as follows:

$$F_G^p = \phi(\boldsymbol{X}_s^p) \cdot \phi(q_{s-1}^p), \tag{2}$$

where $\phi$ is linear transformations, and all of the $\phi$s in this section are independent. The $F_G$ has the information from both the queries and the instance-level features. With $F_G$, we can calculate gates for features and queries with the following formulation:

$$G_q^p = \sigma(\psi(F_G^p)), G_{\boldsymbol{X}}^p = \sigma(\psi(F_G^p)), \tag{3}$$

where $\psi$s are independent fully connected layers and $\sigma$ are sigmoid functions. With the gates, for the panoptic queries, the updated queries can be calculated as follows:

$$\boldsymbol{q}_s^p = G_{\boldsymbol{X}}^p \cdot \psi(\boldsymbol{X}_s^p) + G_{\boldsymbol{q}}^p \cdot \psi(\boldsymbol{q}_{s-1}^p), \tag{4}$$

where $\psi$s are also independent fully connected layers. The updated queries in this stage take advantages of both the queries from the former stage and the grouped features with the former stage masks.

For the depth path, we adopt a similar process as Equation 2 and Equation 3 to calculate depth features $F_G^d$ and gates $G_q^d$, $G_X^d$. The features and queries from both the panoptic path and depth path will contribute to the current stage depth query:

$$\boldsymbol{q}_s^d = G_{\boldsymbol{X}}^d \cdot \psi(\boldsymbol{X}_s^d) + G_{\boldsymbol{q}}^d \cdot \psi(\boldsymbol{q}_{s-1}^d) + \boldsymbol{q}_s^p. \tag{5}$$

We call this process Query Linking. The query linking process will make the depth query get the instance-level information from both the panoptic path and the depth path, and further bolster the mutual benefit between depth and panoptic predictions.

**Query Reasoning.** With the updated queries, we perform QR to predict the depth and mask results. We simply adopt Multi-Head Self Attention [50] (MSA), Feed-Forward Neural Network [50] (FFN), followed by FC-LN-ReLU layers on queries, which has been proved to be effective in [73]. The depth and mask predictions are then calculated by:

$$\begin{aligned}
\boldsymbol{m}_s &= FFN\left(MSA\left(q_s^m\right)\right) \cdot \boldsymbol{X}_{\mathsf{FPN}}^m, \\
\boldsymbol{d}_s &= FFN\left(MSA\left(q_s^d\right)\right) \cdot \boldsymbol{X}_{\mathsf{FPN}}^d.
\end{aligned} \tag{6}$$

The $\boldsymbol{m}_s$ and $\boldsymbol{d}_s$ are instance-level predictions of stage $s$. Note that both query updating and reasoning perform individually (Panoptic Path and Depth Path in Figure 2). The effect of each component can be found in the experiment part.

In our framework, the queries are refined in each stage to be applied to the features from FPN for panoptic and depth paths to get the predictions. The predictions are conducted with the bipartite match for calculating loss during training and merged into dense predictions when inference. All predictions contribute to the final loss, but only the final stage predictions are used when inference. Please refer to Section 3.5 for details about merging the instance-level depth prediction $\boldsymbol{d}_s$ into the final dense depth map.

### 3.3   Tracking with Thing Query

For the tracking part, we adopt previous work design [68,38] and add a tracking head, where we directly learn appearance embeddings among the different frames. During training, we first match predicted thing masks from the panoptic path to ground truth thing mask according to mask IoU. Then, we use the bound box generated by the ground truth thing mask to conduct ROIAlign [18] for extracting the tracking embeddings. We denote tracking embeddings for each instance from the source frame as $v$, and positive and negative tracking embeddings as $k^+$ and $k^-$. The tracking embedding loss can be calculated as:

$$\mathcal{L}_{\text{track}} = \log[1 + \sum_{k^+} \sum_{k^-} \exp(v \cdot k^- - v \cdot k^+)]. \tag{7}$$

During the inference, we use the thing masks, which are generated from the thing queries to obtain final tracking embeddings for tracking, as shown in the bottom part of Fig. 2. The bi-directional softmax is calculated to associate the instances between two frames (annotated as $\mathbf{n}$ and $\mathbf{m}$):

$$\mathbf{f}(i,j) = [\frac{\exp(\mathbf{n}_i \cdot \mathbf{m}_j)}{\sum_{k=0}^{M-1} \exp(\mathbf{n}_i \cdot \mathbf{m}_k)} + \frac{\exp(\mathbf{n}_i \cdot \mathbf{m}_j)}{\sum_{k=0}^{N-1} \exp(\mathbf{n}_k \cdot \mathbf{m}_j)}]/2. \tag{8}$$

### 3.4   Training

To train the proposed PolyphonicFormer, we need to assign ground truth according to cost since all the outputs are encoded via queries. We also assign ground truth for the initial-stage instance segmentation. In particular, we mainly follow the design of MaskFormer and Max-Deeplab [9,53] to use bipartite matching via a cost considering both segmentation mask and classification results of them. For depth prediction, we use the same bipartite matching results to assign depth ground truth for each query.

After the bipartite matching, we apply a loss jointly considering depth estimation, mask prediction, and classification for each thing or stuff. For depth loss, following ViP-Deeplab [43], we use a joint loss including scale-invariant depth loss [12], absolute relative loss, and square relative loss. We did not find an advantage in our framework with the monodepth2 [16] activation strategy, so we use a sigmoid activation function and multiply the max distance for simplicity. For panoptic segmentation, we apply focal loss [33] on classification and adopt dice loss, focal loss, and cross-entropy loss on mask predictions. The loss for each stage can be formulated as follows:

$$\mathcal{L}_i = \lambda_{depth} \cdot \mathcal{L}_{depth} + \lambda_{mask} \cdot \mathcal{L}_{mask} + \lambda_{cls} \cdot \mathcal{L}_{cls}. \tag{9}$$

Note that the losses are applied to each stage, i.e.,

$$\mathcal{L}_{final} = \sum_{i}^{N} (\lambda_i \cdot \mathcal{L}_i) + \lambda_{track} \cdot \mathcal{L}_{track}, \tag{10}$$

where $N$ is the total stages applied to the framework. We adopt $N = 3$ in PolyphonicFormer, and we set $\lambda_i = 1$ for all stages. The $\lambda_{track}$ is set to 0.25.

### 3.5   Inference

We directly get the instance-level panoptic and depth predictions from the corresponding queries. For final panoptic segmentation results, we adopt the method used in Panoptic-FPN [22] to merge panoptic masks. Since the depth queries and panoptic queries have a one-to-one correspondence, the panoptic segmentation results are used to merge the instance-level depth predictions, i.e.:

$$\boldsymbol{D}(u,v) = \boldsymbol{d}(\boldsymbol{I}(u,v), u, v), \boldsymbol{D} \in \boldsymbol{R}^{H \times W}, \tag{11}$$

where $\boldsymbol{d}$ is the instance-level depth predictions with a $N \times H \times W$ shape from the last stage, $\boldsymbol{I}$ is the instance ID for each pixel, which is from the panoptic segmentation results. The dense depth prediction $\boldsymbol{D}$ ($H \times W$) is then calculated.

For tracking, after getting the embeddings of each instance, we calculate the similarities between the embeddings stored before with the bidirectional softmax similarity. If we detect a new instance embedding that does not appear before, we will store the embedding for future matching. We note that this process only needs *single image input* for each sample during the inference process.

## 4   Experiment

**Experimental Setup.** We implement our models with Pytorch [39], and follow the same training settings from Panoptic Deeplab [8] and ViP-Deeplab [43] where we first pretrain our model on both Mapillary [35] and Cityscapes [10] datasets for Panoptic Segmentation, and then we fine-tune the model with our depth query on Cityscapes-DVPS and SemKITTI-DVPS. During pretraining, we randomly resize the origin images with the scale from 1.0 to 2.0, and then we perform random crops with the $2048 \times 1024$ for Cityscapes-DVPS and $1280 \times 384$ for SemKITTI-DVPS. The batch size is set to 16, and we adopt the synchronized batch normalization during the training. For video training settings, we randomly sample one nearly frame to learn the tracking embeddings. All the models use the *single scale inference*. We adopt Swin-B and Resnet-50 backbones, and we add RFP [42] following ViP-Deeplab [43] but for Swin-B backbone only. We carry out ablation studies on Cityscapes-DVPS datasets for evaluating panoptic segmentation results and depth results.

**Evaluation Metrics.** We use DVPQ metric following ViP-Deeplab [43]. Let $P$ and $Q$ be the prediction and ground truth respectively, k be the window size, and $\lambda$ be the depth threshold. We use $P^c$, $P^{id}$, and $P^d$ to represent semantic segmentation, the instance segmentation, and depth estimation results respectively; $Q^c$, $Q^{id}$, and $Q^d$ are alike but for ground truth. The $DVPQ_\lambda^k$ metric can be formulated as: $\mathrm{PQ}\Big(\Big[\ \|_{i=t}^{t+k-1}\big(\hat{P}_i^c, P_i^{id}\big), \|_{i=t}^{t+k-1}\big(Q_i^c, Q_i^{id}\big)\Big]_{t=1}^{T-k+1}\Big)$. where $\hat{P}_i^c := P_i^c$ for pixels that have an absolute relative depth error inside the error threshold ($\lambda$) and $\hat{P}_i^c := void$ otherwise. $\|_{i=t}^{t+k-1}(P_i^c, P_i^{id})$ denotes the horizontal concatenation of the pair $(P_i^c, P_i^{id})$ from $t$ to $t+k-1$.

| $DVPQ^k_\lambda$ on Cityscapes-DVPS | k = 1 | | | k = 2 | | | k = 3 | | | k = 4 | | | Average | | | FLOPs |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PolyphonicFormer $\lambda = 0.50$ | 70.6 | 63.0 | 76.0 | 62.9 | 49.2 | 72.9 | 59.3 | 42.3 | 71.7 | 56.5 | 36.9 | 70.8 | 62.3 | 47.9 | 72.9 | - |
| PolyphonicFormer $\lambda = 0.25$ | 67.8 | 61.0 | 72.8 | 60.4 | 47.6 | 69.8 | 56.9 | 40.8 | 68.6 | 54.3 | 35.8 | 67.8 | 59.9 | 46.3 | 69.8 | - |
| PolyphonicFormer $\lambda = 0.10$ | 50.2 | 43.4 | 55.2 | 44.4 | 33.4 | 52.4 | 41.5 | 28.6 | 51.0 | 39.5 | 24.7 | 50.4 | 43.9 | 32.5 | 52.3 | - |
| Average: PolyphonicFormer | 62.9 | 55.8 | 68.0 | 55.9 | 43.4 | 65.0 | 52.6 | 37.2 | 63.8 | 50.1 | 32.5 | 63.0 | **55.4** | 42.2 | 65.0 | 1,675G |
| Average: ViP-Deeplab [43] | 61.9 | 55.9 | 66.3 | 55.6 | 44.3 | 63.8 | 52.4 | 38.4 | 62.6 | 50.4 | 34.6 | 61.9 | 55.1 | 43.3 | 63.6 | 9,451G |

| $DVPQ^k_\lambda$ on SemKITTI-DVPS | k = 1 | | | k = 5 | | | k = 10 | | | k = 20 | | | Average | | | FLOPs |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PolyphonicFormer $\lambda = 0.50$ | 58.5 | 55.1 | 61.0 | 52.0 | 42.3 | 59.1 | 50.6 | 39.9 | 58.5 | 49.9 | 38.6 | 58.0 | 52.8 | 44.0 | 59.2 | - |
| PolyphonicFormer $\lambda = 0.25$ | 56.3 | 54.0 | 57.9 | 49.7 | 41.1 | 56.0 | 48.4 | 38.7 | 55.5 | 47.7 | 37.6 | 55.0 | 50.5 | 42.9 | 56.1 | - |
| PolyphonicFormer $\lambda = 0.10$ | 41.8 | 41.1 | 42.4 | 35.1 | 28.2 | 40.1 | 33.7 | 26.0 | 39.3 | 33.0 | 25.1 | 38.7 | 35.9 | 30.1 | 40.1 | - |
| Average: PolyphonicFormer | 52.2 | 50.1 | 53.8 | 45.6 | 37.2 | 51.7 | 44.2 | 34.9 | 51.1 | 43.4 | 33.8 | 50.6 | **46.4** | 39.0 | 51.8 | 402G |
| Average: ViP-Deeplab [43] | 48.9 | 42.0 | 53.9 | 45.8 | 36.9 | 52.3 | 44.4 | 34.6 | 51.6 | 43.4 | 33.0 | 51.1 | 45.6 | 36.6 | 52.2 | 2,267G |

Table 1: Experiment results on Cityscapes-DVPS and SemKITTI-DVPS. Each cell shows $DVPQ^k_\lambda$ | $DVPQ^k_\lambda$-Thing | $DVPQ^k_\lambda$-Stuff where $\lambda$ is the threshold of relative depth error, and $k$ is the number of frames. Smaller $\lambda$ and larger $k$ correspond to a higher accuracy requirement. PolyphonicFormer adopts Swin-B backbone in this table.

However, the $DVPQ^k_\lambda$ metric does not measure the depth quality and tracking quality explicitly; instead, they are observed by the $DVPQ^k_\lambda$ dropping when $k$ and $\lambda$ are getting larger. Therefore, we also report $DSTQ$, which is an extension of $STQ$ [61]. The definition of $DSTQ$ is: $DSTQ = (AQ \times SQ \times DQ)^{\frac{1}{3}}$, where the $AQ$ and $SQ$ are association quality (tracking quality) and segmentation quality (mIoU) respectively [61]. $DQ$ is depth prediction quality. In the ablation study, we report widely used PQ [23] and absolute relative error (abs rel) for panoptic segmentation and depth map quality assessment.

### 4.1   Experimental Results

**Cityscapes-(D)VPS.** The Cityscapes-DVPS dataset [43] is an extension of the Cityscapes-VPS [21] dataset, which contains extra depth annotations. Different from VPSNet [21], we only use the keyframes for training and inference, which means our setting needs fewer inputs and execution rounds to get the results compared to VPSNet [21], and thus might be more challenging. The results on the Cityscapes-VPS dataset are shown in Table 2. In the Cityscapes-VPS dataset, with the very same ResNet-50 backbone, the proposed PolyphonicFormer outperforms the existing methods, VPSNet [21] and SiamTrack [63] drastically, although we successfully get rid of the optical flow input. Note that we did not fine-tune our methods on Cityscapes-VPS after training on Cityscapes-DVPS since depth estimation in our framework will not damage the performance of panoptic segmentation performance in our framework. As shown in Table 1, in the Cityscapes-DVPS dataset, our methods achieve 55.4 average DVPQ with a Swin-B [34] backbone. The DVPQ results demonstrate that the proposed PolyphonicFormer has a good performance on the DVPS task and its sub-tasks.

**SemKITTI-DVPS.** The SemKITTI-DVPS dataset has 19,130 / 4,071 / 4,342 images for training, evaluation, and testing respectively. SemKITTI-DVPS is a challenging dataset since the annotation of both the depth and panoptic segmentation are sparse. As the sparse nature of annotations of SemKITTI-DVPS, following ViP-Deeplab [43], we consider the unannotated pixels as unknown

| Method | k = 1 | k = 2 | k = 3 | k = 4 | VPQ |
|---|---|---|---|---|---|
| VPSNet [21] | 65.0 | 57.6 | 54.4 | 52.8 | 57.5 |
| SiamTrack [63] | 64.6 | 57.6 | 54.2 | 52.7 | 57.3 |
| ViP-Deeplab [43] | 69.2 | 62.3 | 59.2 | 57.0 | 61.9 |
| Ours (ResNet50) | 65.4 | 58.6 | 55.4 | 53.3 | 58.2 |
| Ours (Swin-b) | 70.8 | 63.1 | 59.5 | 56.8 | 62.3 |

| Method | $DSTQ$ |
|---|---|
| rl-lab | 54.8 |
| yang26 | 55.6 |
| Vip-Deeplab | 63.3 |
| PolyphonicFormer | 63.6 |
| PolyphonicFormer* | 64.6 |

Table 2: (Left) Experiment results on Cityscapes-VPS validation set. $k = \{0, 5, 10, 15\}$ in [21] correspond to $k = \{1, 2, 3, 4\}$ in this paper as we use different notations. All the methods in this table is *without* test-time augmentation. (Right) Results on ICCV-2021 SemKITTI-DVPS challenge. * indicates post-challenge submission.

when performing the evaluation. The Table 1 reports our DVPQ results on SemKITTI-DVPS validation set. The PolyphonicFomer achieves a 46.4 average DVPQ. On the test set (ICCV-2021 SemKITTI-DVPS challenge), our method won the championship with 63.6 DSTQ, as shown in Table 2. Beyond that, we re-train the proposed PolyphonicFormer with the additional validation set and achieve 64.6 DSTQ, which is the state-of-the-art result on the test set.

**Comparison on DVPQ.** As in Table 1, we compare our proposed Polyphonic-Former (with Swin-B backbone) with previous work ViP-Deeplab [43], and Poly-phonicFormer outperforms ViP-Deeplab on both datasets. On Cityscapes-DVPS dataset, the proposed PolyphonicFormer / ViP-Deeplab has ∼1675 GFLOPs / ∼9451 GFLOPs, respectively. And on SemKITTI-DVPS dataset, Polyphonic-Former / ViP-Deeplab has ∼402 GFLOPs / ∼2267 GFLOPs, respectively. Note that we do not do the experiments on the much larger WR-41 backbone [5], which is the backbone used in ViP-Deeplab [43], due to the limitation of com-putational resources. We also do not apply *test-time augmentation* and *semi-supervised learning* (both proved to be effective in [43]) for simplicity.

## 4.2    Ablation Studies and Analysis

**(a).Effect of Unified Framework.** We evaluate the effect of the unified frame-work, and the results can be found in Table 3a. In Table 3a, **Depth** refers to the dense prediction baseline of depth estimation. **Panoptic** refers to the panoptic segmentation baseline via query learning, i.e., panoptic path only design in Fig-ure 2. **Hybrid** means that PolyphonicFormer removes the instance-level depth prediction scheme (**Ins** in the table) but only contains the initial dense predic-tion head. As shown in the table, although the single-task baselines are strong for multi-task methods, our method still improves beyond two single-task baselines on both depth and panoptic segmentation via query learning. The performance enhancement on depth estimation from **Hybrid** to **PolyphonicFormer** indi-cates the effectiveness of instance-level depth estimation scheme. We also notice that the performance on panoptic segmentation increased from **Panoptic** to **Hybrid**, and we think it may be because depth supervision at least does not affect and may have some benefits on segmentation in our framework.

| Method | Depth | Panoptic | Ins | PQ ↑ | abs rel ↓ |
|---|---|---|---|---|---|
| ViP-Deeplab [43] | ✓ | ✓ | - | 60.6 | 0.112 |
| Depth | ✓ | - | - | N/A | 0.084 |
| Panoptic | - | ✓ | - | 63.7 | N/A |
| Hybrid (ours) | ✓ | ✓ | - | 65.1 | 0.089 |
| PolyphonicFormer (ours) | ✓ | ✓ | ✓ | 65.2 | 0.080 |

(a)

| $L_{depth}$ | PQ ↑ | abs rel ↓ |
|---|---|---|
| 0.1 | 65.4 | 0.101 |
| 1.0 | 65.3 | 0.089 |
| 5.0 | 65.2 | 0.080 |
| 10 | 65.4 | 0.079 |

(b)

| Method | PQ ↑ | abs rel ↓ |
|---|---|---|
| PolyphonicFormer | 65.2 | 0.080 |
| w/o Query Linking | 64.8 | 0.088 |
| w/o Dense Init | 63.2 | 0.094 |
| w/o Both | 63.0 | 0.104 |

(c)

| Method | PQ ↑ | abs rel ↓ |
|---|---|---|
| P | 65.2 | 0.080 |
| B | 63.8 | 0.087 |
| A | 65.3 | 0.081 |

(d)

| Method | DSTQ ↑ | AQ ↑ |
|---|---|---|
| PolyphonicFormer + DeepSort [62] | 51.8 | 25.9 |
| PolyphonicFormer + Unitrack [59] | 49.3 | 22.5 |
| PolyphonicFormer + QuasiDense [38] | 63.6 | 46.2 |

(e)

| Stages | PQ ↑ | abs rel ↓ |
|---|---|---|
| 1 | 64.1 | 0.081 |
| 2 | 64.6 | 0.081 |
| 3 | 65.2 | 0.080 |

(f)

Table 3: Ablation studies on PolyphonicFormer. **N/A** indicates **"not applicable"**, and **w/o** indicates **"without"**. "↑" indicates higher is better, and "↓" indicates lower is better. Please refer to each paragraph for the description of the abbreviations and results analysis. All experiments use ResNet-50 backbone if no further description.

**(b).Robustness of PolyphonicFormer.** We modify the depth loss weight to evaluate the robustness of setting different depth loss weights. We set the depth loss weights from 0.1, 1, to 5 and 10. The results are shown in Table 3b. We found that with different depth loss weights, the PQ results do not change drastically. The phenomenon is different from ViP-Deeplab [43] where the depth estimation and panoptic segmentation are with two separate heads and sensitive to the loss choices. We speculate that it is the unified framework that makes two different tasks "polyphonic" instead of battling with each other.

**(c).Ablation Study on Panoptic Depth Interaction.** We perform ablation study on panoptic depth interaction. As shown in Table 3c, we do experiments on the original settings along with dismissing the "Query Linking" and "Dense Init" (Dense Depth Initialization). "Dense Depth Initialization" is a strategy to initialize the depth queries. We empirically found that instead of initializing the depth queries randomly, initializing it with the queries at the initial stage of depth path is with a better performance. We also found that the "Query Linking" module is a simple but effective module to boost the mutual benefit.

**(d).Effect of Query Linking Module Design.** We have tried different forms of query linking module design. In Table 3d, "**P**" refers to the current PolyphonicFormer design, which simply uses an addition from the panoptic queries to the depth queries. We try to replace the current design to "**B**", a bidirectional query linking module that links queries not only from panoptic to depth but also from depth to panoptic. We found that the information from depth queries may not be that helpful to panoptic queries. We also try to apply a more advanced query linking module, for instance, a self-attention module to fuse panoptic queries and depth queries ("**A**" in the table). The results may indicate that the query linking itself is more important than the linking module design choice. So, we would like to keep it simple and use the current query linking design, an **ADD** module, to propagate the semantic context to the geometry context.
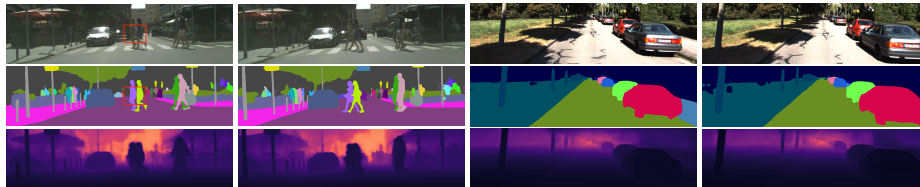
Fig. 3: Visualizations of PolyphonicFormer on Cityscapes-DVPS and SemKITTI-DVPS. From top to down: input image, temporally consistent panoptic segmentation prediction, and monocular depth prediction. The red box is a failure case example.

**(e).Ablation Study on Tracking Head Design.** The DVPQ [43] does not have an explicit tracking quality metric, so we apply Association Quality (AQ) in DSTQ metric [61] for tracking head evaluation. To alleviate the deviation of tracking quality caused by the mask prediction's inaccuracy, we choose to use Swin-B backbone [34] here. As demonstrated in Table 3e, we explore three different tracking methods including DeepSort [62], Unitrack [59] and Quansi-Dense [38]. We found the QuansiDense tracker works best and is chosen as the final tracking head. The QuansiDense tracker is an online tracker and only uses appearance embeddings. For the fair comparison with ViP-Deeplab [43], we do not explore the geometry information for tracking in the current framework.

**(f).Ablation Study on Iterative Rounds.** The PolyphonicFormer performs in an iterative manner. We carry out the ablation study on the choice of iterative stages. We perform studies from 1 to 3 stages to study the effectiveness of iterative design. The results can be found in Table 3f. With more iterative rounds, as we expected, the instance-level information from the features could be more used to refine the query to predict more accurate mask and depth results. We have also tried to use more iterative rounds and do not found salient performance enhancement, so we choose to set the number of stages to 3 following [73].

### 4.3   Visualization and Limitation Analysis

**Visualization and Analysis.** We do visualization analysis on the two DVPS datasets. As shown in Figure 3, the PolyphonicFormer can predict frame-consistent panoptic segmentation along with depth in a unified manner. In the panoptic segmentation map, things with the same color are a single identity over the different frames. Figure 4 shows that the depth estimation task could benefit from the polyphonic design. We take the patch cropped from the depth map (bottom) as an example. The dense prediction neural network has an uncertainty on the boundary and tend to predict vague results. Instead, our PolyphonicFormer can clearly distinguish the boundary of the car and thus predicts a depth map with a clearer boundary. Then we measure the depth-aware capability of the PolyphonicFormer and compare it with the traditional dense depth estimation. Specifically, we verify the depth prediction accuracy of each instance. We consider each instance as **depth awareness** when less than a specific proportion

Fig. 4: The depth estimation results of the traditional dense depth estimation (middle) and PolyphonicFormer (right). For the small objects, the proposed unified framework could successfully distinguish the boundary for enhancing depth results.

(10%) of pixels with absolute relative errors in depth estimation exceeds a certain threshold (25%). The thresholds here follow the design of DQ. Simply combination of depth estimation and panoptic segmentation (similar structure as in [43]) leads to 5,851 depth-aware instances out of 7,998 (73.2%) instances in total, while the PolyphonicFormer achieves 6,119 out of 7,998 (76.5%) results on the Cityscapes-DVPS validation set. The depth-aware capability of PolyphonicFormer is consistent with the visualization results and demonstrates the effectiveness of the unified framework for the DVPS task.

**Limitation and Future Work.** As shown in Figure 3, the tracking head may fail when handling extreme occlusion cases (e.g. the car in the red box). This is because PolyphonicFormer mainly performs tracking with appearance embeddings like the previous work. When applying PolyphonicFormer directly to safety-critical tasks like autonomous driving, the failure cases may cause accidents. The reliability needs further consideration and is potentially future work.

## 5 Conclusion

In this paper, we explore the relationship between semantics and geometry and build PolyphonicFormer for joint modeling panoptic segmentation and depth estimation. Our key insight is to unify all scene queries (thing, stuff) and depth queries into one framework. Based on our experiments, the PolyphonicFormer enhances the robustness towards panoptic segmentation as well as improves the performance of depth estimation. The proposed PolyphonicFormer achieves state-of-the-art results on both Cityscapes-DVPS and SemKITTI-DVPS datasets, and outperforms other methods on the SemKITTI-DVPS challenge. We hope PolyphonicFormer could serve as a good baseline in the DVPS task.

# References

1. Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J.: SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In: ICCV (2019)
2. Bertasius, G., Torresani, L.: Classifying, segmenting, and tracking object instances in video with mask propagation. In: CVPR (2020)
3. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV (2020)
4. Casser, V., Pirk, S., Mahjourian, R., Angelova, A.: Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In: AAAI (2019)
5. Chen, L.C., Lopes, R.G., Cheng, B., Collins, M.D., Cubuk, E.D., Zoph, B., Adam, H., Shlens, J.: Naive-student: Leveraging semi-supervised learning in video sequences for urban scene segmentation. In: ECCV (2020)
6. Chen, P.Y., Liu, A.H., Liu, Y.C., Wang, Y.C.F.: Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation. In: CVPR (2019)
7. Chen, Y., Lin, G., Li, S., Bourahla, O., Wu, Y., Wang, F., Feng, J., Xu, M., Li, X.: Banet: Bidirectional aggregation network with occlusion handling for panoptic segmentation. In: CVPR (2020)
8. Cheng, B., Collins, M.D., Zhu, Y., Liu, T., Huang, T.S., Adam, H., Chen, L.C.: Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In: CVPR (2020)
9. Cheng, B., Schwing, A.G., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation. In: NeurIPS (2021)
10. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR (2016)
11. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2020)
12. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. NIPS (2014)
13. Fang, Y., Yang, S., Wang, X., Li, Y., Fang, C., Shan, Y., Feng, B., Liu, W.: Instances as queries. In: ICCV (2021)
14. Gao, N., Shan, Y., Wang, Y., Zhao, X., Yu, Y., Yang, M., Huang, K.: Ssap: Single-shot instance segmentation with affinity pyramid. In: ICCV (2019)
15. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: CVPR (2012)
16. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth estimation. In: ICCV (2019)
17. Guizilini, V., Hou, R., Li, J., Ambrus, R., Gaidon, A.: Semantically-guided representation learning for self-supervised monocular depth. In: ICLR (2019)
18. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV (2017)
19. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
20. Hou, R., Li, J., Bhargava, A., Raventos, A., Guizilini, V., Fang, C., Lynch, J., Gaidon, A.: Real-time panoptic segmentation from dense detections. In: CVPR (2020)

21. Kim, D., Woo, S., Lee, J.Y., Kweon, I.S.: Video panoptic segmentation. In: CVPR (2020)
22. Kirillov, A., Girshick, R., He, K., Dollár, P.: Panoptic feature pyramid networks. In: CVPR (2019)
23. Kirillov, A., He, K., Girshick, R., Rother, C., Dollár, P.: Panoptic segmentation. In: CVPR (2019)
24. Klingner, M., Termöhlen, J.A., Mikolajczyk, J., Fingscheidt, T.: Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance. In: ECCV (2020)
25. Lee, S., Im, S., Lin, S., Kweon, I.S.: Learning monocular depth in dynamic scenes via instance-aware projection consistency. In: AAAI (2021)
26. Li, J., Raventos, A., Bhargava, A., Tagawa, T., Gaidon, A.: Learning to fuse things and stuff. arXiv:1812.01192 (2018)
27. Li, R., Mao, Q., Wang, P., He, X., Zhu, Y., Sun, J., Zhang, Y.: Semantic-guided representation enhancement for self-supervised monocular trained depth estimation. arXiv preprint arXiv:2012.08048 (2020)
28. Li, X., Xu, S., Yang, Y., Cheng, G., Tong, Y., Tao, D.: Panoptic-partformer: Learning a unified model for panoptic part segmentation. In: ECCV (2022)
29. Li, X., Zhang, W., Pang, J., Chen, K., Cheng, G., Tong, Y., Loy, C.C.: Video k-net: A simple, strong, and unified baseline for video segmentation. In: CVPR (2022)
30. Li, Y., Chen, X., Zhu, Z., Xie, L., Huang, G., Du, D., Wang, X.: Attention-guided unified network for panoptic segmentation. In: CVPR (2019)
31. Li, Y., Zhao, H., Qi, X., Wang, L., Li, Z., Sun, J., Jia, J.: Fully convolutional networks for panoptic segmentation. CVPR (2021)
32. Li, Z., Wang, W., Xie, E., Yu, Z., Anandkumar, A., Alvarez, J.M., Lu, T., Luo, P.: Panoptic segformer. arXiv preprint arXiv:2109.03814 (2021)
33. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV (2017)
34. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV (2021)
35. Neuhold, G., Ollmann, T., Rota Bulo, S., Kontschieder, P.: The mapillary vistas dataset for semantic understanding of street scenes. In: ICCV (2017)
36. Ochs, M., Kretz, A., Mester, R.: Sdnet: Semantically guided depth estimation network. arXiv preprint arXiv:1907.10659 (2019)
37. Oh, S.W., Lee, J.Y., Xu, N., Kim, S.J.: Video object segmentation using space-time memory networks. In: ICCV (2019)
38. Pang, J., Qiu, L., Li, X., Chen, H., Li, Q., Darrell, T., Yu, F.: Quasi-dense similarity learning for multiple object tracking. In: CVPR (2021)
39. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. In: NeurIPS (2019)
40. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: CVPR (2016)
41. Porzi, L., Bulo, S.R., Colovic, A., Kontschieder, P.: Seamless scene segmentation. In: CVPR (2019)
42. Qiao, S., Chen, L.C., Yuille, A.: Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In: CVPR (2021)
43. Qiao, S., Zhu, Y., Adam, H., Yuille, A., Chen, L.C.: Vip-deeplab: Learning visual perception with depth-aware video panoptic segmentation. CVPR (2021)

44. Ramirez, P.Z., Poggi, M., Tosi, F., Mattoccia, S., Di Stefano, L.: Geometry meets semantics for semi-supervised monocular depth estimation. In: ACCV (2018)
45. Saeedan, F., Roth, S.: Boosting monocular depth with panoptic segmentation maps. In: WACV (2021)
46. Saxena, A., Chung, S.H., Ng, A.Y., et al.: Learning depth from single monocular images. In: NIPS (2005)
47. Sun, P., Cao, J., Jiang, Y., Zhang, R., Xie, E., Yuan, Z., Wang, C., Luo, P.: Transtrack: Multiple-object tracking with transformer. arXiv preprint arXiv: 2012.15460 (2020)
48. Sun, P., Zhang, R., Jiang, Y., Kong, T., Xu, C., Zhan, W., Tomizuka, M., Li, L., Yuan, Z., Wang, C., Luo, P.: SparseR-CNN: End-to-end object detection with learnable proposals. CVPR (2021)
49. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: ICML (2021)
50. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NIPS (2017)
51. Voigtlaender, P., Chai, Y., Schroff, F., Adam, H., Leibe, B., Chen, L.C.: Feelvos: Fast end-to-end embedding learning for video object segmentation. In: CVPR (2019)
52. Voigtlaender, P., Krause, M., Osep, A., Luiten, J., Sekar, B.B.G., Geiger, A., Leibe, B.: Mots: Multi-object tracking and segmentation. In: CVPR (2019)
53. Wang, H., Zhu, Y., Adam, H., Yuille, A., Chen, L.C.: Max-deeplab: End-to-end panoptic segmentation with mask transformers. In: CVPR (2021)
54. Wang, H., Zhu, Y., Green, B., Adam, H., Yuille, A., Chen, L.C.: Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. In: ECCV (2020)
55. Wang, L., Zhang, J., Wang, O., Lin, Z., Lu, H.: Sdc-depth: Semantic divide-and-conquer network for monocular depth estimation. In: CVPR (2020)
56. Wang, W., Cao, Y., Zhang, J., Tao, D.: Fp-detr: Detection transformer advanced by fully pre-training. In: ICLR (2021)
57. Wang, W., Zhang, J., Cao, Y., Shen, Y., Tao, D.: Towards data-efficient detection transformers. In: ECCV (2022)
58. Wang, Y., Xu, Z., Wang, X., Shen, C., Cheng, B., Shen, H., Xia, H.: End-to-end video instance segmentation with transformers. In: CVPR (2021)
59. Wang, Z., Zhao, H., Li, Y.L., Wang, S., Torr, P.H., Bertinetto, L.: Do different tracking tasks require different appearance models? In: NeurIPS (2021)
60. Wang, Z., Zheng, L., Liu, Y., Li, Y., Wang, S.: Towards real-time multi-object tracking. In: ECCV (2020)
61. Weber, M., Xie, J., Collins, M.D., Zhu, Y., Voigtlaender, P., Adam, H., Green, B., Geiger, A., Leibe, B., Cremers, D., et al.: Step: Segmenting and tracking every pixel. In: NeurIPS (2021)
62. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: ICIP (2017)
63. Woo, S., Kim, D., Lee, J.Y., Kweon, I.S.: Learning to associate every segment for video panoptic segmentation. In: CVPR (2021)
64. Wu, Y., Zhang, G., Xu, H., Liang, X., Lin, L.: Auto-panoptic: Cooperative multi-component architecture search for panoptic segmentation. NeurIPS (2020)
65. Xiong, Y., Liao, R., Zhao, H., Hu, R., Bai, M., Yumer, E., Urtasun, R.: Upsnet: A unified panoptic segmentation network. In: CVPR (2019)
66. Xu, S., Li, X., Wang, J., Cheng, G., Tong, Y., Tao, D.: Fashionformer: A simple, effective and unified baseline for human fashion segmentation and recognition. In: ECCV (2022)

67. Xu, Y., Zhang, Q., Zhang, J., Tao, D.: Vitae: Vision transformer advanced by exploring intrinsic inductive bias. In: NeurIPS (2021)
68. Yang, L., Fan, Y., Xu, N.: Video instance segmentation. In: ICCV (2019)
69. Yang, T.J., Collins, M.D., Zhu, Y., Hwang, J.J., Liu, T., Zhang, X., Sze, V., Papandreou, G., Chen, L.C.: Deeperlab: Single-shot image parser. arXiv:1902.05093 (2019)
70. Yang, Y., Li, H., Li, X., Zhao, Q., Wu, J., Lin, Z.: Sognet: Scene overlap graph network for panoptic segmentation. In: AAAI (2020)
71. Zhang, J., Li, X., Wang, Y., Wang, C., Yang, Y., Liu, Y., Tao, D.: Eatformer: Improving vision transformer inspired by evolutionary algorithm. arXiv preprint arXiv:2206.09325 (2022)
72. Zhang, Q., Xu, Y., Zhang, J., Tao, D.: Vitaev2: Vision transformer advanced by exploring inductive bias for image recognition and beyond. arXiv preprint arXiv:2202.10108 (2022)
73. Zhang, W., Pang, J., Chen, K., Loy, C.C.: K-net: Towards unified image segmentation. In: NeurIPS (2021)
74. Zhou, Q., Li, X., He, L., Yang, Y., Cheng, G., Tong, Y., Ma, L., Tao, D.: Transvod: End-to-end video object detection with spatial-temporal transformers. arXiv preprint arXiv:2201.05047 (2022)
75. Zhou, X., Koltun, V., Krähenbühl, P.: Tracking objects as points. In: ECCV (2020)
76. Zhu, X., Xiong, Y., Dai, J., Yuan, L., Wei, Y.: Deep feature flow for video recognition. In: CVPR (2017)