Appendix

A Implementation Details

We report the implementation details in this section. To train the PolyphonicFormer, we first pre-train the backbone on ImageNet-1K [8] and the pre-train the panoptic path with Mapillary [5] and Cityscapes [3] datasets, following the ViP-Deeplab [6]. When performing the pre-training on Mapillary, we resize the original images to a random scale from $2048 \times 1024$ to $4096 \times 2048$ and randomly crop a $1024 \times 1024$ sample. We do the Mapillary pre-training for 300 epochs. For Cityscapes pre-training, we also perform random resize from $2048 \times 1024$ to $4096 \times 2048$, but crop to $2048 \times 1024$. After pre-training, we train the image baseline of PolyphonicFormer on Cityscapes-DVPS. Training on Cityscapes-DVPS requires 192 epochs and takes the same data augmentation strategy as the Cityscapes dataset. The depth ground truth needs to be divided by the resize scale factor because resizing an image means zooming the image for depth perception. The ablation studies are performed on the image baseline. With
Fig. 2: We present several videos here. The left and middle are mask and depth results of tracked instance examples output by PolyphonicFormer. The depth results are merged into the final prediction (bottom right).

the image baseline, we fine-tune the PolyphonicFormer with a tracking head on Cityscapes-DVPS and SemKITTI-DVPS respectively for 48 epochs. For each sample, we randomly choose a reference frame from time $t - 2, t - 1, t + 1, \text{and } t + 2$ for source frame at time $t$. We do not perform random scale resizing on SemKITTI-DVPS, and only pad the images to $1280 \times 384$ instead, which is the minimum size that can be divided by 32 to cover all of the KITTI images. All of the datasets we used are without extra data with pseudo labels [2] for self-supervised or semi-supervised training. During inference, for simplicity, we use single scale inference with the original image size from the datasets, and we do not use the test time online depth refinement [1]. In general, except for that we do not use the test-time augmentation and semi-supervised learning, we adopt similar settings with ViP-Deeplab [6].

For the ICCV-2021 SemKITTI-DVPS Challenge submission, we take advantage of the validation set for training. Before and after adding the extra validation samples, PolyphonicFormer can achieve 63.6 and 64.6 DSTQ respectively in the SemKITTI-DVPS test set.

B More Visualization analyses

We show more visualization analysis results in Figure 1. The depth predictions of PolyphonicFormer are merged from depth predictions for each thing or stuff mask. As shown in Figure 1, the final depth predictions of PolyphonicFormer successfully distinguish the boundary between the instances or instances and corresponding background and thus are more accurate than the dense prediction results. We note that the results are presented with a high resolution, so we recommend the readers zoom in to check the details about the depth results of other instances.

We also illustrate the unified query learning with a video, as shown in Figure 2. The PolyphonicFormer generates temporal-consistent instance-level mask and depth predictions and merges them into the final results.
Appendix

Table 1: Experimental results on Cityscapes-DVPS and SemKITTI-DVPS datasets with Resnet-50 backbone. Each cell shows DVPQ$^k_A \mid$ DVPQ$^k$-Thing $\mid$ DVPQ$^k_A$-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. We also estimate the computational cost (FLOPs) of ViP-Deeplab with Resnet-50 backbone and get 1,096G and 280G on Cityscapes-DVPS and SemKITTI-DVPS respectively.

<table>
<thead>
<tr>
<th>method</th>
<th>abs rel</th>
<th>sq rel</th>
<th>RMSE</th>
<th>RMSE log</th>
<th>$\sigma &lt; 1.25$</th>
<th>$\sigma &lt; 1.25^2$</th>
<th>$\sigma &lt; 1.25^3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPT-Hybrid [7]</td>
<td>0.0697</td>
<td>0.4515</td>
<td>4.115</td>
<td>0.1106</td>
<td>0.9434</td>
<td>0.9914</td>
<td>0.9976</td>
</tr>
<tr>
<td>PolyphonicFormer</td>
<td>0.0647</td>
<td>0.3454</td>
<td>3.800</td>
<td>0.1013</td>
<td>0.9524</td>
<td>0.9950</td>
<td>0.9985</td>
</tr>
</tbody>
</table>

Table 2: Comparison results of PolyphonicFormer and the representative depth estimation method. The metrics with orange background means "lower" is better. The metrics with blue background means "higher" is better. ViP-Deeplab [6] has a 0.0721 abs rel.

C More Experiments

We report more results of PolyphonicFormer in this section. The results of PolyphonicFormer with Swin-B backbone are already provided, and we report the DVPQ results with Resnet-50 backbone in this section. As shown in Table 1, with a Resnet-50 backbone, the PolyphonicFormer achieves 48.1, and 40.3 in DVPQ on the Cityscapes-DVPS and SemKITTI-DVPS datasets, respectively.

We compare PolyphonicFormer with recently proposed DPT [7], which is one of the state-of-the-art supervised monocular depth estimation methods on KITTI (eigen split) [4]. As the KITTI dataset lacks the panoptic segmentation annotation and the SemanticKITTI dataset has a very different split strategy compared with eigen split, we cannot directly get the results on the KITTI eigen split. We adopt the pre-trained model of DPT-Hybrid on MIX6 [7] (meta-datasets containing 10 datasets) and KITTI eigen split, and fine-tune on Cityscapes-DVPS with the same schedule of PolyphonicFormer. As in Table 2, our proposed PolyphonicFormer outperforms the DPT-Hybrid [7] and ViP-Deeplab [6].

D More Visualization Results

We show some of the visualization results from the Cityscapes-DVPS and SemKITTI-DVPS datasets along with PolyphonicFormer (Swin-B backbone) predictions in Figure 3 and Figure 4.
References

Fig. 3: Prediction visualizations on Cityscapes-DVPS. From left to right: input images, temporally consistent panoptic segmentation (TCPS), and depth predictions. Color change of the same instance of TCPS indicates an id switch.
Fig. 4: Prediction visualizations on SemKITTI-DVPS. From left to right: input images, temporally consistent panoptic segmentation (TCPS), and depth predictions. Color change of the same instance of TCPS indicates an id switch.