Inverted Pyramid Multi-task Transformer for Dense Scene Understanding –Supplementary Materials–

Hanrong Ye and Dan Xu✉

Department of Computer Science and Engineering, HKUST
Clear Water Bay, Kowloon, Hong Kong
{hyeae,danxu}@cse.ust.hk

1 More Implementation Details

In this section, we provide more details about our model implementation in addition to those discussed in the paper. The code and trained models for our method are publicly available on https://github.com/prismformore/InvPT.

Model Optimization. For evaluation on NYUD-v2 [6] and PASCAL-Context [2], we totally consider six dense prediction tasks, including semantic segmentation (Semseg), monocular depth estimation (Depth), surface normal estimation (Normal), human parsing (Parsing), saliency detection (Saliency), and object boundary detection (Boundary). For the continuous regression tasks (*i.e.* Depth and Normal) a $\mathcal{L}1$ Loss is employed. For the discrete classification tasks (*i.e.* Semseg, Parsing, Saliency, and Boundary), a cross-entropy loss is utilized. For the sake of simplicity, we use the same set of loss functions for both intermediate and final supervision. The whole model can be end-to-end optimized.

Data Processing. For a fair comparison with ATRC [1], we follow its data processing pipeline. On PASCAL-Context, we pad the image to the size of 512×512, while on NYUD-v2, we randomly crop the input image to the size of 448×576 as Swin Transformer [4] requires both the height and width to be even for conducting patch merging. We use typical data augmentation including random scaling, cropping, horizontal flipping and color jittering.

Implementation Details of Encoder Feature Aggregation (EFA). For Swin Transformer encoders [4], we pass feature sequences from the first three stages to Inverted Pyramid Transformer Decoder (InvPT decoder). For ViT encoders [3], since they do not explicitly define the concept of stage, we evenly choose 3 layers based on the depth and unfold their output spatially, and then use transposed convolution to upsample the resolution of feature maps to match the spatial resolution in the corresponding decoder stage before further transformation. Specifically, for ViT-base encoder, we use the output token sequences of layer 3, 6, and 9, while for ViT-large encoder we use output token sequences of layer 6, 12, and 18. The kernel size and stride of the transposed convolution for the feature at the first scale are 4, and those at the second scale are 2.

Details about Self-attention in InvPT Decoder. The specific shapes of the query, the key, and the value matrices (*i.e.* $\mathbf{Q}, \mathbf{K}, \mathbf{V}$) in different UP-Transformer

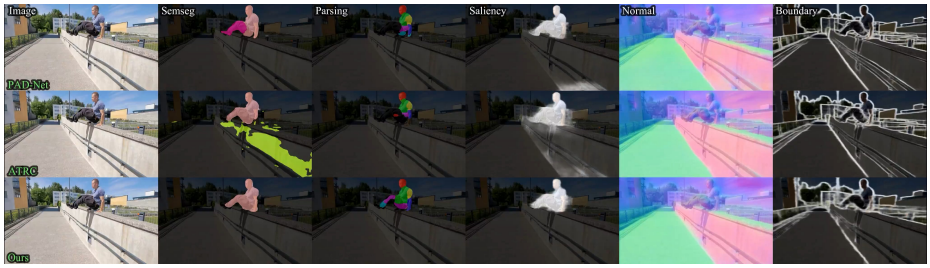

Fig. 1: An example frame of the demo video for the study of generalization performance. Models are all trained on PASCAL-Context and tested on DAVIS video dataset. Our method yields qualitatively better generalization performance compared to PAD-Net [7] and ATRC [1].

Table 1: Shapes of $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ matrices in different upsampling stages. Please refer to Sec. 3.4 in paper for the detailed definitions of the notations in the table.

	$s=0$	$s=1$	$s=2$
$\mathbf{Q}$	$\frac{TH_0W_0}{4}, C_0$	$TH_0W_0, \frac{C_0}{2}$	$4TH_0W_0, \frac{C_0}{4}$
$\mathbf{K}$	$\frac{TH_0W_0}{4}, C_0$	$\frac{TH_0W_0}{4}, \frac{C_0}{2}$	$\frac{TH_0W_0}{4}, \frac{C_0}{4}$
$\mathbf{V}$	$\frac{TH_0W_0}{4}, C_0$	$\frac{TH_0W_0}{4}, \frac{C_0}{2}$	$\frac{TH_0W_0}{4}, \frac{C_0}{4}$

stages are shown in Table 1. Please refer to Sec. 3.4 in paper for the detailed definitions of the notations in the table.

2 More Experimental Results and Analysis

Video Demo for Generalization Performance Comparison on DAVIS Video Dataset. As introduced in the paper, to qualitatively study the generalization ability of the proposed multi-task transformer for dense scene understanding, we compare it with the best performing method, including ATRC [1] and PAD-Net [7], on the challenging DAVIS video Dataset [5]. The results are shown in the github page. All the models are trained on Pascal-Context with 5 tasks, *i.e.* semantic segmentation, surface normal estimation, human parsing, saliency detection, and object boundary detection. Then the models are directly tested on DAVIS to generate multi-task predictions in the demo video. Significantly stronger generalization ability of our InvPT is observed and an example frame is shown in Fig. 1.

Effect of Different Transformer Encoders. Similar to the results on PASCAL-Context in the paper, we compare two families of transformer encoders: Swin Transformer (Swin-T, Swin-B and Swin-L) [4] and ViT (ViT-B and ViT-L) [3] on NYUD-v2 (Table 2). We observe that the bigger model of the same model family consistently brings performance gain on semantic segmentation and monocular

Table 2: Performance comparison of using different transformer encoder structures in InvPT on NYUD-v2.

Model	**Semseg** mIoU ↑	**Depth** RMSE ↓	**Normal** mErr ↓	**Boundary** odsF ↑
Swin-T	44.27	0.5589	20.46	76.10
Swin-B	50.97	0.5071	**19.39**	77.30
Swin-L	**51.76**	**0.5020**	**19.39**	**77.60**
Vit-B	50.30	0.5367	**19.00**	77.60
Vit-L	**53.56**	**0.5183**	19.04	**78.10**

Table 3: Computation analysis of InvPT with different backbones.

InvPT	w/ Swin-T	w/ Swin-B	w/ Swin-L	w/ Vit-B	w/ Vit-L
Runtime (sec/img)	0.0270	0.0375	0.0462	0.0356	0.0633
GPU memory (MB)	2921	3238	4973	3175	4977
Number of parameters	78M	162M	364M	138M	380M

Table 4: Computation efficiency of UP-Transformer Block.

Method	Runtime (sec/img)	GPU memory (MB)	Number of parameters
InvPT w/ vanilla ViT Upsampling	0.1194	11827	400M
InvPT w/ UP-Transformer (**ours**)	**0.0356**	**3175**	**138M**

depth estimation, while on other dense tasks (*i.e.* saliency and boundary detection), it does not necessarily yield significantly better performance despite with higher model capacity. This phenomenon may result from the distinct characteristics of different dense prediction tasks.

Computation cost of InvPT We show the computation cost of the proposed InvPT model in Table 3, including runtime per image, single-sample GPU-memory consumption, and the model size (in terms of number of parameters) of InvPT with different transformer backbone architectures. We run on the test split (*i.e.* 5,105 images in total) of PASCAL-Context dataset and calculate the average inference runtime per sample using a NVIDIA RTX 3090 GPU.

Computation efficiency of UP-Transformer Block Table 4 compares between our UP-Transformer block and a vallina vision transformer-based [3] upsampling block which also upsamples the multi-task outputs with the same three stages. They use the same Vit-B encoder. It is clear that our design achieves significant improvement on efficiency compared to the baseline (*e.g.* approximately 3× more efficient in terms of runtime, GPU memory, and number of parameters).

More Qualitative Results. We show more prediction results by InvPT (ours) and the previous SOTA method ATRC [1] on the challenging PASCAL-Context dataset in Fig. 2 and 3. It is clear that our method produces significantly better results than ATRC, especially on semantic segmentation and human parsing.

Qualitative Comparison of the Preliminary and Final Predictions of InvPT. Fig. 4 shows the qualitative comparison of the preliminary predictions and the final predictions generated by InvPT on PASCAL-Context. We can observe that InvPT decoder successfully refines the preliminary predictions and generates remarkably better results on all these dense prediction tasks.

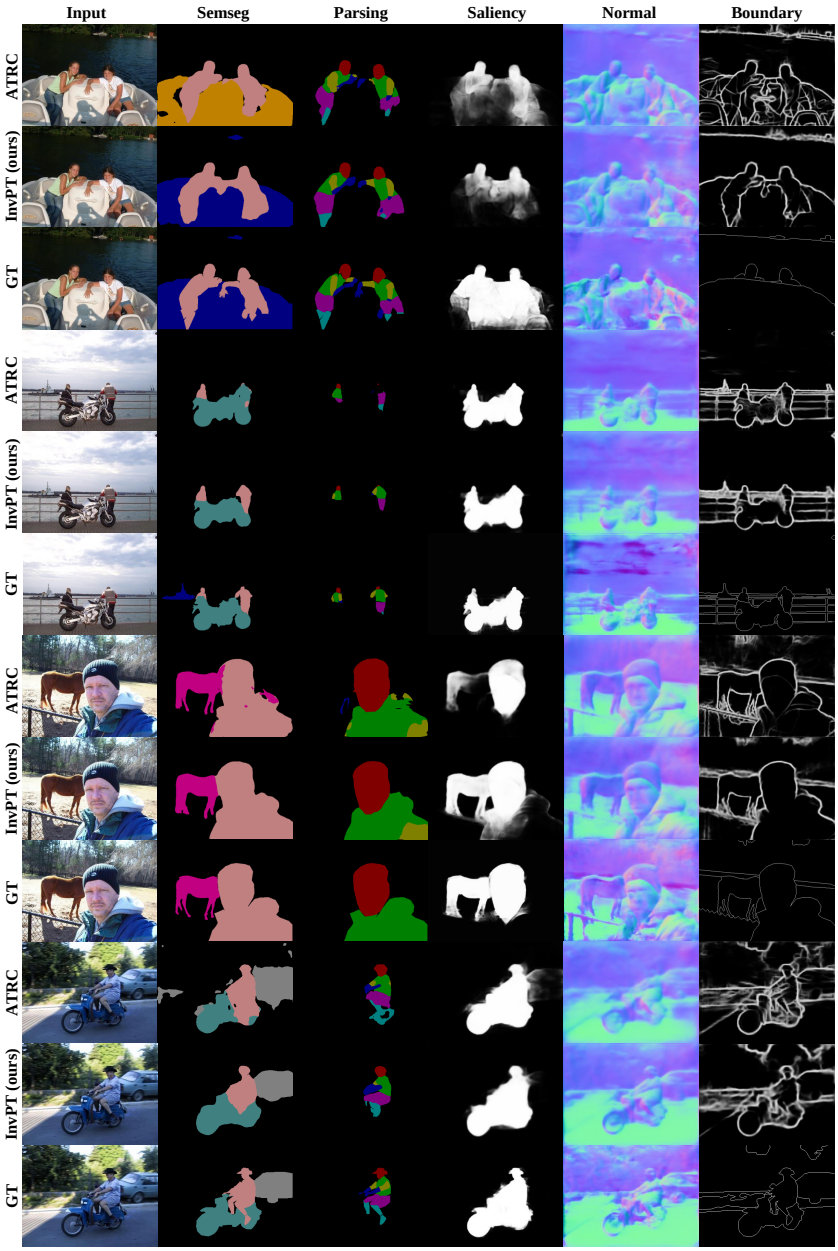

Fig. 2: Qualitative comparison with the best performing method ATRC [1] on PASCAL-Context. Our method generates significantly better results especially on semantic segmentation and human parsing.

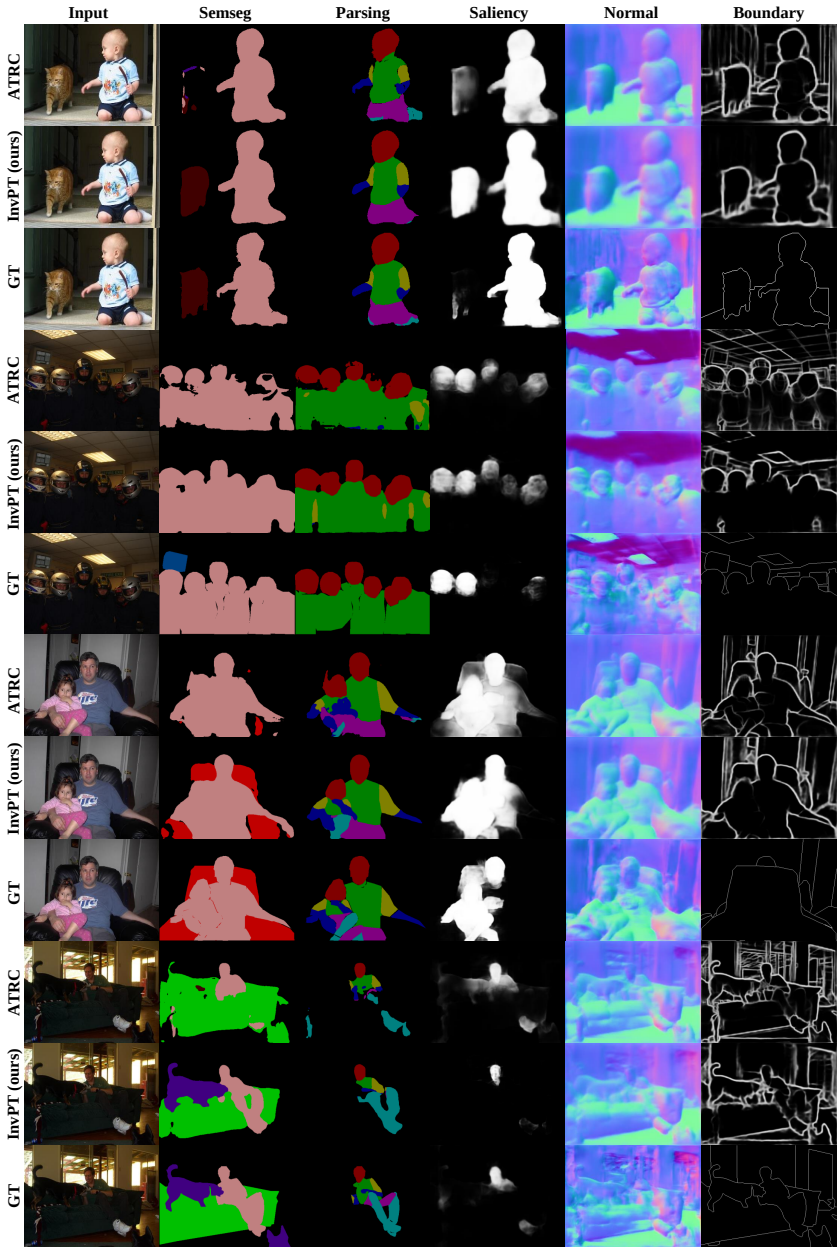

Fig. 3: Qualitative comparison with the best performing method ATRC [1] on PASCAL-Context. Our method generates significantly better predictions especially on semantic segmentation and human parsing.

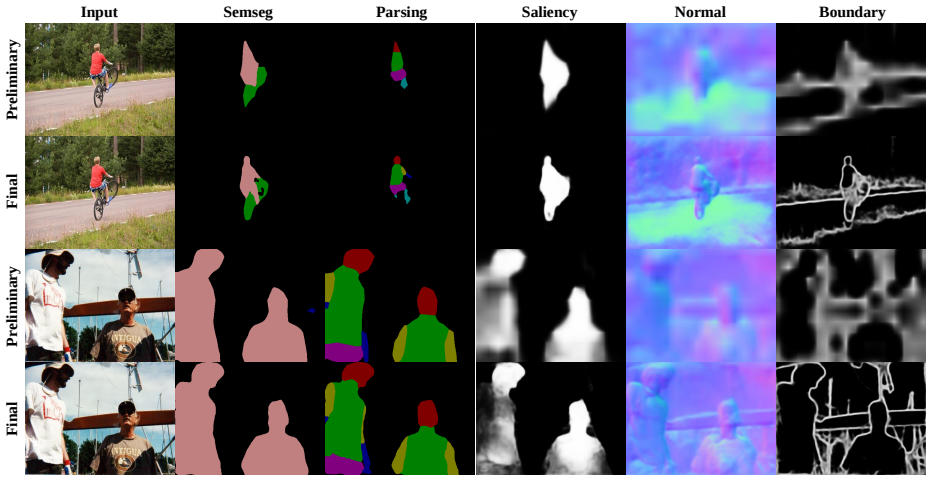

Fig. 4: Qualitative comparison of the predictions from the preliminary decoder and the final predictions of InvPT decoder on PASCAL-Context. The final predictions on all these tasks are significantly more accurate.

References

1. Bruggemann, D., Kanakis, M., Obukhov, A., Georgoulis, S., Van Gool, L.: Exploring relational context for multi-task dense prediction. In: ICCV (2021) 1, 2, 3, 5, 6
2. Chen, X., Mottaghi, R., Liu, X., Fidler, S., Urtasun, R., Yuille, A.: Detect what you can: Detecting and representing objects using holistic models and body parts. In: CVPR (2014) 1
3. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021) 1, 2, 3
4. 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) 1, 2
5. Perazzi, F., Pont-Tuset, J., McWilliams, B., Gool, L.V., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: CVPR (2016) 2
6. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: ECCV (2012) 1
7. Xu, D., Ouyang, W., Wang, X., Sebe, N.: Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In: CVPR (2018) 2