SUPPLEMENTARY MATERIAL: MULTI-TASK LEARNING WITH 3D-AWARE REGULARIZATION

Wei-Hong Li[1], **Steven McDonagh**[1], **Ales Leonardis**[2], **Hakan Bilen**[1]
[1]University of Edinburgh, [2]University of Birmingham
github.com/VICO-UoE/3DAwareMTL

1 IMPLEMENTATION DETAILS

We implement our approach in conjunction with state-of-the-art multi-task learning methods; MTI-Net (Vandenhende et al., 2020) and InvPT (Ye & Xu, 2022) while following identical training, evaluation protocols (Ye & Xu, 2022). We use HRNet-48 (Wang et al., 2020) and ViT-L (Dosovitskiy et al., 2020) to serve as shared encoders and append our 3D-aware regularizer to MTI-Net and InvPT using two convolutional layers, followed by BatchNorm, ReLU, and dropout layer with a dropout rate of 0.15 to transform feature maps to the tri-plane dimensionality, resulting in a common size and channel width (64). A 2-layer MLP with 64 hidden units as in Chan et al. (2022) and a LeakyReLU non-linearity with the negative slope of -0.2 as in Skorokhodov et al. (2022), is used to render each task as in Chan et al. (2022). We use identical hyper-parameters; learning rate, batch size, loss weights, loss functions, pre-trained weights, optimizer, evaluation metrics as MTI-Net and InvPT, respectively. We jointly optimize task-specific losses and losses arising from our 3D regularization. During inference, the regularizer is discarded. We use the same task-specific loss weights as in Ye & Xu (2022). We train all models for 40K iterations with a batch size of 6 for experiments of using InvPT as in Ye & Xu (2022) and a batch size of 8 for experiments of using MTI-Net as in (Vandenhende et al., 2020). We ramp up the α_t from 0 to 4 linearly in 20K iterations and keep $\alpha_t = 4$ for the rest 20K iterations. In the regularizer, we assume that the camera is orthogonal to image center, and depict r as a function that takes only the output of n_t but not the viewpoint as input. In a 3D coordinates (x, y, z), the x and y coordinates are aligned with pixel locations and z is the depth value. We further use a two-pass importance sampling as in NeRF (Mildenhall et al., 2020). For the majority of the experiments in the manuscript, we use 128 total depth samples per ray. We render 56×72 predictions for NYUv2 and 64×64 for PASCAL-Context and resize the predictions via bilinear interpolation to the groundtruth resolution. Our code and models will be made public based upon acceptance.

2 TRAINING COST ANALYSIS

Method	Time	Memory	Params.	FLOPS
MTI-Net (Vandenhende et al., 2020)	1.000	1.000	1.000	1.000
Ours	1.489	1.638	1.005	1.263
InvPT (Ye & Xu, 2022)	1.000	1.000	1.000	1.000
Ours	1.318	1.397	1.016	1.114

Table 1: Training Cost Comparisons to MTI-Net and InvPT; NYUv2 dataset. Note that our method has no additional inference cost as the regularizer is discarded during testing.

3 RESULTS OVER MULTIPLE RUNS

Here, we report the results of our method over 3 runs on NYUv2 and PASCAL-Context and report the results in Tabs. 2 and 3. From the results, we can see that our method is stable (i.e. the std is very small on each task) and improves over the baseline consistently on all tasks.

Here, we analyze memory and computational cost during training for tackling four tasks in NYUv2 and report them in Tab. 1. As shown in Tab. 1, our method that incorporates the regularizer to the MTL

Method	Seg. (mIoU) ↑	Depth (RMSE) ↓	Normal (mErr) ↓	Boundary (odsF) ↑
InvPT (Ye & Xu, 2022)	53.56	0.5183	19.04	78.10
Ours	**54.86 ± 0.29**	**0.5000 ± 0.0010**	**18.49 ± 0.09**	**78.17 ± 0.09**

Table 2: Quantitative comparison of our method to the InvPT over 3 runs; NYUv2 dataset.

Method	Seg. (mIoU) ↑	PartSeg (mIoU) ↑	Sal (maxF) ↑	Normal (mErr) ↓	Boundary (odsF) ↑
InvPT (Ye & Xu, 2022)	79.03	67.61	84.81	14.15	73.00
Ours	**79.92 ± 0.32**	**69.08 ± 0.15**	**84.85 ± 0.06**	**13.70 ± 0.14**	**73.83 ± 0.17**

Table 3: Quantitative comparison of our method to the InvPT over 3 runs; PASCAL-Context dataset.

baseline slightly increases the number of parameters (Ours vs InvPT: 1.016 vs 1) and FLOPS (Ours vs InvPT: 1.114 vs 1) during training, training time (Ours vs InvPT: 1.318 vs 1), and training memory (Ours vs InvPT: 1.397 vs 1). We highlight that there is *NO additional cost* during inference, since the regularizer will be discarded during inference.

4 Comparisons with more recent SotA

Method	Seg. (mIoU) ↑	Depth (RMSE) ↓	Normal (mErr) ↓	Boundary (odsF) ↑
TaskPromper (Ye & Xu, 2023a)	55.30	0.5152	**18.47**	78.20
TaskExpert (Ye & Xu, 2023b)	**55.35**	0.5157	18.54	**78.40**
InvPT (Ye & Xu, 2022)	53.56	0.5183	19.04	78.10
Ours	54.87	**0.5006**	18.55	78.30

Table 4: Quantitative comparison of our method to more SotA methods; NYUv2 dataset.

We include the comparisons of our method incorporated with InvPT to more recent state-of-the-art methods, including TaskPrompter (Ye & Xu, 2023a) and TaskExpert (Ye & Xu, 2023b) and report the results in Tabs. 4 and 5. Methods from Liu et al. (2023) and Chen et al. (2023) are not compared as they did not reported results on NYUv2 and PASCAL benchmarks with the same backbone. Note that TaskExpert (Ye et al., 2023b) is published after we submitting the manuscript. From the results shown in Tab. 4, we can see that, our method incorporated with InvPT achieves much better result on Depth while comparable results on the rest of tasks in NYUv2 compared with TaskPrompter and TaskExpert. In PASCAL benchmark, from Tab. 5 we can see that our method obtains much better results on saliency, surface normal and boundary estimation while obtaining comparable result on Human part segmentation and slightly worse on semantic segmentation. TaskPrompter adds learnable prompts for refining the feautures and the TaskExpert emsembles task-specific features from multiple task-specific experts for final task predictions and they all increase the capacity of the network to achieve better results. Also, they can potentially be complementary to our method and we believe incorporating our method with them can further improve the performance in multi-task learning by regulating the shared features to be 3D-aware with no additional cost during inference.

5 Discussion

Camera parameters. In our paper, the 3D coordinates and strategy of projecting the 3D coordinates onto the feature planes are similar to the ones in PiFU (Saito et al., 2019) and (Yao et al., 2023). The feature planes are generated by the feature encoder and it is pixel-wise feature map instead of a global pooled feature vector. The x and y coordinates are aligned with pixel locations and z is the depth value. We follow Chan et al. (2022) that projects the coordinates (x, y, z) onto three planes e_{xy}, e_{yz}, e_{xz}, retrieving the features via bilinear interpolation, and aggregates features from three planes instead of taking the 2D features and the z values as representations in PiFU (Saito et al., 2019) or dividing the dimension of the feature map channel into D groups (D is the number of depth bins) in (Yao et al., 2023). So our method has similar property as in PiFU (Saito et al., 2019) and (Yao et al., 2023) and does not overfit to the camera parameters.

Also, as we first feed the image into the feature encoder, which should be scaling the 3D coordinates accordingly and the coordinates will not be absolute but at the right scale for rendering. After training, the 3D-aware regularizer is discarded and we only use the multi-task learning branch for generating predictions for different tasks. We also visualize multiple images' predictions of the regularizers on

Method	Seg. (mIoU) ↑	PartSeg (mIoU) ↑	Sal (maxF) ↑	Normal (mErr) ↓	Boundary (odsF) ↑
TaskPrompter (Ye & Xu, 2023a)	**80.89**	68.89	84.83	13.72	73.50
TaskExpert (Ye & Xu, 2023b)	80.64	**69.42**	84.87	13.56	73.30
InvPT (Ye & Xu, 2022)	79.03	67.61	84.81	14.15	73.00
Ours	79.53	69.12	**84.94**	**13.53**	**74.00**

Table 5: Quantitative comparison of our method to the SotA methods; PASCAL-Context dataset.

PASCAL in Fig. 1. The PASCAL dataset consists of annotated consumer photographs collected from the flickr photo-sharing web-site, taken by various cameras with different intrinsics. From Fig. 1, we can see that the regularizer can render good quality predictions for all tasks on all images which also indicates that it does not overfit to the camera parameters.

Figure 1: Qualitative results on PASCAL. Each column shows the image or predictions of our method's regularizer branch or the groundtruth for each task, respectively.

Limitations and future work. Despite the efficient 3D modeling through the triplane encodings, representing 3D representations for higher resolution 3D volumes is still expensive in terms of memory or computational cost. Some common efficient sampling strategies such as random sampling and pixel binning can be useful for reducing the cost. The tri-plane generated from the feature encoder can be relatively small resolution due to the feature downsampling and requires upsampling strategies for generating higher resolution feature planes for better rendering while it will inevitably increase the training cost. Additionally, rendering specular objects will require different rendering or objects with high frequency 3D details may require more accurate 3D modeling. Though our proposed method obtains performance gains consistently over multiple tasks, we balance loss functions with fixed cross-validated hyperparameters, while it would be more beneficial to use adaptive loss balancing strategies (Kendall et al., 2018) or discarding conflicting gradients (Liu et al., 2021). Finally, in the cross-view consistency experiments where only some of the images are labeled for all the tasks, our method does not make use of semi-supervised learning or view-consistency for the tasks with

missing labels which can be further improve the performance of our model. We believe that more advanced techniques in 3D modeling can further improve our method for rendering higher resolution predictions with higher efficiency and better regulating the cross-task correlations and cross-view consistency.

REFERENCES

Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *CVPR*, 2022.

Zitian Chen, Yikang Shen, Mingyu Ding, Zhenfang Chen, Hengshuang Zhao, Erik G Learned-Miller, and Chuang Gan. Mod-squad: Designing mixtures of experts as modular multi-task learners. In *CVPR*, pp. 11828–11837, 2023.

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

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, pp. 7482–7491, 2018.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *NeurIPS*, 2021.

Yajing Liu, Yuning Lu, Hao Liu, Yaozu An, Zhuoran Xu, Zhuokun Yao, Baofeng Zhang, Zhiwei Xiong, and Chenguang Gui. Hierarchical prompt learning for multi-task learning. In *CVPR*, pp. 10888–10898, 2023.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *ICCV*, pp. 2304–2314, 2019.

Ivan Skorokhodov, Sergey Tulyakov, Yiqun Wang, and Peter Wonka. Epigraf: Rethinking training of 3d gans. *Neurips*, 35:24487–24501, 2022.

Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Mti-net: Multi-scale task interaction networks for multi-task learning. In *ECCV*, pp. 527–543. Springer, 2020.

Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *TPAMI*, 43(10):3349–3364, 2020.

Jiawei Yao, Chuming Li, Keqiang Sun, Yingjie Cai, Hao Li, Wanli Ouyang, and Hongsheng Li. Ndc-scene: Boost monocular 3d semantic scene completion in normalized device coordinates space. In *ICCV*, pp. 9455–9465, 2023.

Hanrong Ye and Dan Xu. Inverted pyramid multi-task transformer for dense scene understanding. In *ECCV*, pp. 514–530. Springer, 2022.

Hanrong Ye and Dan Xu. Taskprompter: Spatial-channel multi-task prompting for dense scene understanding. In *ICLR*, 2023a.

Hanrong Ye and Dan Xu. Taskexpert: Dynamically assembling multi-task representations with memorial mixture-of-experts. In *ICCV*, pp. 21828–21837, 2023b.