Supplementary Material: Learning Multiple Dense Prediction Tasks from Partially Annotated Data

Wei-Hong Li, Xialei Liu, and Hakan Bilen
VICO Group, University of Edinburgh, United Kingdom
github.com/VICO-UoE/MTPSL

1. Implementation Details

Tab. 1 and Tab. 2 provide an overview of the experimental settings, in particular report the number of train and test samples for each benchmark and number of labels used in different partially annotated settings respectively. Next we explain the implementation details for each dataset.

Cityscapes. The Cityscapes dataset [6] contains 3475 labelled images. As in [10], we use 2975 images for training and 500 images for testing. In multi-task partially supervised learning setting, we consider the one-label setting in Cityscapes, as there are only two tasks in total, *i.e.* we randomly select and keep label only for 1 task for each training image, resulting in 1487 training images annotated for segmentation and 1488 training images labelled for depth estimation, as shown in Tab. 2.

We follow the training and evaluation protocol in [10] and we use SegNet [1] as the MTL backbone for all methods, use cross-entropy loss for semantic segmentation, l1-norm loss for depth estimation. We use the exactly same hyper-parameters including learning rate, optimizer as in [10]. More specifically, we use Adam optimizer with a learning rate of 0.0001 and train all models for 200 epochs with a batch size of 8 and halve the learning rate at the 100-th epoch. We also employ the same evaluation metrics, mean intersection over union (mIoU) and absolute error (aErr) to evaluate the semantic segmentation and depth estimation task, respectively as in [10].

For our model, we use the encoder architecture of SegNet for instantiating the joint pairwise task mapping ($\bar{m}_\vartheta$) and include one convolutional layer as task specific input layer in $\bar{m}_\vartheta$. For `Direct-Map` and `Perceptual-Map`, as in [15] we use the whole SegNet as the cross-task mapping functions. We use the same data augmentations from the updated implementation in [10][1], *i.e.* random crops and rand horizontal flips.

NYU-v2. The dataset [13] contains 795 training images and 654 test images. To evaluate the multi-task partially supervised learning, we consider one-label and random-label settings. For one-label setting, we randomly select and keep label for only 1 task for each training image, resulting in 265 images with annotation for segmentation, 265 images labelled for depth estimation and 265 images for surface normal. For random-label setting, we randomly select and keep labels for at least 1 and at most 2 tasks (1.49 labels per image), *i.e.* 392 images for semantic segmentation, 408 images for depth estimation, 385 images for surface normal, as shown in Tab. 2.

Dataset	# Train	# Test	Segmentation	Depth	Human Parts	Normals	Saliency	Edges
Cityscapes [6]	2975	500	✓	✓	- -	- -	- -	- -
NYU-v2 [13]	795	654	✓	✓	- -	✓	- -	- -
PASCAL [4]	4998	5105	✓	- -	✓	✓	✓	✓

Table 1. Details of multi-task benchmarks.

Dataset	# label	# labelled images					
		Segmentation	Depth	Human Parts	Normals	Saliency	Edges
Cityscapes [6]	one	1487	1488	- -	- -	- -	- -
NYU-v2 [13]	random	392	408	- -	385	- -	- -
	one	265	265	- -	265	- -	- -
PASCAL [4]	random	2450	- -	2553	2480	2445	2557
	one	1000	- -	999	1000	1000	999

Table 2. Details about multi-task partially supervised learning settings in three benchmarks used in this work. 'random' means the random-label setting where each training image has a random number of task labels and 'one' indicates the one-label setting where each training image is annotated with one task label. '# labelled images' shows the number of images containing labels for each task, *e.g.* segmentation.

We follow the training and evaluation protocol in [10] and we use the the SegNet [1] as the MTL backbone for all methods. As in [10], we use cross-entropy loss for semantic segmentation, l1-norm loss for depth estimation and cosine similarity loss for surface normal estimation, use the same optimizer and hyper-parameters, *i.e.* Adam optimizer with a learning rate of 0.0001. We train the all model for 200 epochs with a batch size of 2 and halve the learning rate at the 100-th epoch and employ the same evaluation metrics, mean intersection over union (mIoU), absolute error (aErr) and mean error (mErr) in the predicted angles to evaluate

[1] https://github.com/lorenmt/mtan

the semantic segmentation, depth estimation and surface normals estimation task, respectively as in [10].

We use the encoder of SegNet architecture for the joint pairwise task mapping ($\bar{m}_\vartheta$) and one convolutional layer as task specific input layer in $\bar{m}_\vartheta$. For `Direct-Map` and `Perceptual-Map`, as in [15] we use the whole SegNet as the cross-task mapping functions. To regularize training, we use the exact same data augmentations from the updated implementation from [10], *e.g.* random crops and rand horizontal flips augmentations.

PASCAL-context. The dataset [4] contains 4998 training images and 5105 testing images for five tasks, *i.e.* semantic segmentation, human parts segmentation, surface normal, saliency detection and edge detection. We consider two partially supervised learning settings, random-label and one-label setting. For one-label setting, we have 1 label per image, *i.e.* 1000, 999, 1000, 1000, 999 labelled images for semantic segmentation, human parts, surface normal, saliency and edge detection, respectively. In random-label setting, we randomly sample and keep labels for at least 1 and at most 4 tasks (2.50 labels per image), resulting in 2450, 2553, 2480, 2445, 2557 labelled images for semantic segmentation, human parts, surface normal, saliency and edge detection, respectively, as shown in Tab. 2.

We follow exactly the same training, evaluation protocol and implementation in [14] and employ the ResNet-18 [7] as the encoder shared across all tasks and Atrous Spatial Pyramid Pooling (ASPP) [3] module as task-specific heads. We use the same hyper-parameters, *e.g.* learning rate, augmentation, loss functions, loss weights in [14]. More specifically, we use Adam as the optimizer with a learning rate of 0.0001 and a weight decay of 0.0001. As in [14] all experiments are performed using pre-trained ImageNet weights. We train all multi-task learning methods for 100 epochs with a batch size of 6 and we anneal the learning rate using the 'poly' learning rate scheduler as in [2, 14]. We follow [14] and use fixed loss weights for training all multi-task learning methods, *i.e.* the loss weight is 1, 2, 10, 5, 50 for semantic segmentation, human parts segmentation, surface normal estimation, saliency detection and edge detection, respectively. Please refer to [14] for more details. For evaluation metrics, we use the optimal dataset F-measure (odsF) [11] for edge detection, the standard mean intersection over union (mIoU) for semantic segmentation, human part segmentation and saliency estimation are evaluated, mean error (mErr) for surface normals. We modify the ResNet-18 to have task specific input layers (one convolutional layer for each task) before the residual blocks as the mapping function $\bar{m}_\vartheta$ in our method.

Multi-task performance. Following prior work [14], we also report the multi-task performance $\triangle$MTL of the multi-task learning model as the average per-task drop in performance w.r.t. the single-task baseline:

$$\triangle\text{MTL} = \frac{1}{K}\sum_{t=1}^{K}(-1)^{\ell_i}(P_t^{mtl} - P_t^{stl})/P_t^{stl}, \qquad (1)$$

where $\ell_i = 1$ if a lower value of P_t means better performance for metric of task t, and 0 otherwise.

2. More results

Here, we report more results from single-task learning (STL) model, Contrastive-Loss and Discriminator-Loss and also qualitative results.

2.1. Quantitative results

Results on Cityscapes. Here, we report the results on Cityscapes for only *one* label setting as there are two tasks in total in Tab. 3. We also report results of single-task learning models which are used to compute the multi-task performance ($\triangle$MTL) to better analyze the results as in [14]. The performance of MTL methods are worse than single-task learning models for some tasks as the MTL models have less capacity and there is a problem of imbalanced optimization etc as discussed in [8, 9, 14].

The results of MTL model learned with SL when all task labels are available for training to serve as a strong baseline for multi-task learning methods. In the partial label setting (one task label per image), the performance of the SL baseline drops substantially compared to its performance in full supervision setting. While the SSL baseline, by extracting task-specific information from unlabelled tasks, improves over SL, further improvements are obtained by exploiting cross-task consistency in various ways except Discriminator-Loss. The methods learn mappings from one task to another one (Perceptual-Map and Direct-Map) surprisingly perform better than the ones learning joint space mapping functions (Contrastive-Loss and Discriminator-Loss), possibly due to insufficient number of negative samples. Finally, the best results (*e.g.* the best multi-task performance $\triangle$MTL) are obtained with our method that can exploit cross-task relations more efficiently through joint pairwise task mappings with the proposed regularization. Interestingly, our method also outperforms the SL baseline that has access to all the task labels, showing the potential information in the cross-task relations.

Results on Cityscapes with larger images. We also provide results for 256 × 512 setting in Tab. 4. Performance of all methods improve significantly compared to their ones using small images (in Tab. 3) and our method achieves significant improvement over the baselines.

Results on NYU-v2 Here, we evaluate our method and related methods in the *random* and *one* label settings on NYU-v2 and we report the results in Tab. 5. We also report

# label	Type	Method	Seg. (IoU) ↑	Depth (aErr) ↓	△MTL ↑
full	STL	Supervised Learning	74.19	0.0124	+0.00
	MTL	Supervised Learning	73.36	0.0165	-17.00
one	STL	Supervised Learning	70.26	0.0141	+0.00
	MTL	Supervised Learning	69.50	0.0186	-16.55
		Semi-supervised Learning	71.67	0.0178	-12.22
		Perceptual-Map	72.82	0.0169	-8.37
		Direct-Map	72.33	0.0179	-11.94
		Contrastive-Loss	71.79	0.0183	-13.77
		Discriminator-Loss	68.94	0.0208	-24.95
		Ours	**74.90**	**0.0161**	**-3.81**

Table 3. Multi-task learning results on Cityscapes. 'one' indicates each image is randomly annotated with one task label. 'STL' means single task learning and 'MTL' indicates multi-task learning.

# label	Type	Method	Seg. (IoU) ↑	Depth (aErr) ↓	△MTL ↑
one	STL	Supervised Learning	77.97	0.0126	+0.00
	MTL	Supervised Learning	77.71	0.0165	-15.95
		Semi-supervised Learning	79.24	0.0161	-13.38
		Ours	**82.41**	**0.0143**	**-4.08**

Table 4. Multi-task learning results on Cityscapes using 256 × 512 images. 'one' indicates each image is randomly annotated with one task label. 'STL' means single task learning and 'MTL' indicates multi-task learning.

results of single-task learning models which are used to compute the multi-task performance (△MTL) to better analyze the results as in [14].

While we observe a similar trend across different methods, overall the performances are lower in this benchmark possibly due to fewer training images than CityScapes. As expected, the performance in random-label setting is better than the one in one-label setting, as there are more labels available in the former. While the best results are obtained with SL trained on the full supervision, our method obtains the best performance (*e.g.* best results on all tasks and the best multi-task performance) among the partially supervised methods. Here SSL improves over SL trained on the partial labels and cross-task consistency is beneficial except for Direct-Map in the one label setting and Discriminator-Loss, possibly because the dataset is too small to learn accurate mappings between two tasks, while our method is more data-efficient and more successful to exploit the cross-task relations. In random-label setting, where images might have labels for more than one task, we also report our method also leveraging the labelled corss-task relations ('Ours+') in Tab. 5 and it can indeed further boost the average performance.

Results on PASCAL. We evaluate all methods on PASCAL-Context, in both label settings, which contains wider variety of tasks than the previous benchmarks and report the results in Tab. 6. As in Cityscapes and NYU-v2, we also report results of single-task learning models which are used to compute the multi-task performance (△MTL) to better analyze the results as in [14].

As the required number of pairwise mappings for Direct-Map and Perceptual-Map grows quadratically (20 mappings for 5 tasks), we omit these two due to their high computational cost and compare our method only to SL, SSL, Contrastive-Loss and Discriminator-Loss baselines. We see that the SSL baseline improves the performance over SL in random-label setting, however, it performs worse than the SL in one label setting, when there are 60% less labels. By leveraging cross-task consistency, Contrastive-Loss and Discriminator-Loss obtains better performance than the SL baseline in one label setting while they get similar multi-task performance to the SL baseline in random label setting. Again, by exploiting task relations, our method obtains better or comparable results to second best method, *i.e.* SSL, while the gains achieved over SL and SSL are more significant in the low label regime (one-label). Interestingly, SSL and our method obtain comparable results in random-label setting which suggests that relations across tasks are less informative than the ones in CityScape and NYUv2.

# labels	Type	Method	Seg. (IoU) ↑	Depth (aErr) ↓	Norm. (mErr) ↓	△MTL ↑
full	STL	Supervised learning	37.45	0.6079	25.94	+0.00
	MTL	Supervised learning	36.95	0.5510	29.51	-1.92
random	STL	Supervised Learning	28.72	0.7540	28.95	+0.00
	MTL	Supervised Learning	27.05	0.6624	33.58	-3.23
		Semi-supervised Learning	29.50	0.6224	33.31	+1.70
		Perceptual-Map	32.20	0.6037	32.07	+7.10
		Direct-Map	29.17	0.6128	33.63	+1.38
		Contrastive-Loss	30.75	0.6143	32.05	+4.96
		Discriminator-Loss	26.76	0.6354	33.13	-1.84
		Ours	34.26	0.5787	**31.06**	+11.81
		Ours+	**34.91**	**0.5738**	31.20	**+12.57**
one	STL	Supervised Learning	24.71	0.7666	30.14	+0.00
	MTL	Supervised Learning	25.75	0.6511	33.73	+1.14
		Semi-supervised Learning	27.52	0.6499	33.58	+3.16
		Perceptual-Map	26.94	0.6342	34.30	+2.31
		Direct-Map	19.98	0.6960	37.56	-12.86
		Contrastive-Loss	26.65	0.6387	34.69	+1.31
		Discriminator-Loss	25.68	0.6566	34.02	+0.04
		Ours	**30.36**	**0.6088**	**32.08**	**+10.24**

Table 5. Multi-task learning results on NYU-v2. 'random' indicates each image is annotated with a random number of task labels and 'one' means each image is randomly annotated with one task. 'STL' means single task learning and 'MTL' indicates multi-task learning.

# labels	Type	Method	Seg. (IoU) ↑	H. Parts (IoU) ↑	Norm. (mErr) ↓	Sal. (IoU) ↑	Edge (odsF) ↑	△MTL ↑
full	STL	Supervised Learning	66.4	58.9	13.9	66.7	68.3	+0.00
	MTL	Supervised Learning	63.9	58.9	15.1	65.4	69.4	-2.75
random	STL	Supervised Learning	60.9	55.3	14.7	64.8	66.8	+0.00
	MTL	Supervised Learning	58.4	55.3	16.0	63.9	**67.8**	-2.67
		Semi-supervised Learning	**59.0**	**55.8**	**15.9**	**64.0**	66.9	-2.44
		Contrastive-Loss	**59.0**	55.3	16.0	63.8	**67.8**	-2.44
		Discriminator-Loss	57.9	55.2	16.2	63.4	67.4	-3.35
		Ours	**59.0**	55.6	**15.9**	**64.0**	**67.8**	**-2.15**
one	STL	Supervised Learning	47.7	56.2	16.0	61.9	64.0	+0.00
	MTL	Supervised Learning	48.0	55.6	17.2	61.5	64.6	-1.34
		Semi-supervised Learning	45.0	54.0	**16.9**	**61.7**	62.4	-3.02
		Contrastive-Loss	48.5	55.4	17.1	61.3	64.6	-1.25
		Discriminator-Loss	48.2	**56.0**	17.1	61.7	64.7	-1.04
		Ours	**49.5**	55.8	17.0	**61.7**	**65.1**	**-0.40**

Table 6. Multi-task learning results on PASCAL. 'random' indicates each image is annotated with a random number of task labels and 'one' means each image is randomly annotated with one task. 'STL' means single task learning and 'MTL' indicates multi-task learning.

Learning from partial and imbalanced task labels. We also evaluate our method and baselines in an imbalanced partially supervised setting in Cityscapes, where we assume the ratio of labels for each task are imbalanced, *e.g.* we randomly sample 90% of images to be labeled for semantic segmentation and only 10% images having labels for depth and we denote this setting by the label ratio between segmentation and depth (Seg.:Depth = 9:1). The opposite case (Seg.:Depth = 1:9) is also considered. We report the results in Tab. 7, where we also report results of single-task learning models which are used to compute the multi-task performance ($\triangle$MTL) to better analyze the results as in [14].

#labels	Type	Method	Seg. (IoU) ↑	Depth (aErr) ↓	$\triangle$MTL ↑
full	STL	Supervised learning	74.19	0.0124	+0.00
	MTL	Supervised Learning	73.36	0.0165	-17.00
1:9	STL	Supervised learning	62.23	0.0126	+0.00
	MTL	Supervised Learning	63.37	0.0161	-13.07
		Semi-supervised Learning	64.40	0.0179	-19.36
		Perceptual-Map	68.84	0.0141	-0.68
		Direct-Map	67.04	0.0153	-6.90
		Contrastive-Loss	67.12	0.0151	-5.95
		Discriminator-Loss	68.92	0.0144	-1.80
		Ours	**71.89**	**0.0131**	**+5.63**
9:1	STL	Supervised learning	72.62	0.0191	+0.00
	MTL	Supervised learning	72.77	0.0250	-15.25
		Semi-supervised Learning	72.97	0.0395	-53.11
		Perceptual-Map	73.36	0.0237	-11.34
		Direct-Map	73.13	0.0288	-19.38
		Contrastive-Loss	73.75	0.0243	-12.86
		Discriminator-Loss	72.97	0.0248	-14.65
		Ours	**74.23**	**0.0235**	**-10.23**

Table 7. Multi-task learning results on Cityscapes. '#label' indicates the number ratio of labels for segmentation and depth, *e.g.* '1:9' means we have 10% of images annotated with segmentation labels and 90% of images have depth groundtruth. 'STL' means single task learning and 'MTL' indicates multi-task learning.

The performance of supervised learning (SL) on the task with partial labels drops significantly. Though SSL improves the performance on segmentation, its performance on depth drops in both cases. Different from SSL, Direct-Map, Contrastive-Loss and Discriminator-Loss improves the performance on both tasks in 1:9 setting while their performance on depth drop in the 9:1 case. In contrast to SL and the baselines, our method and Perceptual-Map obtain better results on all tasks in both settings by learning cross-task consistency while our method obtains the best performance (*i.e.* best results in all tasks and best multi-task performance, $\triangle$MTL) by joint space mapping. This demonstrates that our model can successfully learn cross-task relations from unbalanced labels thanks to its task agnostic mapping function which can share parameters across multiple task pairs.

Cross-task consistency learning in conventional semi-supervised learning. We evaluate our method and SSL baseline on conventional SSL setting where $\frac{1}{3}$ of training data in NYU-v2 are labeled for all tasks and $\frac{2}{3}$ are unlabeled, and report the results in Tab. 8. In this setting, our method obtains better performance than SL and SSL. We will include a more detailed analysis in the final paper.

Type	Method	Seg. (IoU) ↑	Depth (aErr) ↓	Norm. (mErr) ↓	$\triangle$MTL ↑
MTL	Supervised Learning	24.78	0.6681	33.90	+1.48
	Semi-supervised Learning	26.09	0.6510	33.60	+4.37
	Ours	**28.43**	**0.6366**	**33.01**	**+8.83**

Table 8. Multi-task learning results on NYU-v2 in SSL setting where $\frac{1}{3}$ of training data in NYU-v2 are labeled for all tasks and $\frac{2}{3}$ are unlabeled. 'MTL' indicates multi-task learning.

Cross-task consistency learning with full supervision. Our method can also be applied to fully-supervised learning setting where all task labels are available for each sample by mapping one task's prediction and another task's groundtruth to the joint space and measuring cross-task consistency in the joint space. We applied our method to NYU-v2 and compare it with the single task learning (STL) networks, vanilla MTL baseline, recent multi-task learning methods, *i.e.* MTAN [10], X-task [15], and several methods focusing on loss weighting strategies, *i.e.* Uncertainty [8], GradNorm [5], MGDA [12] and DWA [10] in Tab. 9. Here, we also report the multi-task performance ($\triangle$MTL) of all MTL methods.

Method	Seg. (IoU) ↑	Depth (aErr) ↓	Norm. (mErr) ↓	$\triangle$MTL
STL	37.45	0.6079	25.94	+0.00
MTL	36.95	0.5510	29.51	-1.92
MTAN [10]	39.39	0.5696	28.89	+0.03
X-task [15]	38.91	0.5342	29.94	+0.89
Uncertainty [8]	36.46	0.5376	27.58	+0.86
GradNorm [5]	37.19	0.5775	28.51	-1.86
MGDA [12]	38.65	0.5572	28.89	+0.06
DWA [10]	36.46	0.5429	29.45	-1.82
Ours	41.00	0.5148	28.58	+4.88
Ours + Uncertainty	**41.09**	**0.5090**	**26.78**	**+7.57**

Table 9. Multi-task fully-supervised learning results on NYU-v2. 'STL' indicates standard single-task learning and 'MTL' means the standard multi-task learning network.

MTL, MTAN, X-task and Ours are trained with uniform loss weights. We see that our method (Ours) performs better than the other methods with uniform loss weights, *e.g.* MTAN and X-task, where X-task regularizes cross-task consistency by learning perceptual loss with pre-trained cross-task mapping functions. This shows that cross-task consistency is informative even in the fully supervised case and our method is more effective for learning cross-task consistency. Compared to recent loss weighting strategies, our method (Ours) obtains better multi-task performance ($\triangle$MTL) and better performance on segmentation and depth estimation than other methods while slightly worse on normal estimation compared with GradNorm and Uncertainty. This is because the loss weighting strategies enable a more balanced optimization of multi-task learning model than uniformly loss weighting. Thus when we incorporate the loss weighing strategy of Uncertainty [8] to our method, *i.e.* (Ours + Uncertainty), our method obtains further improvement and outperforms both GradNorm and Uncertainty, *e.g.* 'Ours + Uncertainty' obtains the best multi-task performance (+7.57).

2.2. Qualitative results

Here, we present some qualitative results.

Mapped outputs. Here, we visualize the intermediate feature maps of $m^{s \to st}$ and $m^{t \to st}$ for one example in Cityscapes in Fig. 1 where s and t correspond to segmentation and depth estimation respectively and one example in NYU-v2 in Fig. 2 where s and t correspond to segmentation and surface normal estimation respectively. We observe that the functions map both task labels to a joint pairwise space where the common information is around object boundaries, which in turn enables the model to produce more accurate predictions for both tasks.

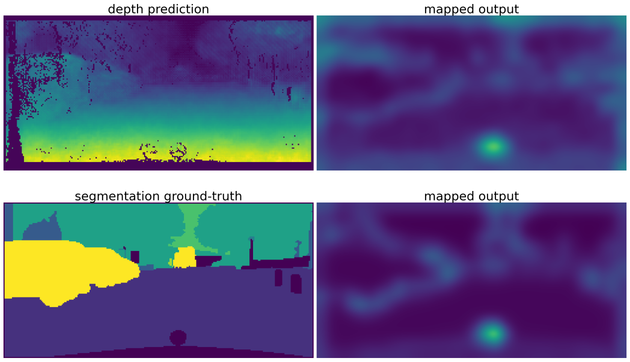

Figure 1. Intermediate feature map of the mapping function of the task-pair (segmentation to depth) of one example in Cityscapes. The first column shows the prediction or ground-truth and the second column present the corresponding mapped feature map (output of the mapping function's last second layer).

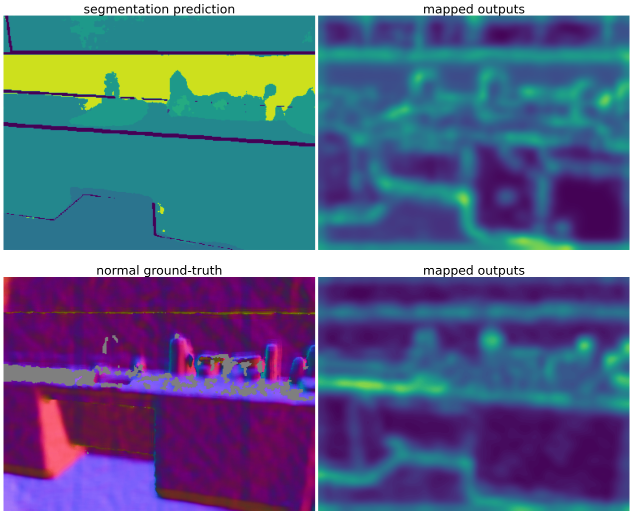

Figure 2. Intermediate feature map of the mapping function of the task-pair (segmentation to surface normal) of one example in NYU-v2. The first column shows the prediction or ground-truth and the second column present the corresponding mapped feature map (output of the mapping function's last second layer).

Predictions. Finally we show qualitative comparisons between our method, SL and SSL baselines, Perceptual-Map (PM), Direct-Map (DM), Contrastive-Loss (CL) and Discriminator-Loss (DL) on Cityscapes in Fig. 3 and on NYU-v2 in Fig. 4. We can see that our method produces more accurate predictions by leveraging cross-task consistency. Specifically, in Fig. 3, compared with methods that do not leverage cross-task consistency, the prediction of segmentation and depth are improved by our method (top left region) and our results are more accurate than related baselines (PM, DM, CL and DL). In Fig. 4, we can see that SSL produces more accurate predictions on segmentation and surface normal than SL. And PM obtains more accurate results on depth and surface normal than SL. While they do not achieve consistent improvement on all three tasks, our method can improve the results consistently on three tasks which shows that our method is more effective on learning cross-task consistency for MTL from partially annotated data.

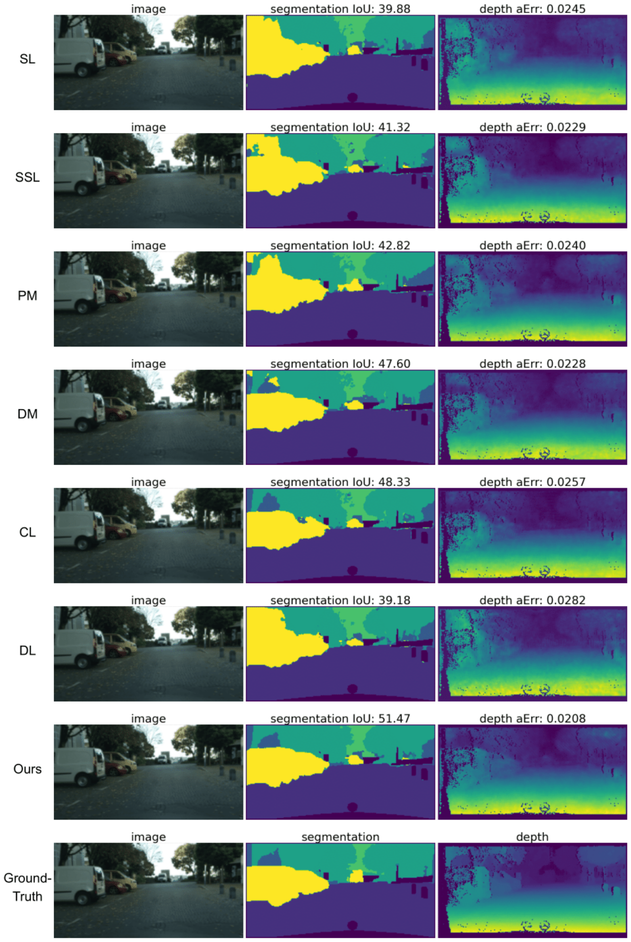

Figure 3. **Qualitative results on Cityscapes.** The fist column shows the RGB image, the second column plots the ground-truth or predictions with the IoU (↑) score of all methods for semantic segmentation and we show the ground-truth or predictions with the absolute error (↓) in the last column.

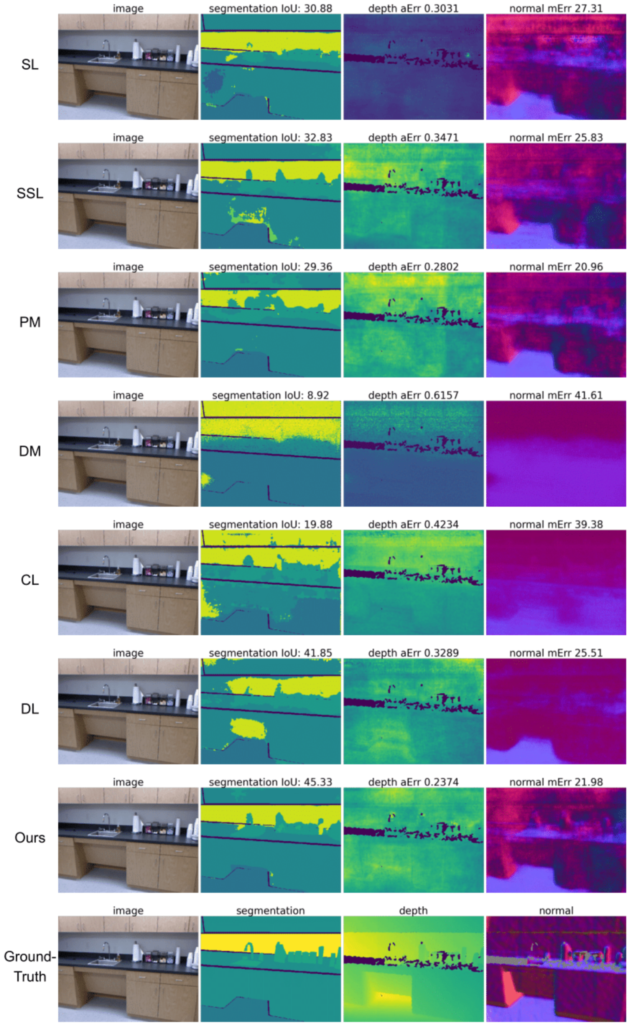

Figure 4. **Qualitative results on NYU-v2.** The fist column shows the RGB image, the second column plots the ground-truth or predictions with the IoU (↑) score of all methods for semantic segmentation, the third column presents the ground-truth or predictions with the absolute error (↓), and we show the prediction of surface normal with mean error (↓) in the last column.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *PAMI*, 39(12):2481–2495, 2017. 1

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *PAMI*, 40(4):834–848, 2017. 2

[3] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 801–818, 2018. 2

[4] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *CVPR*, pages 1971–1978, 2014. 1, 2

[5] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *ICML*, pages 794–803. PMLR, 2018. 4

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Computer Vision and Pattern Recognition*, pages 3213–3223, 2016. 1

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2

[8] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, pages 7482–7491, 2018. 2, 4

[9] Wei-Hong Li and Hakan Bilen. Knowledge distillation for multi-task learning. In *ECCV Workshop on Imbalance Problems in Computer Vision*, pages 163–176. Springer, 2020. 2

[10] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *CVPR*, pages 1871–1880, 2019. 1, 2, 4

[11] David R Martin, Charless C Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, 26(5):530–549, 2004. 2

[12] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *NeurIPS*, 2018. 4

[13] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012. 1

[14] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *PAMI*, 2021. 2, 3, 4

[15] Amir R Zamir, Alexander Sax, Nikhil Cheerla, Rohan Suri, Zhangjie Cao, Jitendra Malik, and Leonidas J Guibas. Robust learning through cross-task consistency. In *CVPR*, pages 11197–11206, 2020. 1, 2, 4