Look Deeper into Depth: Monocular Depth Estimation with Semantic Booster and Attention-Driven Loss (Supplementary Material)

Anonymous ECCV submission

Paper ID 71

Overview

In this supplementary material, we present more details, analysis, and experimental results to facilitate a better understanding of the main paper. Specifically, this material includes the following information:

1. Attention Visualization Details
2. Synergy Representation Learning
3. Network Architecture Details
4. Analysis on Robustness to Depth Range
5. Evaluation on Attention-Driven Loss
6. More Qualitative Results of Depth Estimation
7. Qualitative Results of Semantic Labeling
8. Outdoor Scene Training Details

1 Attention Visualization

Here we show more attetion visualization of the proposed network and present details of the visualization process. As presented in [1], given a CNN layer and the corresponding feature activation tensor $T \in R^{C\times H\times W}$, the attention map is defined as,

$$A(T) = \sum_{i=1}^{C} |T_i|, \tag{1}$$

where A is the attention map and C is the number of channels of the tensor. Taking the activation from the output of the first up-conv layer in the depth estimation branch, we show the attention maps of the network on monocular depth estimation in Fig. 1 (second column), in addition to Fig. 5 in the main paper. Similarly, the attention maps of the shared backbone and semantic labeling are extracted from the layers near the network bottleneck, as shown in the last two columns of Fig. 1.

2 Synergy Representation Learning

In order to leverage semantic information as guidance for monocular depth prediction, here we explore four representative information sharing structures, as shown in Fig.2.

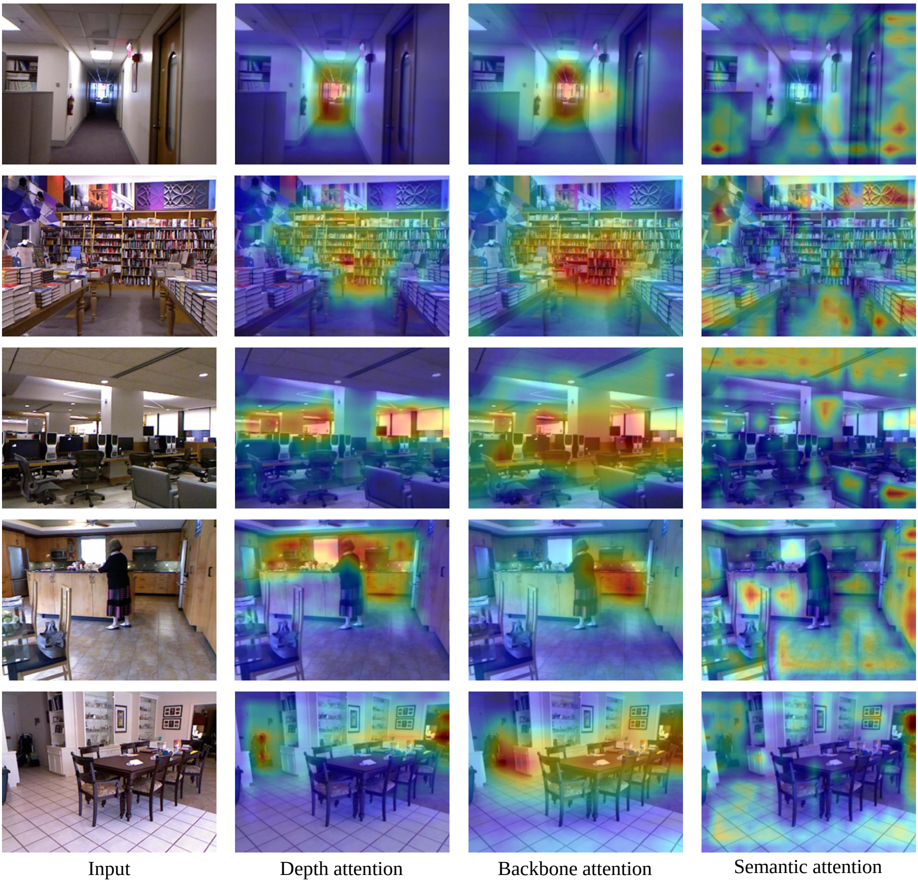
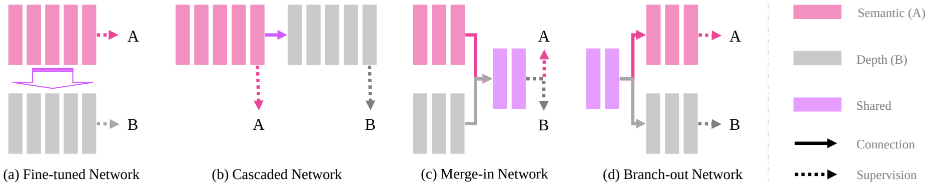


Fig. 1. Network attention visualization. Given an input RGB image, the spatial attention of the network is shown as an overlay to the input. Besides the attention for monocular depth estimation (second column), the attention maps for the shared backbone and the semantic labeling are also shown (last two columns).



(a) Fine-tuned Network (b) Cascaded Network (c) Merge-in Network (d) Branch-out Network

Fig. 2. Information sharing structures. (a) Fine-tuned and (b) cascaded networks share information sequentially, while (c) merge-in and (d) branch-out networks share simultaneously.

Fine-tuned Network: An intuitive solution to transfer information from one task to another is transfer learning [2] by fine-tuning. Suppose we want to transfer the knowledge from task-A to task-B, we need to first train a network on the data from A, and then fine-tune this network with the new data from B.

Cascaded Network: This kind of structure connects the networks of two tasks sequentially, *i.e.* information first passes task-A and then task-B. Here two losses from the corresponding tasks are used as the supervision for training the whole network, results in a multiple-loss training.

Merge-in Network: Two sub-networks are first pre-trained on each dataset, and then merged into several shared layers for the final prediction. Different from the cascaded network, both tasks in the merge-in network sharing information from the final common layers. Corresponding losses are used to supervise the training.

Branch-out Network: Similar to the merge-in network, information between two tasks can be shared in a branch-out fashion, by sharing information at the first common layers and dividing into two parallel branches afterwards. This branch-out network is supervised by multiple losses and backpropagates through both tasks.

We perform an empirical study on the above networks and show the relative performance compared to the baseline single-task networks (either depth or semantic) in Table 1. Each task consists of 5 convolutional layers as shown in Fig. 2 and trained in the same setup (*i.e.* batch size, learning rate, *etc.*) on NYU Depth v2 dataset with 4 semantic categories [3]. From the comparison we can see that, the fine-tuned network performs worse than task-specific ones, indicating knowledge cannot well sharing by directly fine-tuning in our problem. Cascaded network only benefits the first task, possibly because A shares both forward and backward information while B only shares the forward information from A. On the other hand, the merge-in and branch-out networks are more beneficial to multi-task learning, possibly due to the information is shared both forward and backward. We further explore the better performed branch-out architecture by exhaustively enumerating all the possible sharing solutions as in Table 2. The results indicate that best sharing strategy needs careful tuning.

The above results reveal that multi-task learning may not always guarantee a better performance compared to single-task ones, unless optimum architecture is tuned. Fur-

Table 1. Empirical comparison of different structures on depth prediction and semantic labeling. X/Y in cascaded network (Casc.) means depth/semantic acting as the first half part. Red color indicates performance decrease, while green one increase.

Task	Finet.	Casc.	Merge	Branch
Depth (RMSE)	-0.54	+0.03/-0.06	+0.09	+0.28
Semantic (mIoU)	-0.06	-0.03/+0.003	+0.007	+0.09

Table 2. Branch-out network splitting strategy comparison. SX-Y means layers X to Y are shared by both tasks. Best performance marked bold.

Task	S1	S1-2	S1-3	S1-4	S1-5
Depth (RMSE)	+0.275	+0.279	+0.273	**+0.283**	+0.277
Semantic (mIoU)	**+0.095**	+0.090	+0.091	+0.088	+0.082

thermore, the best performance in Table 2 may not the optimum, as the tuning of layers is constrained to integer-based splitting.

3 Network Architecture Details

The proposed synergy network mainly consists of three parts:

1) *Shared backbone encoder* utilizes the ResNet-50 [4] network with the final classification layers removed.
2) *Depth estimation branch* reconstructs the depth information in a coarse-to-fine manner with shared knowledge from the semantic branch and up-skip connections.
3) *Semantic labeling branch* shares knowledge to the depth estimation branch through the lateral sharing units and predicts the semantic labels simultaneously.

Table 3 shows the detailed configurations of the proposed network architecture when using the 40-category data for the semantic labeling task. Batch normalization and non-linear activation ReLU are added to the corresponding conv layers.

4 Analysis on Robustness to Depth Range

In Section 4.2 (Table 2) of the main paper, we perform a study on the robustness to the long-tailed depth data with different tail length (*i.e.* depth range). In this supplementary material, we further demonstrate the robustness of our method to depth range, by comparing with state-of-the-art methods across different depth ranges as in Table 4 - Table 6. In addition to the results shown in Table 2 in the main paper, we also include the result of our proposed method with 40-category semantic labels, as shown in the last rows of

Table 3. Detailed architecture of the proposed synergy network. Semantic labels mapped to the 40-category are used for the semantic branch.

Shared Backbone Encoder			
Input	Output	I/O channels	Scale
RGB	conv1	3/64	1/2
conv1	pool1	64/64	1/4
pool1	res2a	64/256	1/4
res2a	res2b	256/256	1/4
res2b	res2c	256/256	1/4
res2c	res3a	256/512	1/8
res3a	res3b	512/512	1/8
res3b	res3c	512/512	1/8
res3c	res3d	512/512	1/8
res3d	res4a	512/1024	1/16
res4a	res4b	1024/1024	1/16
res4b	res4c	1024/1024	1/16
res4c	res4d	1024/1024	1/16
res4d	res4e	1024/1024	1/16
res4e	res4f	1024/1024	1/16
res4f	res5a	1024/2048	1/32
res5a	res5b	2048/2048	1/32
res5b	res5c	2048/2048	1/32
res5c	conv2	2048/1024	1/32

Depth Estimation Branch			
Input	Output	I/O channels	Scale
conv2	upconv_dep1	1024/512	1/16
$(1+\varphi_{D1})$.upconv_dep1 $+\varphi_{S1}$.upconv_sem1	upconv_dep2	512/256	1/8
$(1+\varphi_{D2})$.upconv_dep2 $+\varphi_{S2}$.upconv_sem2	up_conv_dep3	256/128	1/4
$(1+\varphi_{D3})$.upconv_dep3 $+\varphi_{S3}$.upconv_sem3	upconv_dep4	128/64	1/2
$\hbar$(conv2)+$\hbar$(upconv_dep1)+$\hbar$(upconv_dep2) +$\hbar$(upconv_dep3)+upconv_dep4	conv_dep	64/1	1/2
conv_dep	out_dep	1/1	1

Semantic Labeling Branch			
Input	Output	I/O channels	Scale
conv2	upconv_sem1	1024/512	1/16
$(1+\gamma_{S1})$.upconv_sem1 $+\gamma_{D1}$.upconv_dep1	upconv_sem2	512/256	1/8
$(1+\gamma_{S2})$.upconv_sem2 $+\gamma_{D2}$.upconv_dep2	up_conv_sem3	256/128	1/4
$(1+\gamma_{S3})$.upconv_sem3 $+\gamma_{D3}$.upconv_dep3	upconv_sem4	128/64	1/2
$\hbar$(conv2)+$\hbar$(upconv_sem1)+$\hbar$(upconv_sem2) +$\hbar$(upconv_sem3)+upconv_sem4	conv_sem	64/40	1/2
conv_sem	out_sem	40/40	1

each table. From the tables, we can see that although our method is proposed to focus more on the distant depth regions, it still performs better than the compared methods on nearby regions, *i.e.* depth range $\leq$ 4m, 6m, 8m. When comparing among different depth ranges, the performances of the compared methods decrease as the depth range increases, while our method performs consistently in these depth ranges. This validates the effectiveness and robustness of our proposed method on long-tailed data. As complimentary to Table 2 in the main paper, we also present the depth-only version without semantics to show the robustness of our depth-aware objective, as in Table 7.

Table 4. Performance comparison on depth range $\leq$ 4m. Best and second best performances are shown in **bold** and *italic*, respectively.

Depth $\leq$4m	lower is better				higher is better		
	rel	log10	rms	rms (log)	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Liu *et al.* [5]	0.215	0.084	0.555	0.259	0.671	0.916	0.978
Eigen & Fergus [6]	0.160	0.066	0.464	0.212	0.777	0.953	0.988
Laina *et al.* [7]	0.118	0.050	0.416	0.187	0.820	0.956	0.989
Wang *et al.* [8]	0.225	0.087	0.566	0.264	0.649	0.909	0.978
Proposed-4c	*0.105*	*0.042*	*0.300*	*0.130*	*0.908*	*0.981*	**0.995**
Proposed-40c	**0.103**	**0.041**	**0.294**	**0.129**	**0.910**	**0.982**	**0.995**

Table 5. Performance comparison on depth range $\leq$ 6m. Best and second best performances are shown in **bold** and *italic*, respectively.

Depth $\leq$6m	lower is better				higher is better		
	rel	log10	rms	rms (log)	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Liu *et al.* [5]	0.212	0.086	0.647	0.261	0.660	0.914	0.978
Eigen & Fergus [6]	0.158	0.067	0.529	0.212	0.774	0.954	0.989
Laina *et al.* [7]	0.113	0.049	0.473	0.187	0.819	0.956	0.989
Wang *et al.* [8]	0.222	0.089	0.663	0.267	0.637	0.905	0.979
Proposed-4c	*0.101*	*0.041*	*0.326*	*0.127*	*0.915*	**0.983**	**0.996**
Proposed-40c	**0.099**	**0.040**	**0.311**	**0.126**	**0.916**	**0.983**	**0.996**

Table 6. Performance comparison on depth range $\leq$ 8m. Best and second best performances are shown in **bold** and *italic*, respectively.

Depth $\leq$8m	lower is better				higher is better		
	rel	log10	rms	rms (log)	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Liu *et al.* [5]	0.214	0.087	0.714	0.265	0.653	0.909	0.977
Eigen & Fergus [6]	0.160	0.068	0.591	0.215	0.766	0.951	0.988
Laina *et al.* [7]	0.113	0.049	0.519	0.190	0.814	0.955	0.989
Wang *et al.* [8]	0.224	0.091	0.748	0.273	0.627	0.898	0.977
Proposed-4c	*0.100*	**0.040**	*0.326*	*0.127*	*0.915*	**0.983**	**0.996**
Proposed-40c	**0.099**	**0.040**	**0.322**	**0.125**	**0.916**	**0.983**	**0.996**

Table 7. Analysis on robustness to data "tail", without semantic labeling task.

Depth range	lower is better				higher is better		
	rel	log10	rms	rms (log)	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
$\leq$ 4m	0.131	0.051	0.367	0.157	0.861	0.970	0.992
$\leq$ 6m	0.126	0.050	0.389	0.154	0.870	0.973	0.993
$\leq$ 8m	0.126	0.050	0.404	0.154	0.870	0.973	0.993
All	0.126	0.050	0.416	0.154	0.868	0.973	0.993

5 Evaluation on Attention-Driven Loss.

In addition to the architecture analysis in the main paper, we further perform an ablation study on the proposed attention-driven loss function. To evaluate the proposed depth-aware loss L_{DA}, we train an encoder-decoder network structure [9] and compare the L_2 loss with our L_{DA}. A state-of-the-art structure [7] is also trained on our dataset and we substitute their loss by L_{DA} to evaluate our loss. The result is shown in Table 8. From the result we can see, the proposed L_{DA} loss performs better than commonly used L_2 loss and also improves the state-of-the-art method [7]. To better understand the effectiveness of each component in our combined loss, we show the results of our network trained with different loss components (with 40-c semantic labels), as in the lower part of Table 8. The results indicate that the depth-aware loss contributes the most, with the other two components giving secondary contributions.

Table 8. Evaluation on attention-driven loss. Upper part shows the effectiveness of our L_{DA} when applied to different structures. Lower part shows the ablation study on the loss components.

Model	lower is better				higher is better		
	rel	log10	rms	rms (log)	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
EncdDecd [9] (L_2)	0.256	0.102	0.852	0.292	0.581	0.864	0.958
EncdDecd [9] (L_{DA})	0.188	0.076	0.630	0.224	0.731	0.928	0.980
Laina [7]	0.173	0.070	0.653	0.208	0.767	0.945	0.985
Laina [7] (L_{DA})	0.145	0.062	0.532	0.189	0.801	0.952	0.987
w/o $L_{attention}$	0.136	0.053	0.439	0.164	0.853	0.969	0.992
+L_{DA}	0.101	0.041	0.334	0.128	0.913	0.982	0.996
+L_{DA}+L_{semF}	0.099	0.040	0.330	0.126	0.916	0.983	0.996
+L_{DA}+L_{semF}+L_{JG}	0.098	0.040	0.329	0.125	0.917	0.983	0.996

6 More Qualitative Results of Depth Estimation

In addition to the results shown in the main paper for monocular depth estimation, we show more qualitative results with comparison to state-of-the-art methods in Fig. 3 - Fig. 4. For better visual comparison, as in the main paper, all the depth maps are in the same depth range with the ground truth depths, *i.e.* same color indicates the same absolute depth value. From these additional results, we can see that the depth maps estimated by our proposed method are more accurate than those by the compared methods. Due to the proposed attention-driven loss and the depth-aware objective, our method performs much better than others in the distant depth regions.

To better evaluate the depth quality, we further project the 2D depth into 3D and generate several point cloud examples shown in Fig. 5. The RGB color images are also mapped to the corresponding point clouds for better visualization. From the projected point clouds, we can see that our method produces more accurate 3D structure compared to other methods, *e.g.* the distant regions in all these projections, the door and cabinet structures in the first and second projections, and the ceiling and lamp in the last projection.

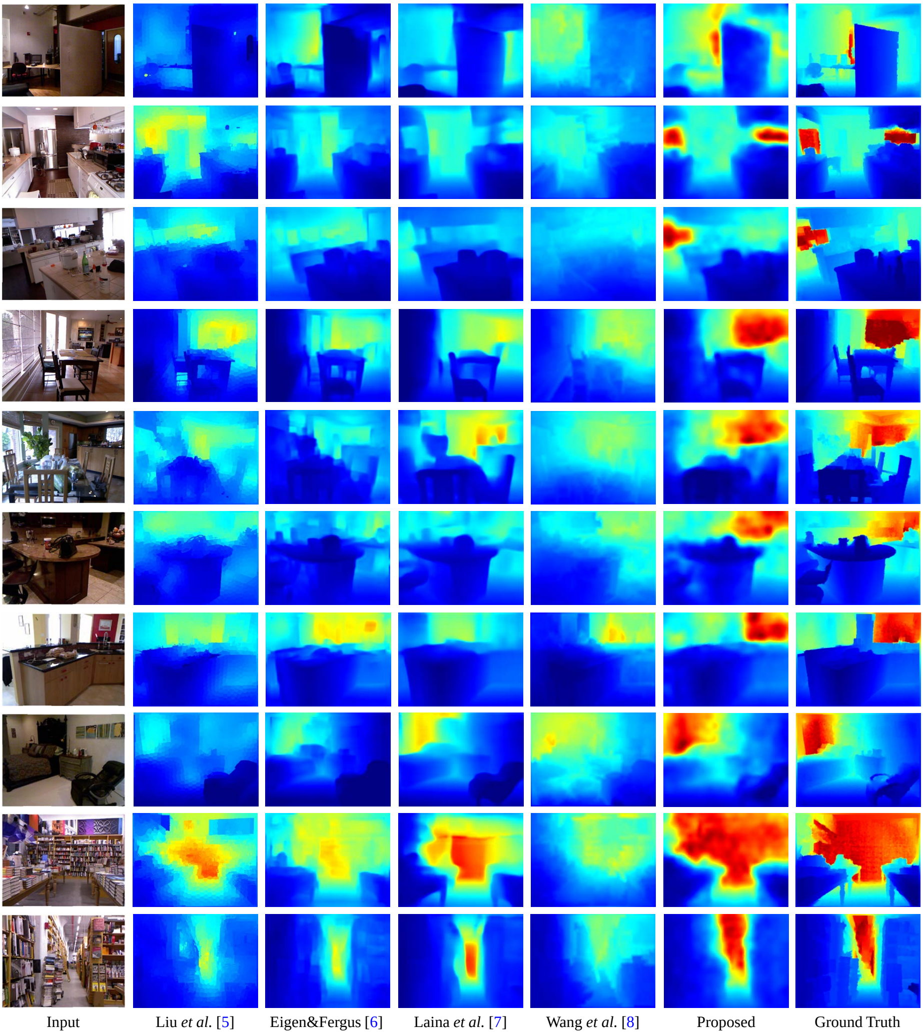


Fig. 3. More qualitative results on monocular depth estimation. All the depth maps are in the same depth range as the ground truth. Red region indicates large depth while blue indicates small.

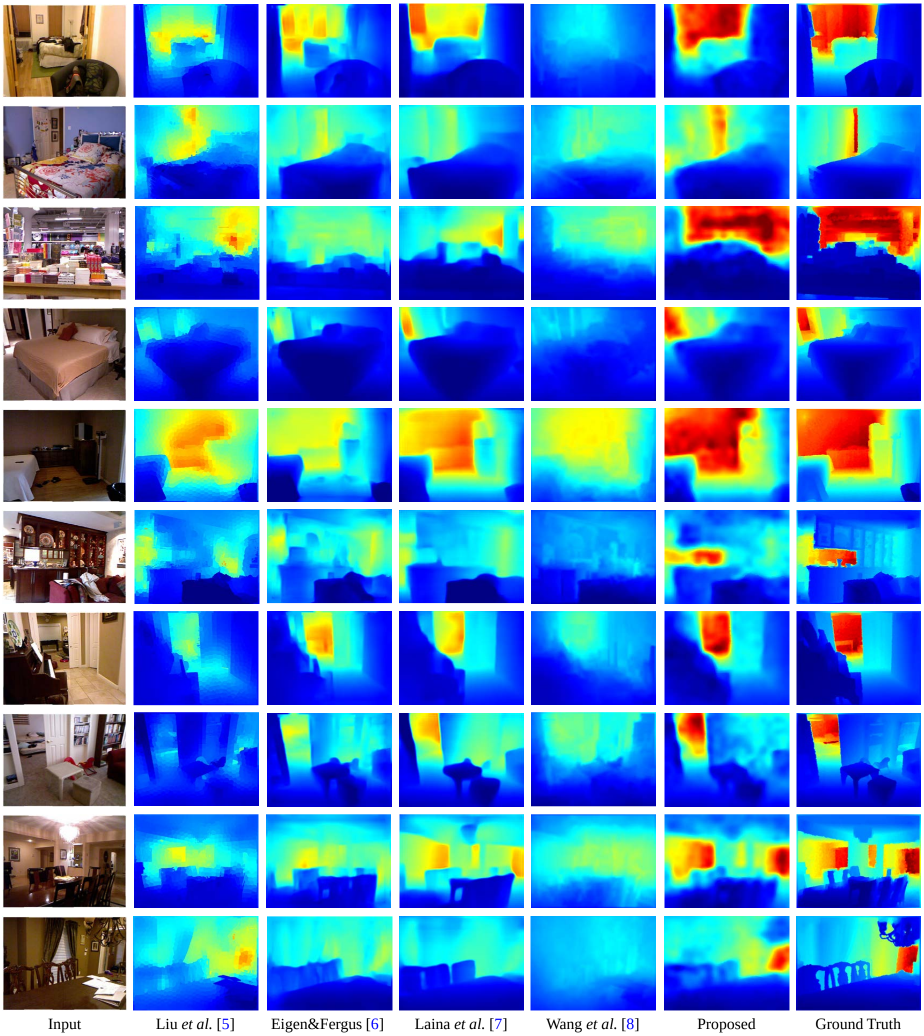


Fig. 4. More qualitative results on monocular depth estimation. All the depth maps are in the same depth range as the ground truth. Red region indicates large depth while blue indicates small.

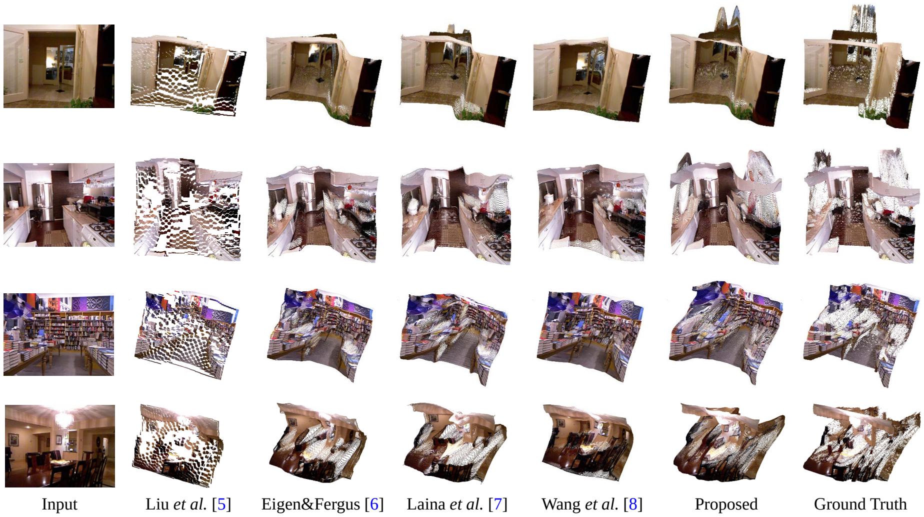


Fig. 5. Qualitative results of 3D point cloud reconstructed from the depth map. All the point clouds in each example are with the same camera settings.

7 Qualitative Results of Semantic Labeling

As mentioned in the main paper, besides the depth estimation, our model is able to predict the semantic labels of the scene at the same time. In Fig. 6, we show some qualitative results on the semantic labeling with 40-category on the NYUD2 dataset. Although the semantic labels are leveraged to boost the monocular depth estimation, the results in Fig. 6 validates the effectiveness of our proposed model on semantic labeling and the knowledge sharing strategy.

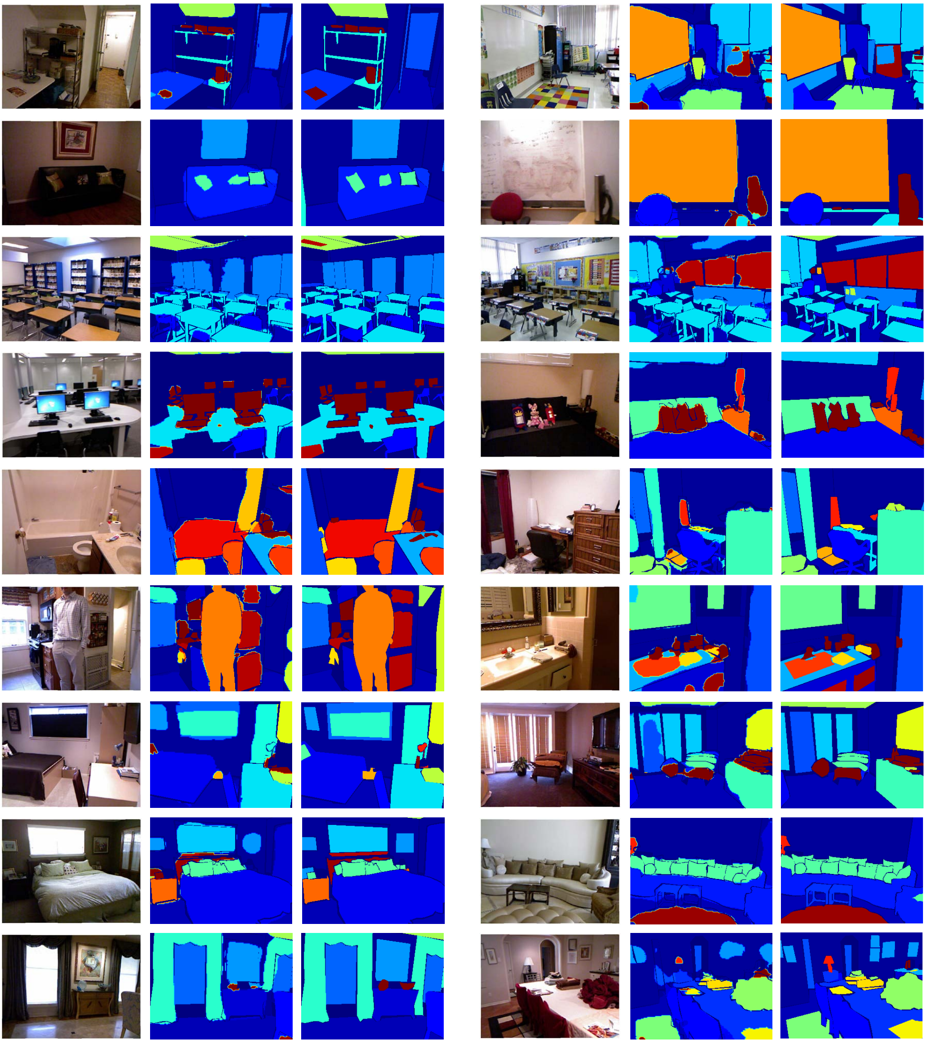


Fig. 6. Qualitative results on semantic labeling with 40-category on NYUD2. For each example from left to right are: input RGB image, our result, and ground truth.

8 Outdoor Scene Training Details

As presented in Section 4.2 in the main paper, we extend our model to outdoor scenes to illustrate the generalization ability. Specifically, our model is trained on the KITTI [10] and Cityscapes [11] datasets. For KITTI, we use the official training set of 200 images, which covers 28 scenes. Although stereo image pairs are provided, we only take the left color images with corresponding sparse disparity maps as our training set. The disparity value is converted to depth with the same method as in [12]. Since no corresponding semantic labels are available, we leverage the semantic labels from the Cityscapes dataset due to similar scene appearances between KITTI and Cityscapes (both are outdoor street scenes). We use the suggested 21 categories (train IDs) of semantic labels and randomly select 200 images with fine annotations, from the Cityscapes dataset. As a result, our training set consists of 400 pairs in total. As for each training sample only one direction (either depth or semantic labels) is available each time, fully-supervised learning is not applicable here. As a result, we perform the training in a weakly-supervised manner: whenever a training pair is fed into the network, if it is an “RGB-depth” pair, the backpropagation of the semantic branch is frozen; and accordingly the backpropagation of depth branch is frozen if an “RGB-semantic” pair comes. The training is based on our best-performed 40-category model on NYUD2, with the final layer of semantic branch modified to meet the new category number (*i.e.* 40 to 21). The learning rate is initialized as 10^{-2} for the final semantic layer and 10^{-4} for the rest of the model. All the images are downscaled to 512×256 and resized to the original size at the end. Other training configurations are the same as the model on NYUD2. We test the generalization ability of our model on the Cityscapes test images. Please refer to the sample video clip (outdoor_scene.mp4) accompanied with this supplementary material for the qualitative performance. Note that this study is only performed to show the generalization of our model and the model is trained on a small dataset. We believe more training data and better weakly-supervised training strategy would lead to a much better performance, which will be left as our future work.

References

1. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In: ICLR. (2017) 1
2. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on knowledge and data engineering **22**(10) (2010) 1345–1359 3
3. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: ECCV. (2012) 3
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. (2016) 4
5. Liu, F., Shen, C., Lin, G.: Deep convolutional neural fields for depth estimation from a single image. In: CVPR. (2015) 6
6. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: ICCV. (2015) 6
7. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: Deeper depth prediction with fully convolutional residual networks. In: 3D Vision (3DV). (2016) 6, 7
8. Wang, P., Shen, X., Lin, Z., Cohen, S., Price, B., Yuille, A.L.: Towards unified depth and semantic prediction from a single image. In: CVPR. (2015) 6
9. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE TPAMI **39**(12) (2017) 2481–2495 7
10. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: CVPR. (2012) 13
11. 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) 13
12. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: CVPR. (2017) 13