

## Supplementary Materials for “Uncertainty Quantification in Depth Estimation via Constrained Ordinal Regression”

### 1 Algorithms of Re-sampling Methods

In the section, we provide our detailed procedure of using Wild Bootstrap and Multiplier Bootstrap for estimation variance approximation.

---

#### **Algorithm 1** Training Procedure of Wild Bootstrap

---

```

1: Train a single network  $\hat{\Phi} \leftarrow \operatorname{argmin}_{\Phi} \sum_{i=1}^n \sum_{w=1}^W \sum_{h=1}^H \ell(x_i, y_i^{(w,h)}, \Phi)$ 
2: for  $(x_i, y_i^{(w,h)}) \in \mathcal{D}$  do
3:    $\hat{y}_i^{(w,h)} \leftarrow \mathbb{E}[Y_i^{(w,h)} | x_i; \hat{\Phi}]$                                  $\triangleright$  fitted value
4:    $\hat{\epsilon}_i^{(w,h)} \leftarrow y_i^{(w,h)} - \hat{y}_i^{(w,h)}$                                  $\triangleright$  residual
5: end for
6: Initialize  $\Phi_1, \Phi_2, \dots, \Phi_M$  with the parameters of  $\hat{\Phi}$ 
7: for  $m = 1 : M$  do
8:   for  $(x_i, y_i^{(w,h)}) \in \mathcal{D}$  do
9:     Sample  $\tau_{m,i}^{(w,h)} \sim \mathcal{N}(0, 1)$ 
10:     $v_{m,i}^{(w,h)} \leftarrow \hat{y}_i^{(w,h)} + \hat{\epsilon}_i^{(w,h)} \cdot \tau_{m,i}^{(w,h)}$             $\triangleright$  re-sampled response
11:   end for
12:   Train  $\hat{\Phi}_m \leftarrow \operatorname{argmin}_{\Phi} \sum_{i=1}^n \sum_{w=1}^W \sum_{h=1}^H \ell(x_i, v_{m,i}^{(w,h)}, \Phi)$ 
13: end for

```

---



---

#### **Algorithm 2** Training Procedure of Multiplier Bootstrap

---

```

1: Train a single network  $\hat{\Phi} \leftarrow \operatorname{argmin}_{\Phi} \sum_{i=1}^n \sum_{w=1}^W \sum_{h=1}^H \ell(x_i, y_i^{(w,h)}, \Phi)$ 
2: Initialize  $\Phi_1, \Phi_2, \dots, \Phi_M$  with the parameters of  $\hat{\Phi}$ 
3: for  $m = 1 : M$  do
4:   for  $(x_i, y_i^{(w,h)}) \in \mathcal{D}$  do
5:     Sample  $\omega_{m,i}^{(w,h)} \sim \mathcal{N}(1, 1)$                                  $\triangleright$  re-sampled weight
6:   end for
7:   Train  $\hat{\Phi}_m \leftarrow \operatorname{argmin}_{\Phi} \sum_{i=1}^n \sum_{w=1}^W \sum_{h=1}^H \omega_{m,i}^{(w,h)} \ell(x_i, y_i^{(w,h)}, \Phi)$ 
8: end for

```

---

## 2 Experiments of Toy Datasets

To examine the effectiveness of our approach, we conduct the experiment on one-dimensional regression toy datasets. In this section, we first start with the experiment design and setting, then introduce the evaluation approaches, finally discuss the qualitative and quantitative results.

### 2.1 Setup and Baselines

To make the mean function to be a skewed function and cover a decent range of response values, we adopt the shape of Gamma probability density function. Given an input feature  $x_i$ , the sample of response is drawn from a non-linear generator:

$$y_i = 10 + \frac{25}{48}x_i^3 e^{-0.5x_i} + \sqrt{V(x_i)}\epsilon_i, \quad (1)$$

where  $\epsilon_i$  draws from a standard Gaussian distribution,  $V(x)$  is the variance function designed as a non-linear function of  $x$ :

$$V(x) = 0.5 + 5e^{-0.1(x-5)^2}. \quad (2)$$

Using two differently simulated datasets, we can showcase the effect of the different methods to estimate *error variance*  $V(x_i)$  and *estimation variance*  $\text{Var}[\hat{g}(x_i)]$ :

- Dataset I: sample  $x \sim \text{Uniform}(0, 10)$ , by which we focus on the *error variance* as the estimation variance is negligible given sufficiently large sample size (Fig. 1a).
- Dataset II: sample  $x \sim P(x) = \sum_{i=1}^2 \phi_i \mathcal{N}(\mu_i, \sigma_i^2)$  (mixture Gaussian), for  $0 < x < 10$ , where  $\phi_1 = 0.25, \phi_2 = 0.75, \mu_1 = 2, \mu_2 = 7, \sigma_1^2 = 0.6, \sigma_2^2 = 1$ , by which we focus on the *estimation variance* as the mixture Gaussian results in a wide range of estimation variance for different  $x$  (Fig. 2a). The estimation variance is relatively high when  $x$  is around 4.

Each training set contains 2000 samples. We identically built our model with 8 hidden layers, each with 100 neurons. The exponential linear unit (ELU) activation function is used at each hidden layer and the softmax activation function is used at the output layer. For the bootstrap,  $M$  is set to 100, and the re-sampling noise is drawn from Gaussian distribution.

For comparison, we implement Gaussian Likelihood (GL) and Log Gaussian Likelihood (LGL) for estimating the *error variance*. We additionally re-implement Binary Classification [9] (BC) with the same SID to illustrate the effectiveness of using classification to capture the uncertainty of a regression problem. We adopt the identical architecture for all the models, with only the substitution of the network head. We apply Monte Carlo Dropout (MCD) [4] and Deep Ensemble (DE) [7] for approximating *estimation variance*.

## 2.2 Evaluation

We measure the accuracy of *error variance* estimation against the ground truth value on Dataset I. The ground truth value of error variance can be obtained using Equation (2). Hence, the ground truth entropy is derived using differential entropy [2]. Regarding the evaluation of *estimation variance* approximation on Dataset II, we train 100 models with 100 sampled i.i.d datasets and obtain the ground truth value of estimation variance. We measure the accuracy of estimation by mean absolute error (mae) and absolute relative error (rel), and report the result by averaging the value from 10 experiments with different seeds.

## 2.3 Qualitative Results

We visualize the estimation performance at each position of  $x$  in the range  $[0, 10]$ . Regarding the error variance, it appears that ConOR learns a good conditional variance function compared to GL and LGL (Fig. 1d), but with a more stable and faster training (Fig. 1e). For comparison, we plot the mean estimated by different models on Dataset I in Fig 1b. We then visualize the Shannon entropy estimated from BC [9], and compared with the ground-truth differential entropy [2]. As shown in Fig. 1c, it appears there is a gap between inferred entropy and the uncertainty of the original regression problem.

Fig. 2 demonstrate the performance of different approaches to approximate variances under various models. It appears that bootstrapping can produce more accurate estimation variance approximations.

Furthermore, from the frequentist view, we visualize the 95% confidence interval (CI) and prediction interval (PI) of our proposed methods in Fig 3.

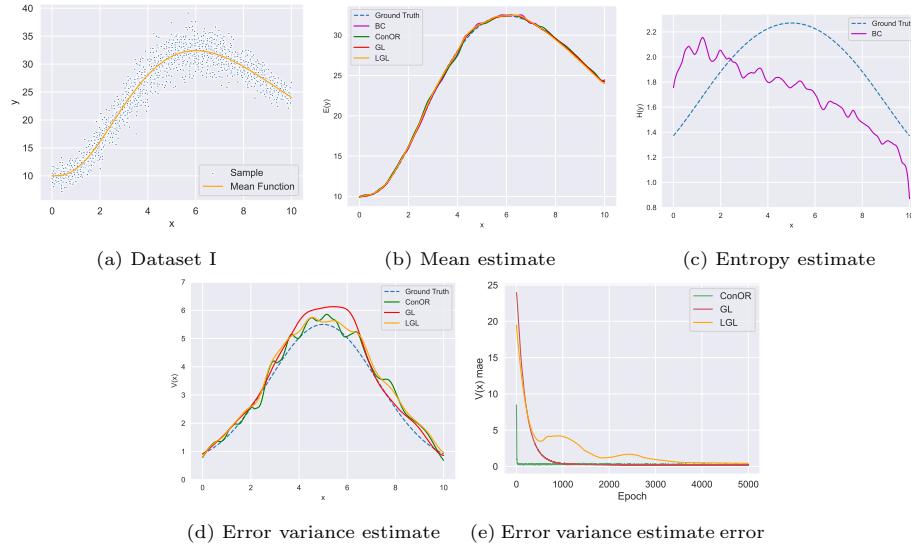


Fig. 1: We sample  $x$  from Uniform distribution to generate Dataset I (1a). We plot the estimated *error variance*  $\hat{V}(x)$  against the ground truth (dashed line) in 1d, together with the estimation error during the training process (1e)

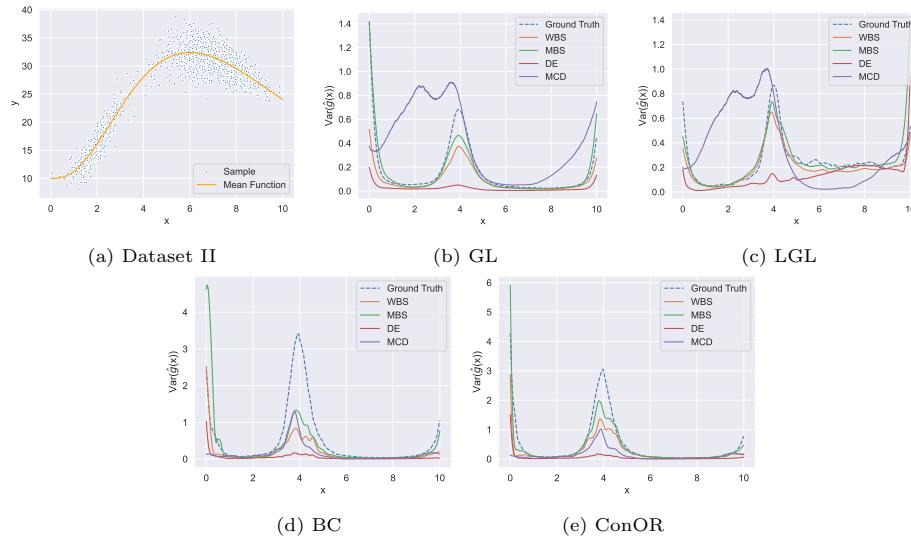


Fig. 2: We sample  $x$  from a mixture Gaussian distribution to generate Dataset II (2a). We demonstrate the performance of different approaches approximating *estimation variance* on GL, LGL, BC and ConOR in 2b, 2c, 2d and 2e respectively

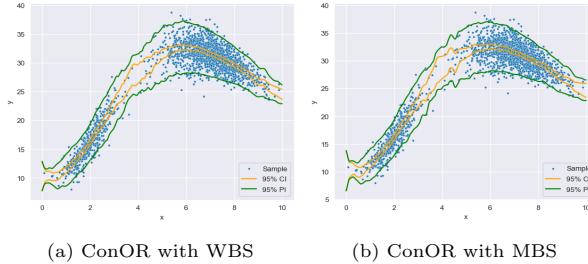


Fig. 3: Visualization of the 95% CI and PI inferred by our methods on the Dataset II

## 2.4 Quantitative Results

In the section, we provide the quantitative experimental results of the toy datasets. Table 1 reports the performance of mean and error variance estimation on Dataset I. The result of approximating estimation variance on Dataset II under different models is shown in Table 2.

Table 1: Comparison of mean estimation and error variance estimation on Dataset I

Method	Mean		Error Variance	
	mae	rel	mae	rel
GL	<b>0.048</b>	<b>0.002</b>	<b>0.297</b>	<b>0.092</b>
LGL	0.095	0.004	<b>0.297</b>	0.112
ConOR	0.122	0.005	<b>0.297</b>	0.112

Table 2: Comparison of estimation variance approximation on Dataset II

Method \ Model	GL		LGL		BC		ConOR	
	mae	rel	mae	rel	mae	rel	mae	rel
MCD [4]	0.278	3.93	0.277	2.64	0.339	0.661	0.198	0.741
DE [7]	0.131	0.801	0.120	0.433	0.451	0.865	0.405	0.851
WBS	0.062	0.299	<b>0.046</b>	<b>0.166</b>	0.319	0.562	0.204	0.213
MBS	<b>0.039</b>	<b>0.266</b>	0.050	0.189	<b>0.314</b>	<b>0.455</b>	<b>0.126</b>	<b>0.164</b>

### 3 More Experimental Results of Benchmark

We report more detailed experimental results of Benchmark dataset KITTI [5] and NYUv2 [8] here.

#### 3.1 Depth Estimation Evaluation

In this section, we report the complete evaluation of depth prediction on the metrics traditionally adopted in this field, in Table 3 and Table 4 respectively.

Table 3: Depth estimation evaluation on KITTI

Method	rmse↓	rmse log↓	absrml↓	sqrel↓	log10↓	silog↓	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$
MCC [1,3]	3.011	0.138	0.081	0.368	0.036	13.28	0.915	0.979	0.993
BC [9]	2.878	0.128	0.078	0.341	0.034	12.31	0.919	0.981	0.994
GL+MCD [4]	3.337	0.166	0.102	0.452	0.057	14.28	0.875	0.967	0.990
GL+DE [7]	2.900	0.162	0.089	0.407	0.057	12.69	0.908	0.982	0.995
GL+WBS	3.064	0.164	0.083	0.434	0.056	12.86	0.906	0.980	0.994
GL+MBS	3.064	0.164	0.083	0.434	0.056	12.86	0.906	0.980	0.994
LGL+MCD [4]	3.219	0.141	0.158	0.386	0.040	14.45	0.836	0.962	0.990
LGL+DE [7]	2.852	0.140	0.132	0.383	0.040	12.95	0.873	0.981	0.995
LGL+WBS	2.965	0.140	0.132	0.384	0.041	13.25	0.870	0.978	0.995
LGL+MBS	2.965	0.140	0.132	0.384	0.041	13.25	0.870	0.978	0.995
ConOR+WBS	<b>2.709</b>	<b>0.122</b>	<b>0.075</b>	<b>0.312</b>	<b>0.033</b>	<b>11.47</b>	<b>0.928</b>	<b>0.986</b>	<b>0.996</b>
ConOR+MBS	<b>2.709</b>	<b>0.122</b>	<b>0.075</b>	<b>0.312</b>	<b>0.033</b>	<b>11.47</b>	<b>0.928</b>	<b>0.986</b>	<b>0.996</b>

Table 4: Depth estimation evaluation on NYUv2

Method	rmse↓	rmse log↓	absrml↓	sqrel↓	log10↓	silog↓	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$
MCC [1,3]	3.658	1.849	1.518	6.411	0.708	135.7	0.017	0.042	0.078
BC [9]	0.519	0.184	0.141	0.106	0.062	15.75	0.815	0.955	0.988
GL+MCD [4]	0.533	0.184	0.168	0.108	0.062	16.33	0.770	0.944	0.987
GL+DE [7]	0.503	0.188	0.158	0.106	0.063	15.23	0.790	0.953	0.990
GL+WBS	0.534	0.199	0.171	0.120	0.067	15.68	0.770	0.945	0.986
GL+MBS	0.534	0.199	0.171	0.120	0.067	15.68	0.770	0.945	0.986
LGL+MCD [4]	0.773	0.266	0.222	0.250	0.095	21.10	0.618	0.895	0.974
LGL+DE [7]	0.746	0.256	0.216	0.195	0.092	20.58	0.621	0.900	0.976
LGL+WBS	0.756	0.264	0.221	0.204	0.095	21.03	0.618	0.895	0.974
LGL+MBS	0.756	0.264	0.221	0.204	0.095	21.03	0.618	0.895	0.974
ConOR+WBS	<b>0.490</b>	<b>0.175</b>	<b>0.132</b>	<b>0.099</b>	<b>0.058</b>	<b>14.61</b>	<b>0.832</b>	<b>0.963</b>	<b>0.991</b>
ConOR+MBS	<b>0.490</b>	<b>0.175</b>	<b>0.132</b>	<b>0.099</b>	<b>0.058</b>	<b>14.61</b>	<b>0.832</b>	<b>0.963</b>	<b>0.991</b>

### 3.2 Sparsification Plots

To give some intuition about the estimated sparsification and oracle sparsification, we provide the sparsification curve [6] of some methods with three error metrics  $\xi$ , as shown in Fig. 4 and Fig. 5. We then plot the sparsification error [6] of different methods in Fig. 6. All the value is averaged over the test set. Our proposed methods behave similarly regarding sparsification error (Fig. 6) due to low estimation variance. There would be some subtle difference between the curve of “ConOR+WBS” and “ConOR+MBS” in Fig. 6 if zooming in.

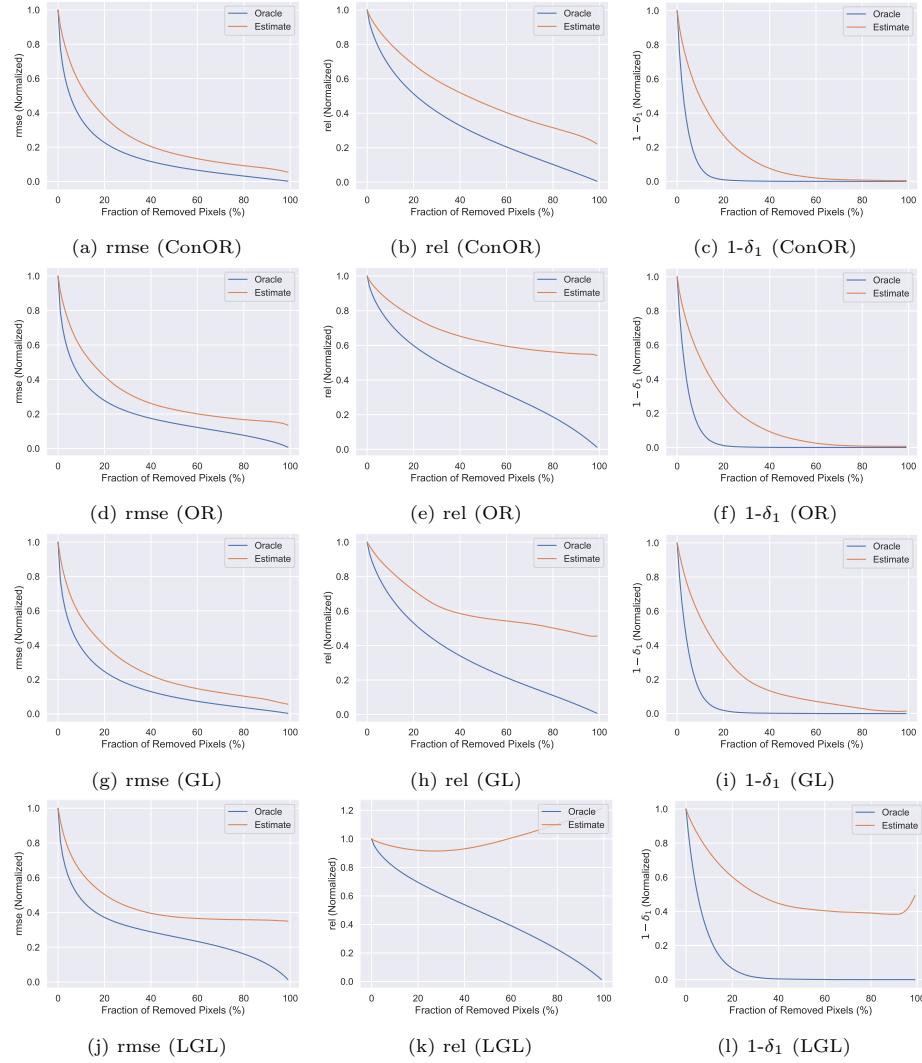


Fig. 4: Sparsification Curve of ConOR, OR, GL, LGL on KITTI

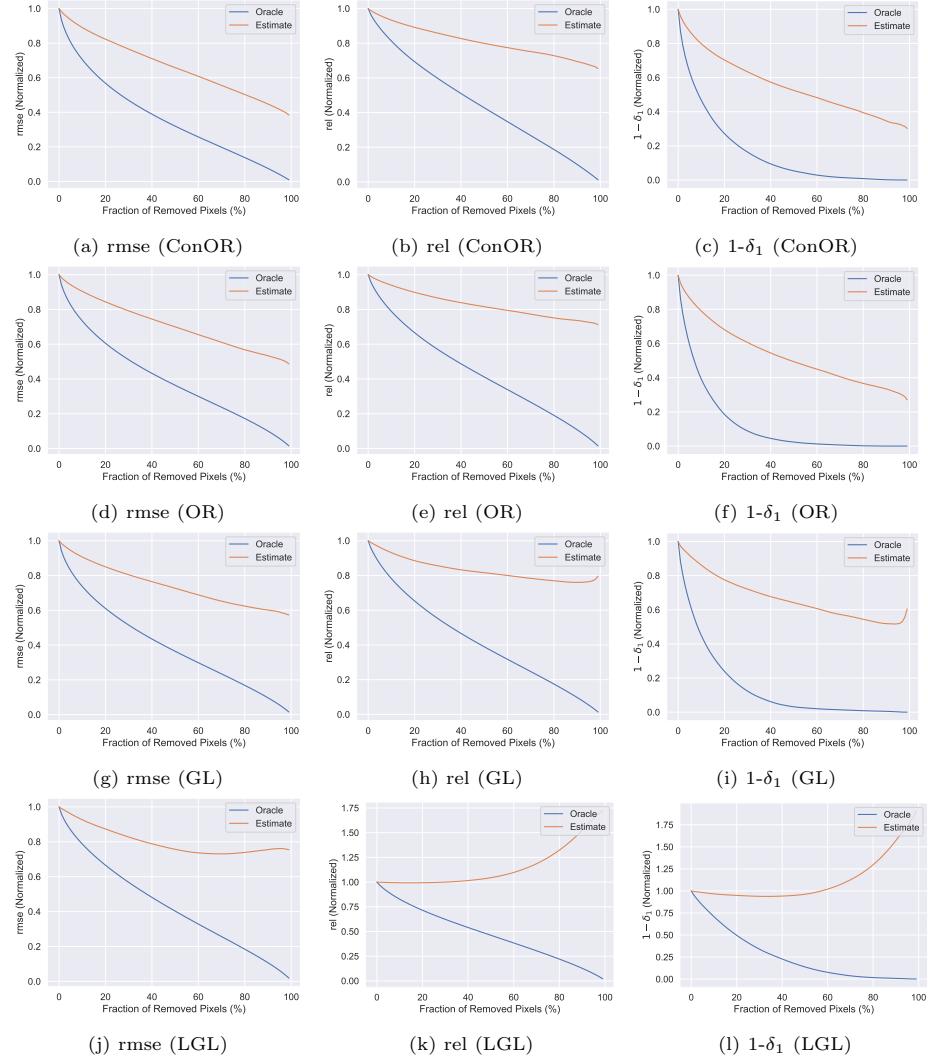


Fig. 5: Sparsification Curve of ConOR, OR, GL, LGL on NYUv2

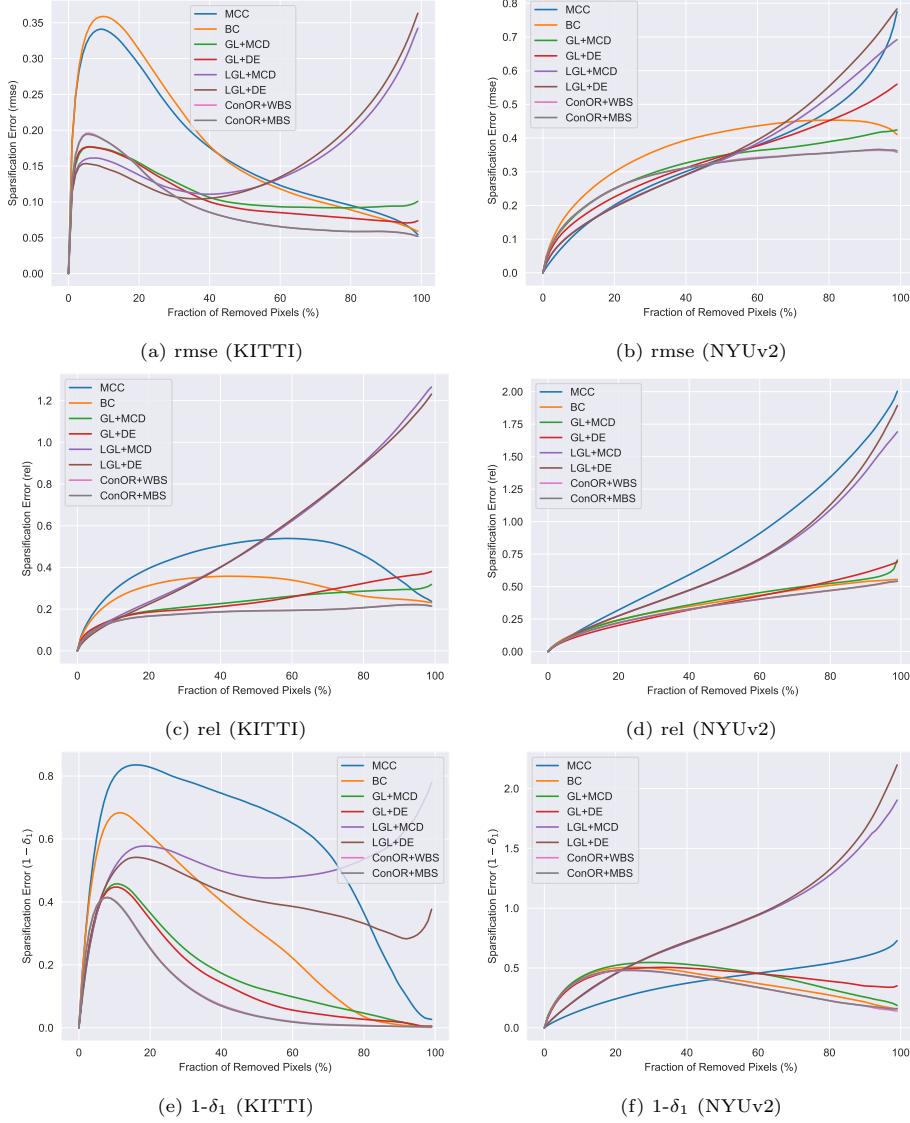


Fig. 6: Plots of Sparsification Error on KITTI and NYUv2. From top to bottom,  $\xi$  is adopted as rmse, rel and  $1 - \delta_1$ . AUSE is computed as the area under the curve, smaller is better

### 3.3 Qualitative Results

We demonstrate the inference results of ConOR+MBS on KITTI [5] (Fig 7, 8,9,10,11,12) and ConOR+WBS on NYUv2 [8] (Fig 13,14,15,16,17,18,19,20,21). We include the plots of depth prediction error for the reference of estimated uncertainty quality.

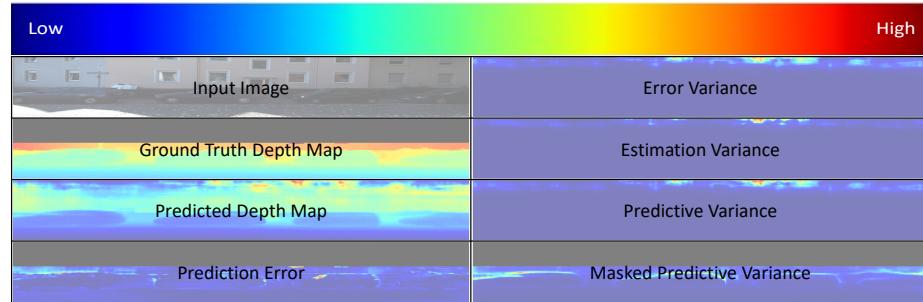


Fig. 7: Demonstration of layout of KITTI qualitative results. Navy blue and crimson indicate lower and higher values respectively, the black parts do not have ground truth depth value

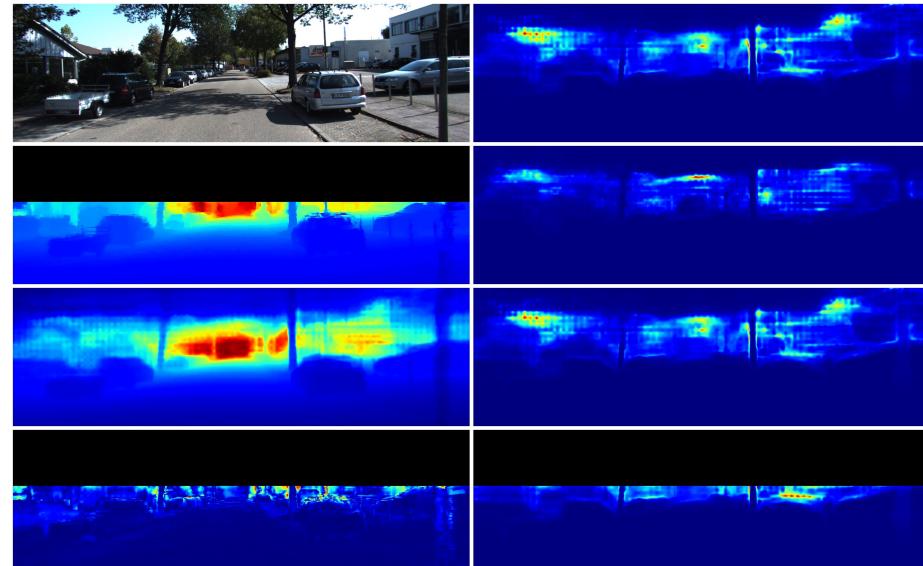


Fig. 8: Visualization results of proposed method on KITTI

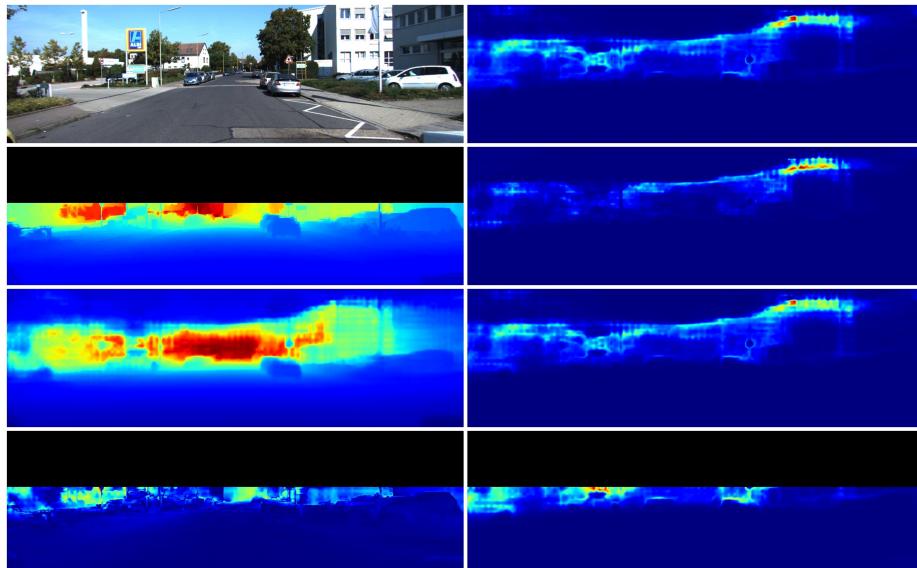


Fig. 9: Visualization of proposed method on KITTI

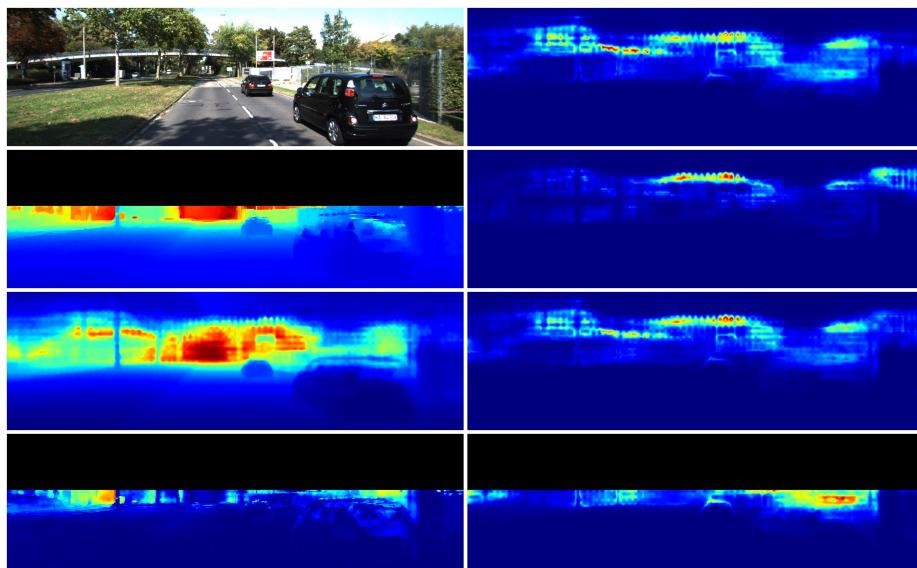


Fig. 10: Visualization results of proposed method on KITTI

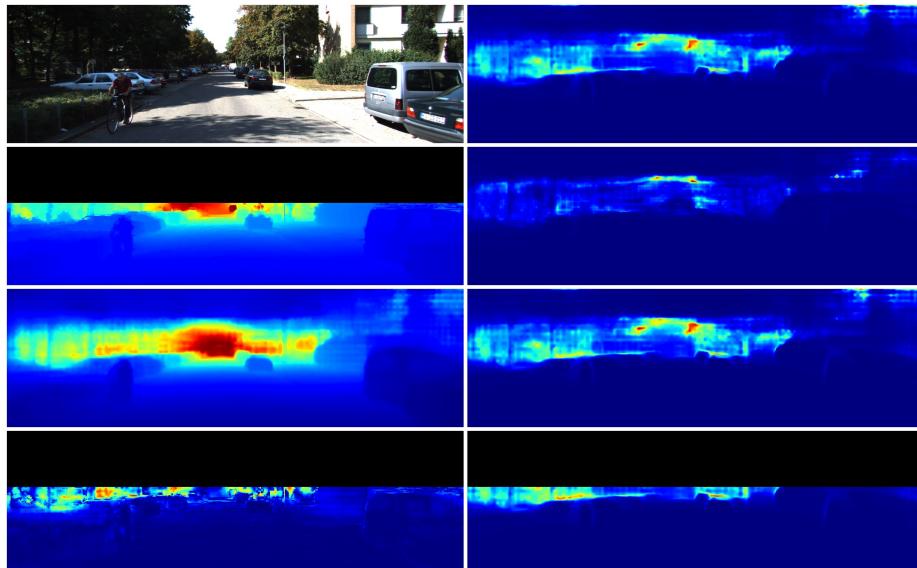


Fig. 11: Visualization results of proposed method on KITTI

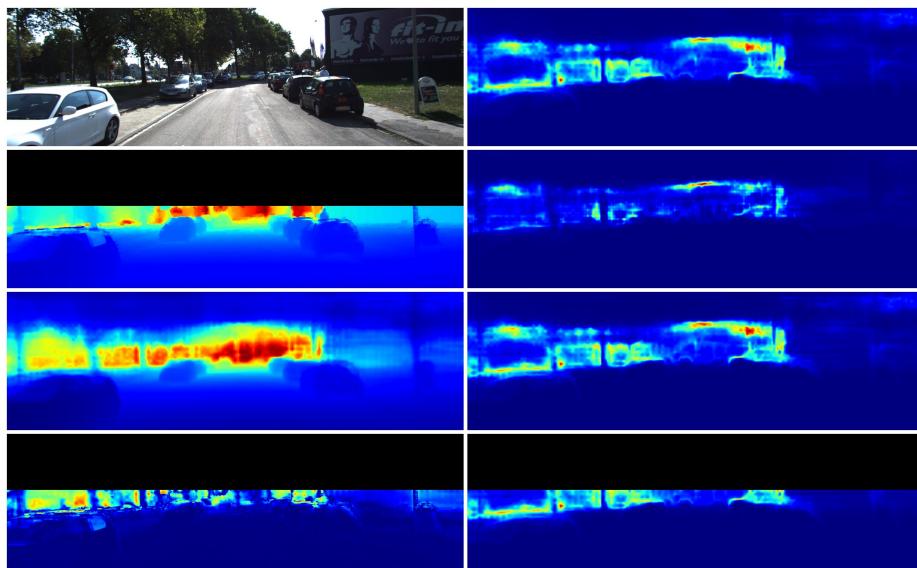


Fig. 12: Visualization results of proposed method on KITTI



Fig. 13: Demonstration of layout of NYUv2 qualitative results. Navy blue and crimson indicate lower and higher values respectively

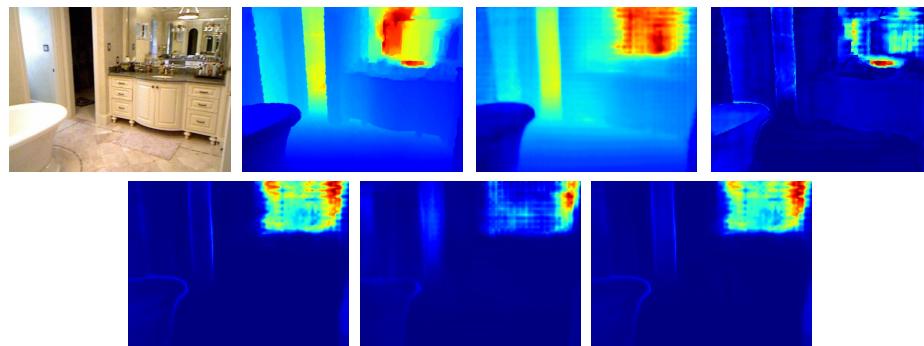


Fig. 14: Visualization of proposed method on NYUv2

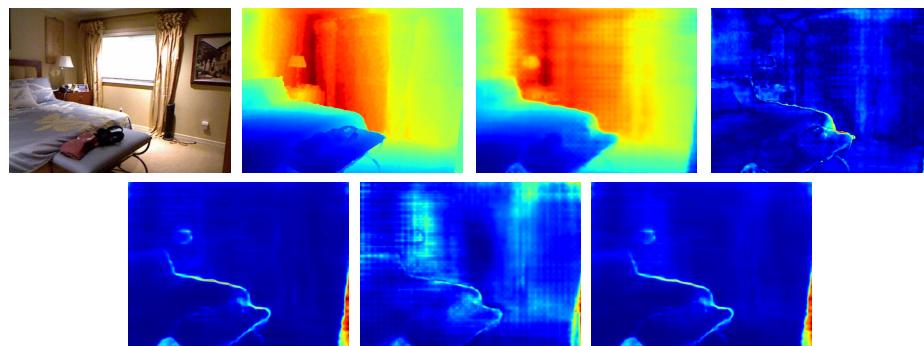


Fig. 15: Visualization of proposed method on NYUv2

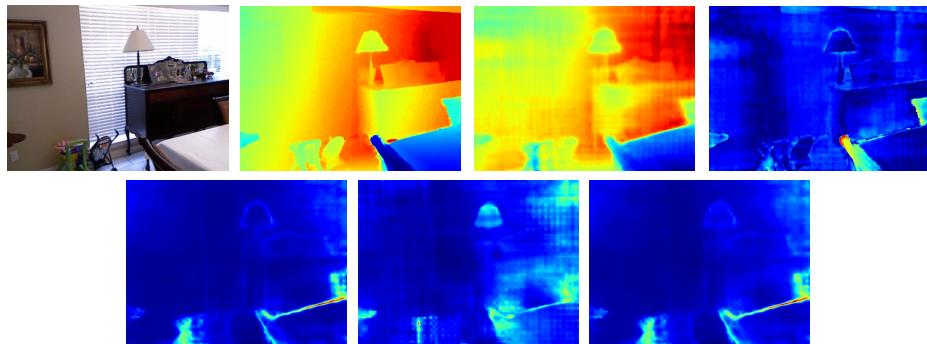


Fig. 16: Visualization of proposed method on NYUv2

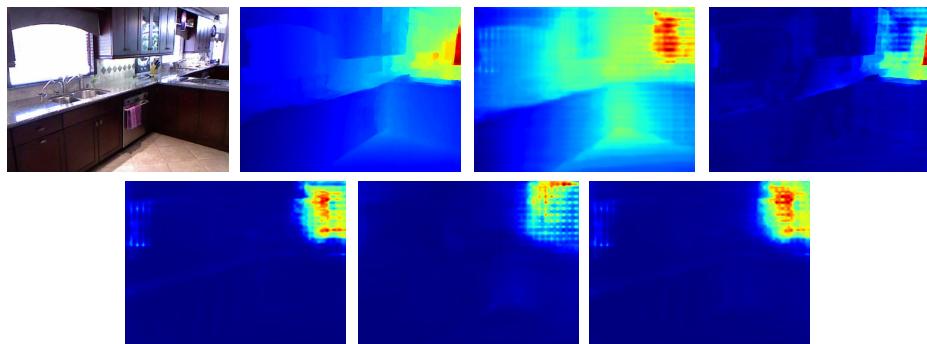


Fig. 17: Visualization of proposed method on NYUv2

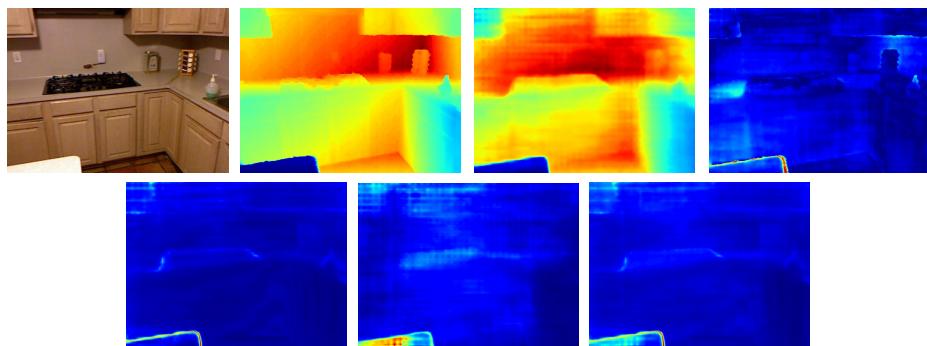


Fig. 18: Visualization of proposed method on NYUv2

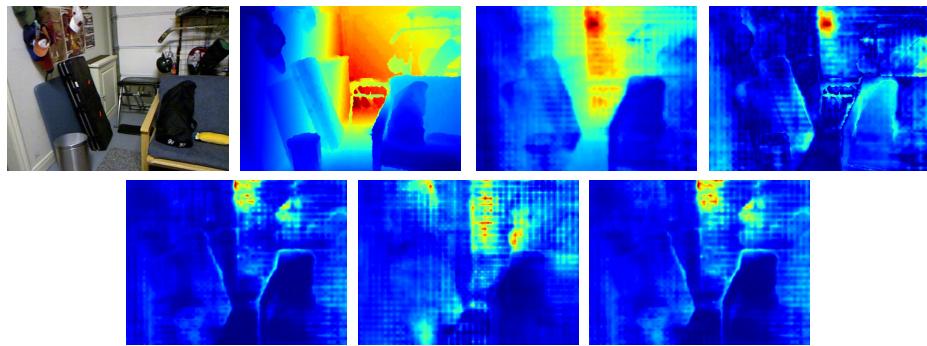


Fig. 19: Visualization of proposed method on NYUv2

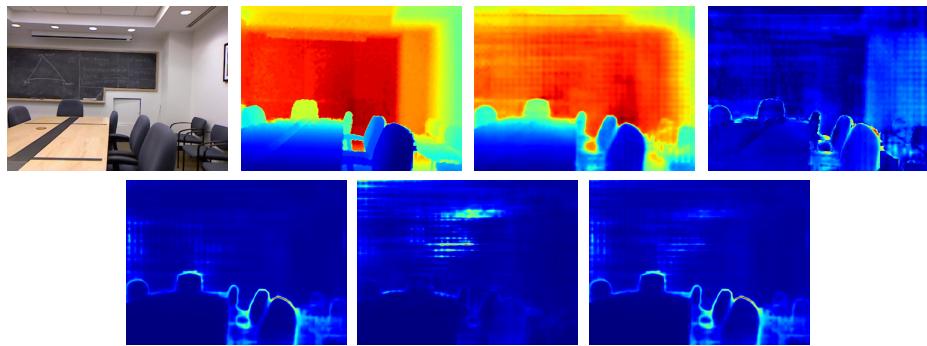


Fig. 20: Visualization of proposed method on NYUv2

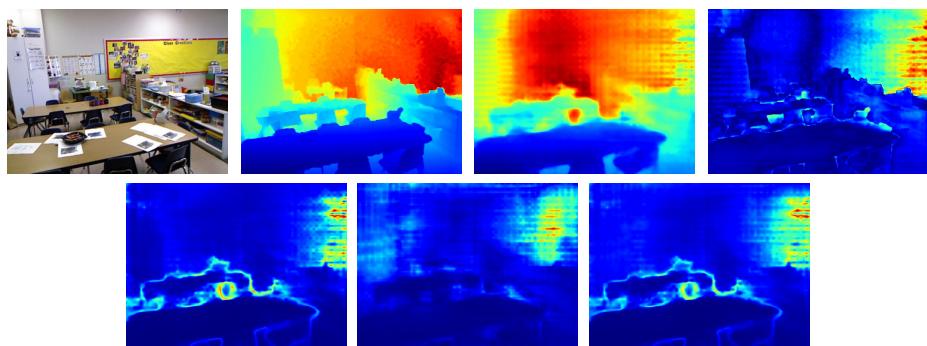


Fig. 21: Visualization of proposed method on NYUv2

## References

1. Cao, Y., Wu, Z., Shen, C.: Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Transactions on Circuits and Systems for Video Technology* **28**(11), 3174–3182 (2017)
2. Cover, T.M., Thomas, J.A.: Elements of information theory second edition solutions to problems. Internet Access pp. 19–20 (2006)
3. Fu, H., Gong, M., Wang, C., Tao, D.: A compromise principle in deep monocular depth estimation. arXiv preprint arXiv:1708.08267 (2017)
4. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: international conference on machine learning. pp. 1050–1059. PMLR (2016)
5. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research* **32**(11), 1231–1237 (2013)
6. Ilg, E., Cicek, O., Galessio, S., Klein, A., Makansi, O., Hutter, F., Brox, T.: Uncertainty estimates and multi-hypotheses networks for optical flow. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 652–667 (2018)
7. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. arXiv preprint arXiv:1612.01474 (2016)
8. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: European conference on computer vision. pp. 746–760. Springer (2012)
9. Yang, G., Hu, P., Ramanan, D.: Inferring distributions over depth from a single image. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 6090–6096. IEEE (2019)