SLURP: Side Learning Uncertainty for Regression Problems — Supplementary Material —

Xuanlong Yu[1]
xuanlong.yu@universite-paris-saclay.fr
Gianni Franchi[2]
gianni.franchi@ensta-paris.fr
Emanuel Aldea[1]
emanuel.aldea@universite-paris-saclay.fr

[1] SATIE, Paris-Saclay University
Gif-sur-Yvette, France
[2] U2IS, ENSTA Paris
Institut Polytechnique de Paris
Palaiseau, France

The document is structured as follows. We recall first the abbreviations which are used and which will help the reader follow the next sections (Section 1). Then, we present and discuss the ablation study (Section 2). Section 3 provides all the information which is needed in order to duplicate and evaluate the result of the experiments performed in the main paper, more specifically the monocular depth estimation (Section 3.2), the optical flow estimation (Section 3.3) and the toy example (Section 3.4). Information about the accuracy of the main task models is included as well. Finally, Section 4 presents some additional qualitative results which provide further insights about the behavior of SLURP and of the other considered strategies.

1 Notations

In Table 1 we summarize the abbreviations used in the paper.

Abbreviation	Meaning
AUROC	Area under receiver operating characteristic curve
AUSE	Area under sparsification error
Confid	ConfidNet uncertainty estimation solution
DE	Deep ensembles
EE	Empirical ensemble
MC	MC-Dropout
MD	Monocular depth
MHP	Multi-hypothesis prediction network for optical flow uncertainty estimation
OF	Optical flow
SC	Sparsification curve
Single-PU	Single predictive uncertainty

Table 1: Summary of the abbreviations of the paper.

2 Ablation study

1. Ablation study settings: The ablation study for the monocular depth (MD) task is implemented on KITTI [8, 18] Eigen-split test set [5], Cityscapes test set [3], Foggy Cityscapes-DBF test set [17] and Rainy Cityscapes test set [9]. The ablation study for the optical flow (OF) task is implemented on FlyingChairs test set [2], Sintel training set [1] and KITTI 2015 training set [8, 15, 16]. Brief descriptions of these datasets can be found in the main paper. We use the same uncertainty evaluation metrics (AUSE and AUROC) as section 4.2 in the main paper.

2. Ablation study goals: We want to highlight the impact of the two considered inputs on the final performance (namely the image features and the prediction results features), and the impact of the considered loss (binary cross entropy loss and mean square error loss).

3. Results: The models with different inputs and different loss functions are presented as follows. Table 2 presents the model performance on OF task and Table 3 illustrates the results on MD task. Note that in the tables BCE and MSE denote binary cross entropy loss and mean square error loss respectively, PredOnly and RGBOnly denote the models taking only prediction map as input and the models taking only RGB image as input respectively. No special note means that the model will use both RGB and prediction results as input and BCE as the loss function (the default behavior).

4. Discussions: Firstly, regarding the performance of the different loss functions, we found that the results obtained with the BCE loss are almost systematically better than those provided when using MSE loss. We think this is because when we have a correctly trained predictor for the main task, most of the data points have minor errors, while a small number of data points have high errors. Using the MSE loss will amplify the more significant prediction errors and reduce the minor errors, making the model unable to fit well. Our target scaling uses a soft clipping strategy to centralize the distribution of data for better fitting.

For different inputs, we found that it is essential to use the prediction map as the input through the evaluation results. The input of the RGB image sometimes affects the generalization ability of uncertainty estimation if the main task model can generate already very good prediction results. According to the visualizations and evaluation results, we can see that the influence of the input of the prediction map is dominant because the uncertainty map of dual input and the one with only the prediction map input are similar. On the other hand, the RGB image can supplement some missing semantics of the prediction map, such as the Fig 1 FlyingChairs where RGB input can supplement the lack of chair legs and in the Fig 2 where RGB input supplement the uncertainty of the sky (although the sky does not have ground truth of depth, it should have a high degree of uncertainty).

Conditions					
Input source	RGB Input	✓	✗	✓	✓
	Prediction map Input	✗	✓	✓	✓
Loss	MSE	✗	✗	✓	✗
	BCE	✓	✓	✗	✓
Datasets	Criteria	Ours RGBOnly	Ours PredOnly	Ours MSE	Ours BCE
FlyingChairs	AUSE-EPE	1.82	1.24	1.41	**1.20**
	AUROC	0.944	0.972	0.967	**0.974**
KITTI	AUSE-EPE	8.40	4.87	5.40	**4.69**
	AUROC	0.586	**0.800**	0.793	**0.800**
Sintel Clean	AUSE-EPE	7.43	**2.73**	3.19	2.91
	AUROC	0.639	**0.898**	0.883	0.896
Sintel Final	AUSE-EPE	8.24	**2.71**	3.11	2.86
	AUROC	0.575	**0.907**	0.889	0.906

Table 2: Ablation study for the OF task. Bold value: result with the best performance. Blue value: second performance.

Conditions					
Input source	RGB Input	✓	✗	✓	✓
	Predinction map Input	✗	✓	✓	✓
Loss	MSE	✗	✗	✓	✗
	BCE	✓	✓	✗	✓
Dataset	Criteria	Ours RGBOnly	Ours PredOnly	Ours MSE	Ours BCE
KITTI	AUSE-RMSE	1.84	1.76	1.74	**1.68**
	AUSE-Absrel	4.45	4.31	**4.19**	4.36
	AUROC	0.879	0.890	0.894	**0.895**
Cityscapes	AUSE-RMSE	9.95	**9.40**	9.82	9.48
	AUSE-Absrel	10.68	**9.23**	10.29	10.90
	AUROC	0.344	**0.446**	0.414	0.400
After fine-tuning on Cityscapes					
Cityscapes	AUSE-RMSE	3.47	3.45	4.77	**3.05**
	AUSE-Absrel	6.71	**6.47**	6.93	6.55
	AUROC	0.837	0.844	0.766	**0.849**
Cityscapes Rainy s=1	AUSE-RMSE	3.97	3.43	4.80	**3.39**
	AUSE-Absrel	6.95	**5.52**	7.30	5.62
	AUROC	0.739	**0.795**	0.68	0.788
Cityscapes Rainy s=2	AUSE-RMSE	3.98	3.39	4.92	**3.36**
	AUSE-Absrel	6.68	**5.16**	7.09	5.28
	AUROC	0.747	**0.801**	0.689	0.794
Cityscapes Rainy s=3	AUSE-RMSE	4.11	**3.41**	5.07	**3.41**
	AUSE-Absrel	6.77	**4.85**	7.06	5.05
	AUROC	0.748	**0.811**	0.694	0.801
Cityscapes Foggy s=1	AUSE-RMSE	3.55	3.42	4.92	**3.04**
	AUSE-Absrel	6.40	**6.15**	6.92	6.25
	AUROC	0.835	0.841	0.763	**0.847**
Cityscapes Foggy s=2	AUSE-RMSE	3.51	3.39	5.03	**3.01**
	AUSE-Absrel	6.23	**5.98**	6.89	6.06
	AUROC	0.838	0.845	0.767	**0.852**
Cityscapes Foggy s=3	AUSE-RMSE	3.48	3.36	5.24	**3.08**
	AUSE-Absrel	5.97	**5.72**	6.75	5.80
	AUROC	0.845	0.852	0.773	**0.857**

Table 3: Ablation study for the MD task. Bold value: result with the best performance. Blue value: second performance. s (*e.g* s=1) indicates severity, higher the s value, higher the severity.

Figure 1: Uncertainty estimation examples in ablation study for OF task. The first row of each dataset block represents the input image pair, ground truth and predicted optical flow and the prediction error. The prediction map and error map are made by a single FlowNetS model as an example. The second row of each dataset block represents the uncertainty results in using SLURP side learner with different inputs and different loss functions. Black indicates higher uncertainty, white indicates lower uncertainty.

Figure 2: Uncertainty estimation examples in ablation study for MD task. The first row of each dataset block represents the input image, ground truth and predicted depth map and the prediction error. Since the ground truth is sparse, we use interpolation to rebuild the ground truth map just for visualization. The predicted depth map and error map are made by a single BTS model as an example. The second row of each dataset block represent the uncertainty results in using SLURP side learner with different inputs and different loss functions. For uncertainty maps, black indicates higher uncertainty, white indicates lower uncertainty. For depth maps, black represents deeper depth, and white represents shallower depth.

3 Experiments

3.1 Evaluation protocol details

3.1.1 Optical flow

Let us consider an optical flow dataset $D = \{(\mathbf{x}_i, \mathbf{y}_i)\}_i$, where $\mathbf{y}_i \in \mathcal{R}^2$, $\mathbf{y}_i = (u_i, v_i)$ is the ground truth optical flow for pixel $\mathbf{x}_i$. Below, $\hat{\mathbf{y}}_i = (\hat{u}_i, \hat{v}_i)$ represents the optical flow prediction.

1. End point error (EPE): The average end point error for valid pixels in D is the Euclidean distance between $\mathbf{y}_i$ and $\hat{\mathbf{y}}_i$:

$$EPE = \frac{1}{|D|} \sum_{(u_i, v_i) \in D} \sqrt{(u_i - \hat{u}_i)^2 + (v_i - \hat{v}_i)^2} \tag{1}$$

EPE metric is used for illustrating the performance of the models we use in OF tasks shown in Table 6 and it's also used in AUSE and AUROC evaluation for OF task.

3.1.2 Monocular depth

Let us consider a monocular depth dataset $D = \{(\mathbf{x}_i, d_i)\}_i$ where $d_i \in \mathcal{R}^+$ is the ground truth depth value for pixel $\mathbf{x}_i$. Below, $\hat{d}_i$ represents the depth prediction. The metrics we used in the evaluations are as follows:

1. Root mean square error (RMSE):

$$RMSE = \sqrt{\frac{1}{|D|} \sum_{d_i \in |D|} ||\hat{d}_i - d_i||^2} \tag{2}$$

2. Absolute relative error (Absrel):

$$Absrel = \frac{1}{|D|} \sum_{d_i \in D} |\hat{d}_i - d_i| / d_i \tag{3}$$

3. Threshold dk: Inlier metrics as proposed in [5], k in dk indicates the power of the threshold (thr), we take $thr = 1.25$. In this case, d1: $thr = 1.25$; d2: $thr = 1.25^2$; d3: $thr = 1.25^3$, and dk represents the proportion of pixels that meet the threshold condition:

$$dk = max(\frac{\hat{d}_i}{d_i}, \frac{d_i}{\hat{d}_i}) = \delta < thr^k \tag{4}$$

$$dk = \frac{|A|}{|D|}, \text{where } A = \left\{ \mathbf{x}_i, \text{such that } \delta_i = \max(\frac{\hat{d}_i}{d_i}, \frac{d_i}{\hat{d}_i}) \text{ and } \delta_i < thr^k \right\} \tag{5}$$

4. Squared relative difference (SqRel):

$$SqRel = \frac{1}{|D|} \sum_{d_i \in D} ||\hat{d}_i - d_i||^2 / d_i \tag{6}$$

5. Root mean square log error (RMSElog):

$$RMSElog = \sqrt{\frac{1}{|D|} \sum_{d_i \in |D|} ||\log \hat{d}_i - \log d_i||^2} \tag{7}$$

Models	Datasets	higher is better			lower is better				
		d1	d2	d3	Abs Rel	Sq Rel	RMSE	RMSE log	log10
MC	KITTI	0.945	0.992	0.998	0.072	0.287	2.902	0.107	0.031
	Cityscapes	0.103	0.255	0.453	1.051	18.942	18.986	0.842	0.324
EE (DE)	KITTI	0.957	0.993	0.999	0.059	0.233	2.688	0.093	0.026
	Cityscapes	0.214	0.430	0.560	0.837	14.459	18.441	0.845	0.298
Ours	KITTI	0.955	0.993	0.998	0.060	0.249	2.798	0.096	0.027
(Single-PU, Original [12])	Cityscapes	0.183	0.386	0.519	0.963	17.230	18.948	0.896	0.321
After fine-tuning on Cityscapes									
MC	Cityscapes	0.882	0.974	0.992	0.117	0.917	5.625	0.169	0.049
EE (DE)	Cityscapes	0.920	0.983	0.995	0.098	0.635	4.889	0.149	0.043
Ours (Single-PU)	Cityscapes	0.906	0.980	0.993	0.104	0.711	5.216	0.159	0.046

Table 4: The performance on KITTI and the performance on Cityscapes dataset before and after fine-tuning the models. Before fine-tuning: Training set: KITTI Eigen-split training set, Test set: KITTI Eigen-split test set and Cityscapes test set; After fine-tuning: Fine-tuning training set: Cityscapes training set, Test set: Cityscapes test set. The three main task models used in EE are also used in DE and Single-PU also shares the same main task model with our approach.

6. Average log10 error (log10):

$$log10 = \frac{1}{|D|} \sum_{d_i \in |D|} |\log_{10} \hat{d}_i - \log_{10} d_i| \tag{8}$$

These six metrics measure the performance of the MD models we use, and are shown in Table 4. At the same time, the first three metrics are also applied for AUSE and AUROC evaluations. Additionally, for AUROC, we choose $k = 1$ for threshold dk metric.

3.1.3 Sparsification plot settings

For both MD and OF tasks, the area under the sparsification error curve (AUSE) made by the sparsification curve is computed image-wise and not dataset-wise in our evaluations because of the high memory consumption which would be required for sorting vaues across the entire dataset.

Image-wise: We calculate the standardized AUSE for every image in the test set by ranking its pixels according to the corresponding predicted uncertainty and true error, then calculate the average AUSE after traversing the entire test set.

Dataset-wise: By collecting all pixels of all images in the test dataset, we calculate the AUSE by sorting their predicted uncertainty and true error.

3.2 Monocular depth estimation task supplement

3.2.1 Model precision

Table 4 shows the main task model performance for different uncertainty estimation approaches [6, 12, 13]. We have noticed that after the model is trained on KITTI, it cannot obtain reasonable accuracy on Cityscapes. This is because the ground truth of KITTI dataset is sparse, and only the lower half of the content is present. At the same time, the scene of Cityscapes is more complicated. Therefore, we fine-tune all models on Cityscapes to obtain reasonable accuracy.

3.2.2 Training settings

In the MD task, we choose to use a sequential training strategy for single predicted uncertainty (Single-PU) [12], deep ensembles (DE) [13] and our SLURP side learner. In other words, we first complete the training of the main task models (BTS [14]) and then train different uncertainty predictors according to the settings. Specifically, for Single-PU, we use an identical BTS model to estimate the uncertainty with using the output of the main task its corresponding main task model in the loss. For DE, it is a mixture of multiple Single-PUs, so we just repeat the previous procedures. Because of the ensemble property of empirical ensemble (EE) and DE, EE and DE can share the same main task predictors. In the same sense, the main task predictor of our side learner is chosen from one of EE(DE)'s main task predictors, which is the same one as the main task model of Single-PU. This method can ensure that the prediction accuracy of the main task will not be affected by the training of the uncertainty predictor. The ConfidNet [7] (Confid) implementation for BTS references its implementation on SegNet. The detailed operations are consistent with the descriptions in the main paper.
We build our side learner according to SLURP solution in the main paper (also shown in Fig 2 in the main paper) for BTS and here are some supplements. We directly use the frozen RGB feature maps from the encoder of main task BTS model. To convert 1-channel predicted depth map to 3-channel input, we expand it three times. The detailed settings for different uncertainty estimation models are listed in Table 5, all main task models are trained identically according to the original BTS [14] model training settings.

3.3 Optical flow supplement

3.3.1 Model precision

Table 6 shows the main task precision for different uncertainty estimation strategies. Our SLURP side learner picks one of the models from EE as our main task predictor. The main task models are trained only on FlyingChairs training set with official split and the KITTI dataset we choose for main task precision evaluation and also uncertainty estimation/evaluation is KITTI 2015 with occlusions. In the original FlowNetS paper [4], the precision evaluation is based on KITTI 2012 [7].

3.3.2 Training settings

In the optical flow task, for EE, we directly train multiple main task prediction models FlowNetS [4], and our side learner selects one of the models as our main task predictor. For Single-PU, because FlowNetS is relatively simple, we directly modify the original model to output two values for each pixel, one representing the predicted value of the main task and the other the uncertainty value. For DE, we train multiple Single-PU models. For multi-hypothesis prediction network (MHP) [11], we modified FlowNetS so that it can output eight (number of hypothesis) pairs of main task - uncertainty results. Futhermore, we use another FlowNetS as the MergeNet. It should be noted that, the authors did not mention the structural information about MergeNet in the paper. We choose FlowNetS based on the use of model stacking in the article. We train MHP followed by the two-stage training schedule provided in the supplementary of this paper.
For our SLURP side learner, since the encoder in FlowNetS is designed for capturing the object movement for two images and the total uncertainty will reflect only the semantics

Hyper-parameters	MC	EE	DE (Single-PU, Confid)	Ours
learning rate for main task model (Training on KITTI)	1e-4	1e-4	/	/
number of training epoch (Training on KITTI)	50	50	50	8
learning rate forside learner (Training on KITTI)	/	/	/	1e-4
learning rate for main task model (Fine-tuning on Cityscapes)	5e-5	5e-5	/	/
number of training epoch (Fine-tuning on Cityscapes)	30	30	30	16
learning rate for side learner (Fine-tuning on Cityscapes)	/	/	/	8e-5
learning rate for identical uncertainty estimator	/	/	5e-5	/
learning rate for side learner	/	/	/	1e-4
batch size	4			
number of training epoch	50	50	50	8
weight decay for main task model	1e-2	1e-2	/	/
weight decay for identical uncertainty estimator	/	/	1e-2	/
weight decay for side learner feature extractor	/	/	/	1e-3
weight decay for side learner uncertainty generation blocks	/	/	/	4e-4
Model structure and other settings				
encoder backbone for main task model, identical uncertainty estimator and side learner feature extractor	Densenet 161 [10]			
loss	same as BTS [10]	same as BTS [10]	Laplacian NLL	BCE $\lambda = 0.0125$
number of latent stages n	/	/	/	5
number of latent stage output channel c	/	/	/	1
number of final uncertainty output channel C_{out}	/	/	/	1
dropout rate p_d	0.4	/	/	/
ensemble size M	1	3	3 (1)	1
during inference time number of forward propagation	8	3	3 (1)	1

Table 5: MD model settings for MC, EE, DE, Single-PU, Confid and Ours.

Datasets	EE	MC	Single-PU	DE	Ours	Orginal [9]
FlyingChairs test	1.79	3.71	2.04	1.93	1.96	2.71
KITTI 2015 occ	18.36	16.53	21.21	20.78	19.39	/
KITTI 2012 noc	6.77	19.02	8.34	8.40	7.65	8.26
Sintel clean train	5.10	6.31	5.12	5.00	5.20	4.50
Sintel final train	6.50	6.97	6.53	6.41	6.62	5.45

Table 6: The main task accuracy for the uncertainty estimators in OF task. The values present the end-point error (EPE). Training set: FlyingChairs training set [9], Test set: FlyingChairs test set, KITTI 2012 noc [7] which was used in the original FlowNetS paper, KITTI 2015 occ [8, 15, 16] and Sintel full training set [1]

.

from the first image, we use two DenseNet161 backbones [10] as RGB and prediction map encoders respectively. We also used two DenseNet121 backbones for the lighter version. In order to transfer the 2-channel flow prediction to a 3-channel input, we just add one convolution layer to expend the channel number before the RGB feature extractor. All uncertainty model training settings are shown in Table 7.

3.4 Synthetic 1D regression task supplement

Because of the simplicity of the data, the main task predictor we use is a neural network composed of one hidden layer and 3000 neurons. In SLURP joint-training, we train our main

Hyper-parameters	MC	EE	DE (Single-PU, Confid)	Ours
learning rate for (modified) main task model	1e-4	1e-4	1e-4	/
learning rate for side learner	/	/	/	1e-4
batch size			8	
number of training epoch	216	216	216	30
weight decay for (modified) main task model	4e-4	4e-4	4e-4	/
weight decay for side learner feature extractors	/	/	/	1e-4
weight decay for side learner uncertainty generation blocks	/	/	/	4e-4
Model structure and other settings				
loss	same as FlowNetS [8]	same as FlowNetS [8]	Laplacian NLL	BCE $\lambda = 0.05$
number of latent stages n	/	/	/	5
number of latent stage output channel c	/	/	/	2
number of final uncertainty output channel C_{out}	/	/	/	1
dropout rate p_d	0.4	/	/	/
ensemble size M	1	3	3 (1)	1
during inference time number of forward propagation	8	3	3 (1)	1

Table 7: OF model settings for MC, EE, DE, Single-PU, Confid, MHP and Ours. MHP training setting is followed by the same schedule provided by is original paper.

task model and side learner at the same time without freezing any layers, and in SLURP sequential-training, we train our side learner while freezing the main task and using the latent values of it. For the side learner, following general SLURP solution, we use the same hidden layer as the prediction result feature extractor and three hidden layers with 128, 64, 16 neurons respectively as the context block in the uncertainty generation block, since we have only one stage, there is no fusion block in the end. The training details for all uncertainty estimation approaches are listed in Table 8.

We can give an insight that SLURP strategy can also work on 1D-regression tasks. In addition, the structure of the SLURP side learner is variable, and other uncertainty estimation methods are limited to the structure of the main task model, we are able to get better uncertainty results.

Hyper-parameters	MC	EE	DE (Single-PU)	SLURP joint-training	SLURP sequential-training
number of main task latent features			3000		
learning rate for main task model	1e-1	1e-1	1e-2	1e-1	/
learning rate for side learner feature extractor	/	/	/	1e-1	1e-4
learning rate for side learner uncertainty generation blocks	/	/	/	1e-4	1e-4
batch size			50		
number of training epoch			50		
weight decay for main task model	1e-2	1e-2	1e-2	1e-2	/
weight decay for side learner feature extractor	/	/	/	1e-2	1e-3
weight decay for side learner uncertainty generation blocks	/	/	/	1e-3	1e-2
Model structure and other settings					
loss	MSE	MSE	Gaussian NLL	Gaussian NLL	MSE
dropout rate	0.4	/	/	/	/
ensemble size M	1	3	3 (1)	1	1

Table 8: 1D regression task model settings.

4 More visualization results

Figure 3: Uncertainty estimation results for MD task. The ground truth maps are rebuilt by interpolation just for visualization. The depth prediction map and the error map are generated by a single BTS model as an example. MC-Dropout uncertainty maps are obtained by eight forward propagation, Deep ensembles and Empirical ensembles uncertainty maps are obtained from three models ensembles. For uncertainty maps, black indicates higher uncertainty, white indicates lower uncertainty. For depth maps, black represents deeper depth, and white represents shallower depth.

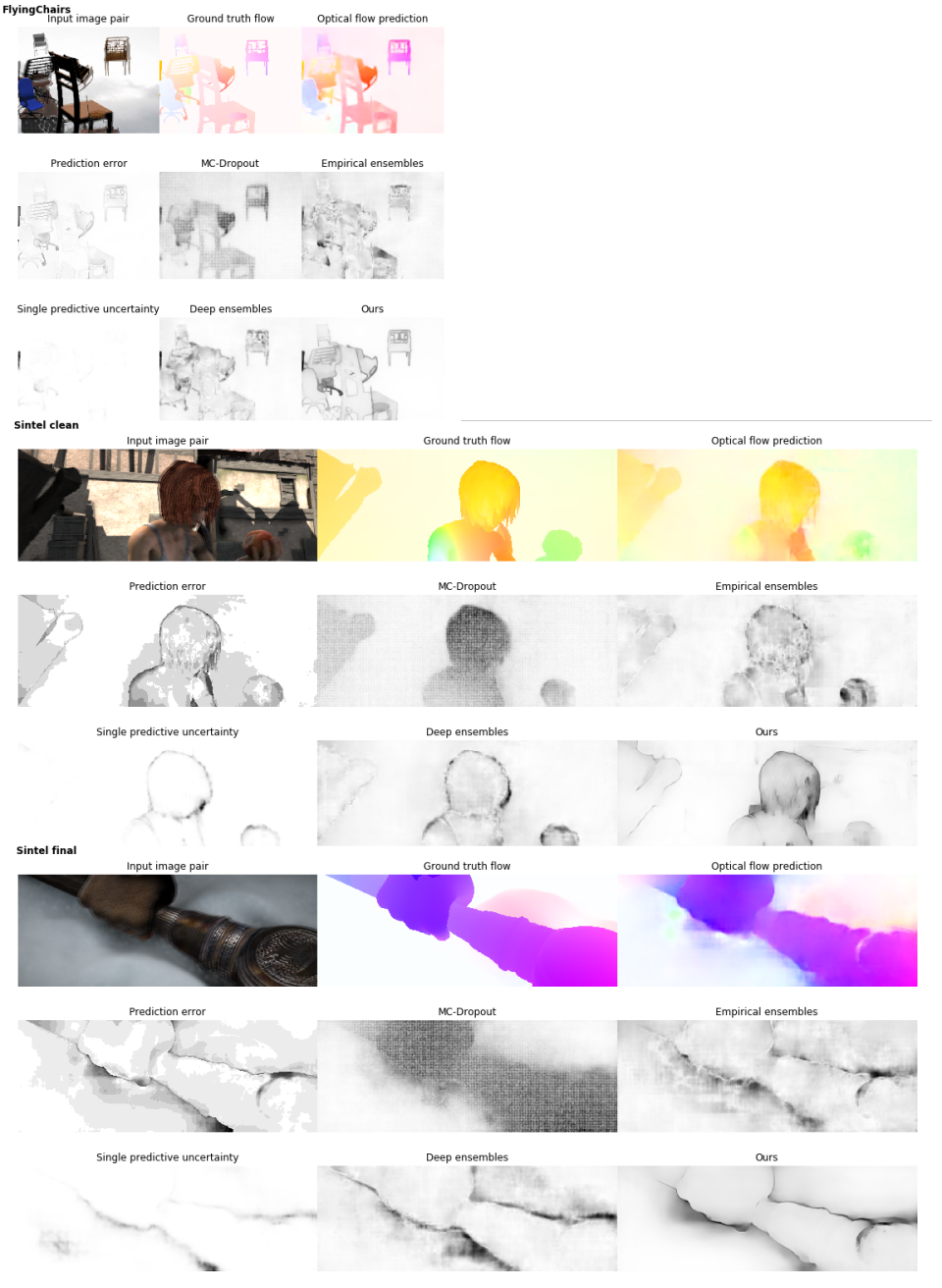

Figure 4: Uncertainty estimation results for OF task. The optical flow prediction map and the error map are generated by a single FlowNetS model as an example. MC-Dropout uncertainty maps are obtained by eight forward propagation, Deep ensembles and Empirical ensembles uncertainty maps are obtained from three models ensembles. For uncertainty maps, black indicates higher uncertainty, white indicates lower uncertainty.

References

[1] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, *European Conf. on Computer Vision (ECCV)*, Part IV, LNCS 7577, pages 611–625. Springer-Verlag, October 2012.

[2] Charles Corbière, Nicolas Thome, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez. Addressing failure prediction by learning model confidence. *arXiv preprint arXiv:1910.04851*, 2019.

[3] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[4] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015.

[5] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *arXiv preprint arXiv:1406.2283*, 2014.

[6] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[7] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[8] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11): 1231–1237, 2013.

[9] Xiaowei Hu, Chi-Wing Fu, Lei Zhu, and Pheng-Ann Heng. Depth-attentional features for single-image rain removal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8022–8031, 2019.

[10] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[11] Eddy Ilg, Ozgun Cicek, Silvio Galesso, Aaron Klein, Osama Makansi, Frank Hutter, and Thomas Brox. Uncertainty estimates and multi-hypotheses networks for optical flow. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 652–667, 2018.

[12] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[13] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *arXiv preprint arXiv:1612.01474*, 2016.

[14] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019.

[15] Moritz Menze, Christian Heipke, and Andreas Geiger. Joint 3d estimation of vehicles and scene flow. In *ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015.

[16] Moritz Menze, Christian Heipke, and Andreas Geiger. Object scene flow. *ISPRS Journal of Photogrammetry and Remote Sensing (JPRS)*, 2018.

[17] Christos Sakaridis, Dengxin Dai, Simon Hecker, and Luc Van Gool. Model adaptation with synthetic and real data for semantic dense foggy scene understanding. In *European Conference on Computer Vision (ECCV)*, pages 707–724, 2018.

[18] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *International Conference on 3D Vision (3DV)*, 2017.