Supplementary material
MonoProb: Self-Supervised Monocular Depth Estimation with Interpretable Uncertainty

A. Proof that we only need the marginals of the depth distribution

Let $N \in \mathbb{N}_+^*$, $D = [D_1, ..., D_N]$ be a multi-variate random variable and $\text{recons}(., I_s^*)$ a function so that:

$$\begin{aligned}\text{recons}(., I_s^*)\colon \mathbb{R}^N &\longrightarrow \mathbb{R}^{N\times 3}\\ d &\mapsto [\text{recons}_1(d_1, I_s^*), ..., \text{recons}_N(d_N, I_s^*)],\end{aligned}$$

where $\forall\ i \in \{1, ..., N\}$ and $x \in \mathbb{R}$, $\text{recons}_i(x, I_s^*) \in \mathbb{R}^3$ and $\eta \in \mathbb{R}^{N\times 2}$.

$$\begin{aligned}&\mathbb{E}_D[\text{recons}(D, I_s^*)|\eta]\\ =&\int_{\mathcal{D}} [\text{recons}_1(d_1, I_s^*), ..., \text{recons}_N(d_N, I_s^*)] p_D(d|\eta)\mathrm{d}d\\ =&\left[\int_{\mathcal{D}} \text{recons}_1(d_1, I_s^*) p_D(d|\eta)\mathrm{d}d, ..., \right.\\ &\qquad\left.\int_{\mathcal{D}} \text{recons}_N(d_N, I_s^*) p_D(d|\eta)\mathrm{d}d\right]\\ =&\left[\int_{\mathcal{D}} \text{recons}_1(d_1, I_s^*) p_{D_1}(d_1|\eta)\mathrm{d}d_1, ..., \right.\\ &\qquad\left.\int_{\mathcal{D}} \text{recons}_N(d_N, I_s^*) p_{D_N}(d_N|\eta)\mathrm{d}d_N\right]\\ =&[\mathbb{E}_{D_1}[\text{recons}_1(D_1, I_s^*)|\eta], ...,\\ &\qquad\mathbb{E}_{D_N}[\text{recons}_N(D_N, I_s^*)|\eta]].\end{aligned}$$

This shows that the marginals of D are sufficient to compute $\mathbb{E}_D[\text{recons}(D, I_s^*)|\eta]$.

B. Sampling strategy

The MonoProb depth estimator returns a map of HW univariate depth distributions of parameters η that are independently sampled n times so that for each distribution D_k with $k \in [\![1, HW]\!]$, the sample set $\mathcal{S}_\eta^k$ accurately represents the corresponding distributions. Finally, n depth maps of size HW are obtained. We design our sampling strategy for families of depth distributions belonging to the symmetric generalized normal distribution with a density of the form:

$$\begin{aligned}f &: \mathbb{R} \to \mathbb{R}\\ x &\mapsto \frac{\beta}{2\gamma\Gamma(1/\beta)} \exp\left(-(|x-\mu|/\gamma)^\beta\right)\end{aligned} \tag{1}$$

where μ is the mean, β is the shape parameter (1 for a Laplace distribution and 2 for a Gaussian distribution) and γ is the scale parameter, which can be expressed as a function of the standard deviation σ (so that $\gamma = \sigma/\sqrt{2}$ for the Laplace distribution and $\gamma = \sqrt{2}\sigma$ for the Gaussian distribution). We sample only indices for which the ratio between their density and the density of the mean $f(\mu)$ is in $\{\frac{2i}{n+1}\}_{i=1}^{\lfloor\frac{n+1}{2}\rfloor}$. This results in a set $\mathcal{S}_\eta^k$ of n samples, symmetrically and evenly distributed around the mean. These samples are also controlled to be close enough to the mean so that their density is not too close to zero and so that at most one sample only is equal to the mean. Furthermore, the sampling strategy was chosen because the relationship between the samples and the parameters of the distribution D_k facilitates the computation of the backpropagation compared to if we had used quantiles of the distributions. These samples are:

$$\begin{aligned}\forall n > 0, \mathcal{S}_\eta^k &= \left\{ s \;\middle|\; \frac{f(s)}{f(\mu)} = \frac{2i}{n+1} \right\}_{i=1}^{\lfloor\frac{n+1}{2}\rfloor}\\ &= \left\{ s \;\middle|\; \exp(-(|s-\mu|/\gamma)^\beta = \frac{2i}{n+1} \right\}_{i=1}^{\lfloor\frac{n+1}{2}\rfloor}\\ &= \left\{ \mu \pm \gamma \left(-\log\left(\frac{2i}{n+1}\right)\right)^{\frac{1}{\beta}} \right\}_{i=1}^{\lfloor\frac{n+1}{2}\rfloor}\end{aligned} \tag{2}$$

In the paper, we carried out experiments with $n \in \{5, 9, 13\}$. For $n = 5$, the corresponding sample set $\mathcal{S}_\eta^k$ is:

$$\mathcal{S}_\eta^k = \left\{ \mu + \delta \;\middle|\; \delta \in \left\{ -\gamma \left(-\log \frac{1}{3} \right)^{\frac{1}{\beta}}, \right. \right.$$
$$-\gamma \left(-\log \frac{2}{3} \right)^{\frac{1}{\beta}},$$
$$0, \qquad (3)$$
$$\gamma \left(-\log \frac{2}{3} \right)^{\frac{1}{\beta}},$$
$$\left. \left. \gamma \left(-\log \frac{1}{3} \right)^{\frac{1}{\beta}} \right\} \right\}.$$

C. Comparison with [2, 4]'s methods

In Tab. 1, we compare our MonoProb methods with and without self-distillation to all the methods introduced in [2, 4], including those that require more than one inference to predict an uncertainty. We provide results for the three training paradigms that are: monocular video supervision only (M), stereo supervision only (S), and both monocular video and stereo supervision (MS). For each method, we report the number of trainings (#Trn) and the number of inferences (#Inf) required to generate the depth and uncertainty at test time. #Inf must not be confused with #Fwd used in [4], which is the number of forwards required at test time to estimate the depth only. The results of the concurrent methods were taken from [2, 4]. Our new metrics have been computed using the checkpoints given by [2, 4] and on methods that provide an interpretable uncertainty as defined in the paper. The results show that our MonoProb and self-distilled MonoProb methods have similar depth performance to the other approaches. Likewise, our MonoProb without self-distillation provides competitive results in terms of uncertainty estimation. Our self-distilled MonoProb method shows overall better uncertainty estimation performance than other approaches. It even compares favorably with the methods that require more than one inference to predict uncertainty.

D. Complementary qualitative results

We provide qualitative results on KITTI [3] in Fig. 1 Make3D [5] in Fig. 2 and Nuscenes [1] in Fig. 3.

Sup	Methods	#Trn	#Inf	Abs Rel ↓	RMSE ↓	$\delta < 1.25$ ↑	Abs Rel		RMSE		$\delta < 1.25$		ARU↓	RMSU↓
							AUSE ↓	AURG↑	AUSE ↓	AURG↑	AUSE ↓	AURG↑		
M	[2]	1	1	0.090	3.942	0.914	-	-	-	-	-	-	-	-
M	[2]-Post	1	2	0.088	3.841	0.917	0.044	0.012	2.864	0.412	0.056	0.022	1.670	17.23
M	[4]-Repr	1	1	0.092	3.936	0.912	0.051	0.008	2.972	0.381	0.069	0.013	-	-
M	[4]-Log	1	1	0.091	4.052	0.910	0.039	0.020	2.562	0.916	0.044	0.038	-	-
M	[4]-Self	2	1	**0.087**	3.826	**0.920**	0.030	0.026	2.009	1.266	0.030	0.045	0.074	3.730
M	[4]-Drop	1	8	0.101	4.146	0.892	0.065	0.000	2.568	0.944	0.097	0.002	3.041	33.90
M	[4]-Boot	8	8	0.092	3.821	0.911	0.058	0.001	3.982	-0.743	0.084	-0.001	0.791	8.635
M	[4]-Snap	1	8	0.091	3.921	0.912	0.059	-0.001	3.979	-0.639	0.083	-0.002	0.361	3.956
M	[4]-Boot+Log	8	8	0.092	3.850	0.910	0.038	0.021	2.449	0.820	0.046	0.037	-	-
M	[4]-Boot+Self	9	8	0.088	3.799	0.918	0.029	0.028	1.924	1.316	0.028	0.049	0.333	3.609
M	[4]-Snap+Log	1	8	0.092	3.961	0.911	0.038	0.022	2.385	1.001	0.043	0.039	-	-
M	[4]-Snap+Self	2	8	0.088	3.832	0.919	0.031	0.026	2.043	1.230	0.030	0.045	0.233	**2.554**
M	Ours	1	1	0.089	3.852	0.914	0.031	0.026	0.719	2.560	0.030	0.050	**0.064**	2.912
M	Ours-self	2	1	**0.087**	**3.762**	0.919	**0.022**	**0.034**	**0.326**	**2.880**	**0.014**	**0.061**	0.066	2.969
S	[2]	1	1	0.085	3.942	0.912	-	-	-	-	-	-	-	-
S	[2]-Post	1	2	0.084	3.777	0.915	0.036	0.020	2.523	0.736	0.044	0.034	0.292	3.016
S	[4]-Repr	1	1	0.085	3.873	0.913	0.040	0.017	2.275	1.074	0.050	0.030	-	-
S	[4]-Log	1	1	0.085	3.860	0.915	0.022	0.036	0.938	2.402	**0.018**	0.061	-	-
S	[4]-Self	2	1	0.084	3.835	0.915	0.022	0.035	1.679	1.642	0.022	0.056	0.083	3.686
S	[4]-Drop	1	8	0.129	4.908	0.819	0.103	-0.029	6.163	-2.169	0.231	-0.080	5.494	61.84
S	[4]-Boot	8	8	0.085	**3.772**	0.914	0.028	0.029	2.291	0.964	0.031	0.048	0.496	5.211
S	[4]-Snap	1	8	0.085	3.849	0.912	0.028	0.029	2.252	1.077	0.030	0.051	0.255	**2.684**
S	[4]-Boot+Log	8	8	0.085	3.777	0.913	0.020	**0.038**	0.807	2.455	**0.018**	**0.063**	-	-
S	[4]-Boot+Self	9	8	0.085	3.793	0.914	0.023	0.035	1.646	1.628	0.021	0.058	0.398	4.288
S	[4]-Snap+Log	1	8	**0.083**	3.833	0.914	0.021	0.037	0.891	2.426	**0.018**	0.061	-	-
S	[4]-Snap+Self	2	8	0.086	3.859	0.912	0.023	0.035	1.710	1.623	0.023	0.058	0.282	3.025
S	Ours	1	1	0.084	3.834	**0.916**	0.023	0.033	0.661	2.655	0.023	0.055	0.075	3.540
S	Ours-self	2	1	0.084	3.792	0.914	**0.018**	**0.038**	**0.349**	**2.924**	0.019	0.060	**0.072**	3.068
MS	[2]	1	1	0.084	3.739	0.918	-	-	-	-	-	-	-	-
MS	[2]-Post	1	2	**0.082**	**3.666**	**0.919**	0.036	0.018	2.498	0.655	0.044	0.031	0.290	2.974
MS	[4]-Repr	1	1	0.084	3.828	0.913	0.046	0.010	2.662	0.635	0.062	0.018	-	-
MS	[4]-Log	1	1	0.083	3.790	0.916	0.028	0.029	1.714	1.562	0.028	0.050	-	-
MS	[4]-Self	2	1	0.083	3.682	**0.919**	0.022	0.033	1.654	1.515	0.023	0.052	0.083	3.686
MS	[4]-Drop	1	8	0.172	5.885	0.679	0.103	-0.027	7.114	-2.580	0.303	-0.081	5.547	62.54
MS	[4]-Boot	8	8	0.086	3.787	0.910	0.028	0.030	2.269	0.985	0.034	0.049	0.564	5.898
MS	[4]-Snap	1	8	0.085	3.806	0.914	0.029	0.028	2.245	1.029	0.033	0.047	0.254	**2.706**
MS	[4]-Boot+Log	8	8	0.086	3.771	0.911	0.030	0.028	1.962	1.282	0.032	0.051	-	-
MS	[4]-Boot+Self	9	8	0.085	3.704	0.915	0.023	0.033	1.688	1.494	0.023	0.056	0.355	3.842
MS	[4]-Snap+Log	1	8	0.084	3.828	0.914	0.030	0.027	2.032	1.272	0.032	0.048	-	-
MS	[4]-Snap+Self	2	8	0.085	3.715	0.916	0.023	0.034	1.684	1.510	0.023	0.055	0.276	2.979
MS	Ours	1	1	0.084	3.806	0.915	0.027	0.029	0.840	2.436	0.029	0.049	**0.077**	3.573
MS	Ours-self	2	1	**0.082**	3.667	**0.919**	**0.016**	**0.039**	**0.293**	**2.859**	**0.014**	**0.061**	0.078	3.528

Table 1. Results of monocular video supervision only (M), stereo supervision only (S), and both monocular video and stereo supervision (MS) trainings of our MonoProb methods with and without self-distillation compared to methods that require one or more inferences to provide an uncertainty estimate.

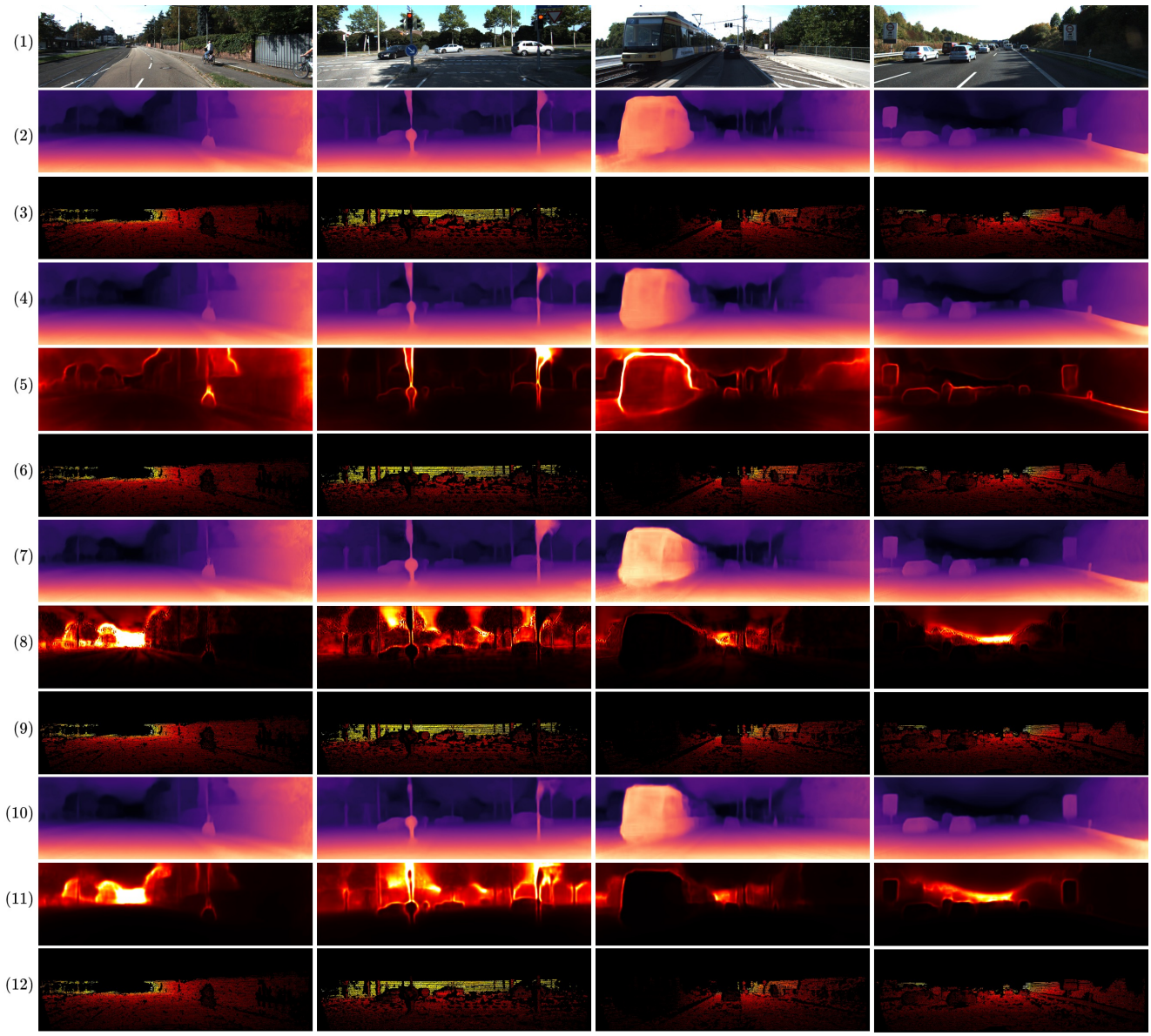

Figure 1. Qualitative results of monocular trainings on KITTI [3]. From top to bottom, (1) input image, (2) depth map from [2], (3) error map from [2], (4) depth map from [4]-Self, (5) uncertainty map from [4]-Self, (6) error map from [4]-Self, (7) depth map from our MonoProb without self-distillation, (8) uncertainty map from our MonoProb without self-distillation, (9) error map from our MonoProb without self-distillation, (10) depth map from our self-distilled MonoProb, (11) uncertainty map from our self-distilled MonoProb, (12) error map from our self-distilled MonoProb.

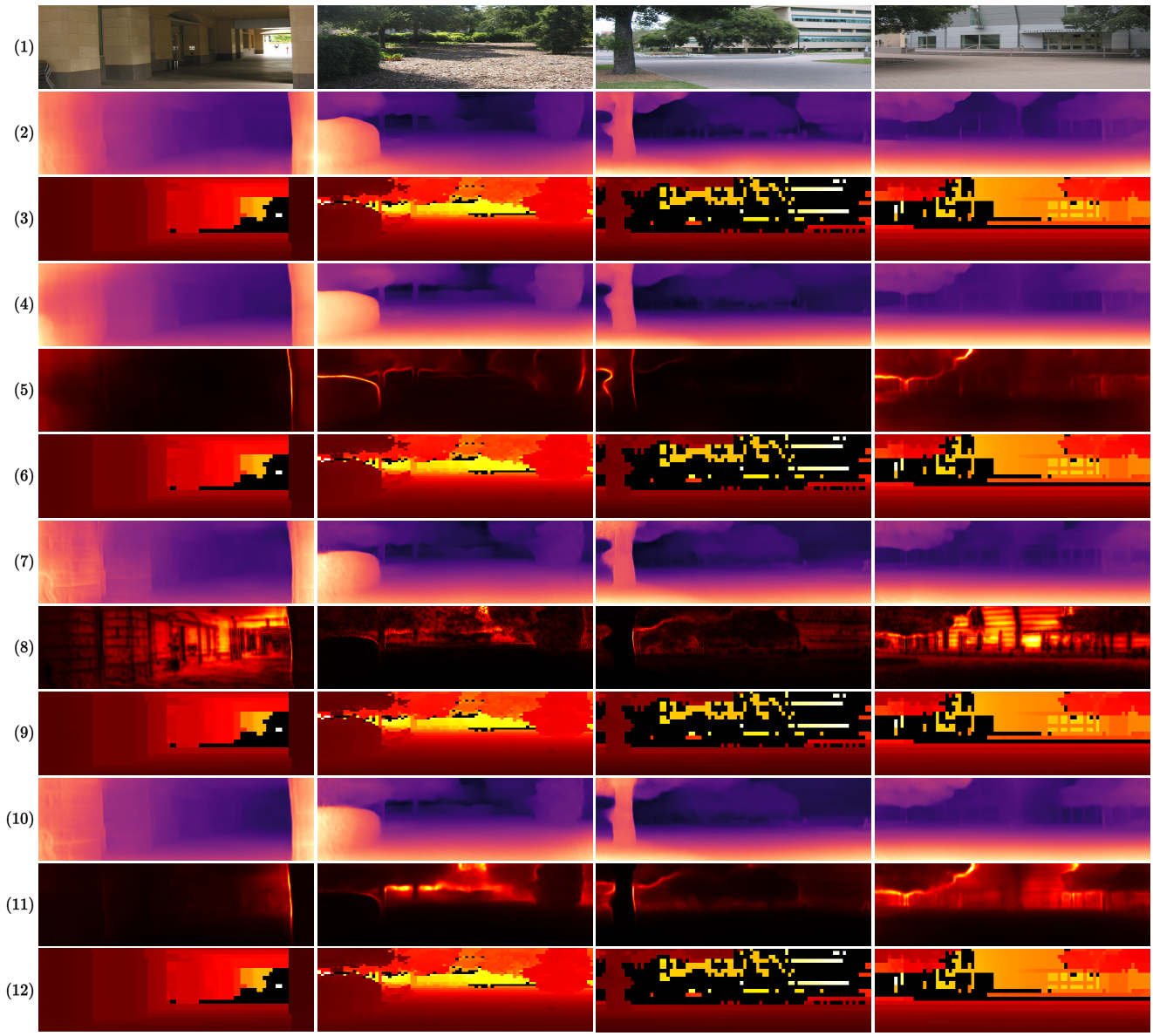

Figure 2. Qualitative results of monocular trainings on Make3D [5]. From top to bottom, (1) input image, (2) depth map from [2], (3) error map from [2], (4) depth map from [4]-Self, (5) uncertainty map from [4]-Self, (6) error map from [4]-Self, (7) depth map from our MonoProb without self-distillation, (8) uncertainty map from our MonoProb without self-distillation, (9) error map from our MonoProb without self-distillation, (10) depth map from our self-distilled MonoProb, (11) uncertainty map from our self-distilled MonoProb, (12) error map from our self-distilled MonoProb.

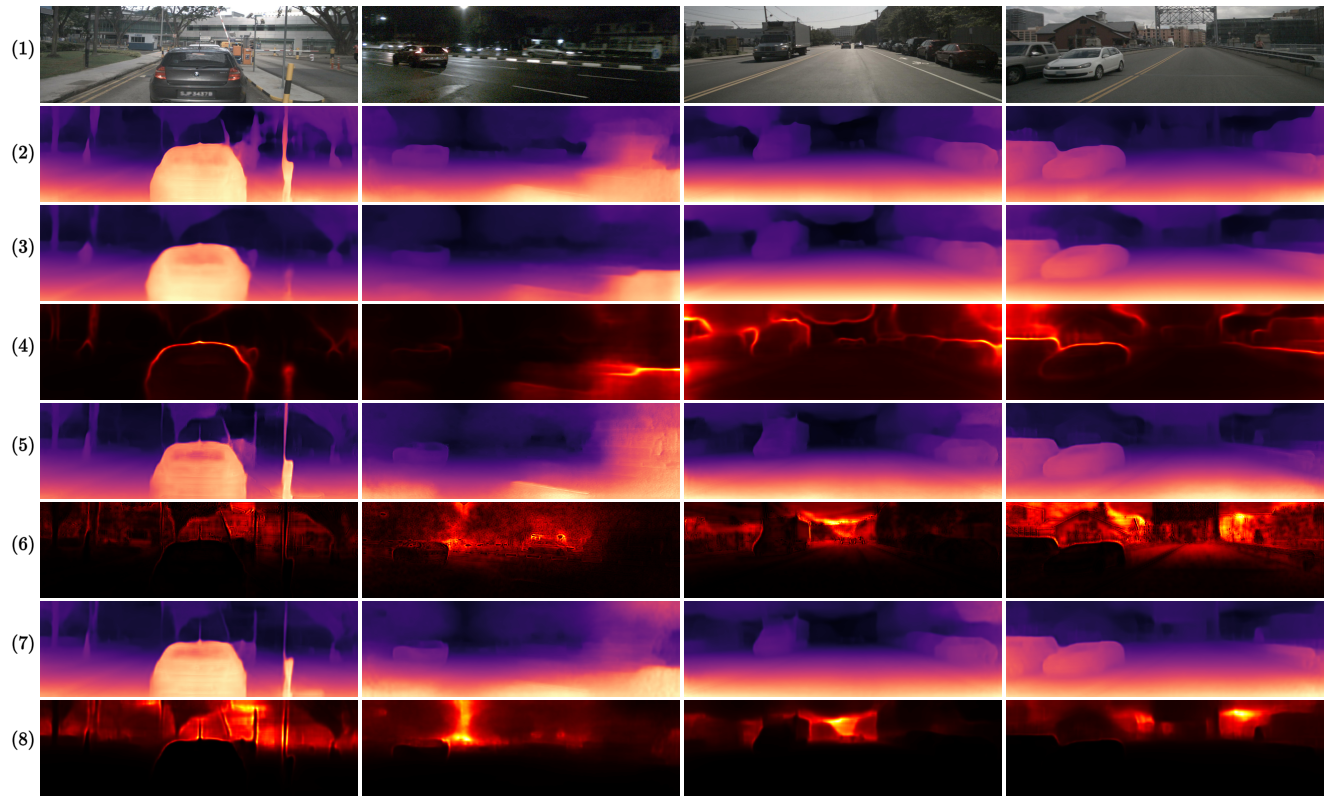

Figure 3. Qualitative results of monocular trainings on Nuscenes [1]. From top to bottom, (1) input image, (2) depth map from [2], (3) depth map from [4]-Self, (4) uncertainty map from [4]-Self, (5) depth map from our MonoProb without self-distillation, (6) uncertainty map from our MonoProb without self-distillation, (7) depth map from our self-distilled MonoProb, (8) uncertainty map from our self-distilled MonoProb. We do not provide error maps because the high sparsity of the ground truth makes them unreadable.

References

[1] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 2, 6

[2] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3828–3838, 2019. 2, 3, 4, 5, 6

[3] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015. 2, 4

[4] Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Mattoccia. On the uncertainty of self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3227–3237, 2020. 2, 3, 4, 5, 6

[5] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Learning 3-d scene structure from a single still image. In *2007 IEEE 11th international conference on computer vision*, pages 1–8. IEEE, 2007. 2, 5