On the uncertainty of self-supervised monocular depth estimation

Matteo Poggi Filippo Aleotti Fabio Tosi Stefano Mattoccia
Department of Computer Science and Engineering (DISI)
University of Bologna, Italy
{m.poggi, filippo.aleotti2, fabio.tosi5, stefano.mattoccia}@unibo.it

Abstract

Self-supervised paradigms for monocular depth estimation are very appealing since they do not require ground truth annotations at all. Despite the astonishing results yielded by such methodologies, learning to reason about the uncertainty of the estimated depth maps is of paramount importance for practical applications, yet uncharted in the literature. Purposely, we explore for the first time how to estimate the uncertainty for this task and how this affects depth accuracy, proposing a novel peculiar technique specifically designed for self-supervised approaches. On the standard KITTI dataset, we exhaustively assess the performance of each method with different self-supervised paradigms. Such evaluation highlights that our proposal i) always improves depth accuracy significantly and ii) yields state-of-the-art results concerning uncertainty estimation when training on sequences and competitive results uniquely deploying stereo pairs.

1. Introduction

Depth estimation is often pivotal to a variety of high-level tasks in computer vision, such as autonomous driving, augmented reality, and more. Although active sensors such as LiDAR are deployed for some of the applications mentioned above, estimating depth from standard cameras is generally preferable due to several advantages. Among them: the much lower cost of standard imaging devices, their higher resolution and frame rate allow for more scalable and compelling solutions.

In computer vision, depth perception from two [59] or multiple images [60] has a long history. Nonetheless, only in the last decade depth estimation from a single image [57] became an active research topic. On the one hand, this direction is particularly attractive because it overcomes several limitations of the traditional multi-view solutions (e.g., occlusions, overlapping framed area, and more), enabling depth perception with any device equipped with a camera. Unfortunately, it is an extremely challenging task due to the ill-posed nature of the problem.

Deep learning ignited the spread of depth-from-mono frameworks [13, 38, 15], at the cost of requiring a large number of image samples annotated with ground truth depth labels [47, 68] to achieve satisfying results. However, sourcing annotated depth data is particularly expensive and cumbersome. Indeed, in contrast to many other supervised tasks for which offline handmade annotation is tedious, yet relatively easy; gathering accurate depth labels requires active (and often expensive) sensors and specific calibration, making offline annotation hardly achievable otherwise. Self-supervised [19, 82, 45, 56, 53] or weakly supervised [76, 65, 72] paradigms, leveraging on image reprojection and noisy labels respectively, have removed this issue and yield accuracy close to supervised methods [15], neglecting at all the deployment of additional depth sensors for labeling purposes. Among self-supervised paradigms, those deploying monocular sequences are more challenging since...
scale and camera poses are unknown, yet preferred for most practical applications since they allow gathering of training data with the same device used to infer depth.

As for other perception strategies, it is essential to find out failure cases, when occurring, in monocular depth estimation networks. For instance, in an autonomous driving scenario, the erroneous perception of the distance to pedestrians or other vehicles might have dramatic consequences. Moreover, the ill-posed nature of depth-from-mono perception task makes this eventuality much more likely to occur compared to techniques leveraging scene geometry [59, 60]. In these latter cases, estimating the uncertainty (or, complementary, the confidence) proved to be effective for depth-from-stereo, by means of both model-based [24] and learning-based [55, 30] methods, optical flow [27], and semantic segmentation [26, 30]. Despite the steady progress in other related fields, uncertainty estimation for self-supervised paradigms remains almost unexplored or, when faced, not quantitatively evaluated [32].

Whereas concurrent works in this field [20, 72, 65] targeted uniquely depth accuracy, we take a breath on this rush and focus for the first time, to the best of our knowledge, on uncertainty estimation for self-supervised monocular depth estimation networks, showing how this practise enables to improve depth accuracy as well.

Our main contributions can be summarized as follows:

- A comprehensive evaluation of uncertainty estimation approaches tailored for the considered task.
- An in-depth investigation of how the self-supervised training paradigm deployed impacts uncertainty and depth estimation.
- A new and peculiar Self-Teaching paradigm to model uncertainty, particularly useful when the pose is unknown during the training process, always enabling to improve depth accuracy.

Deploying standard metrics in this field, we provide exhaustive experimental results on the KITTI dataset [18]. Figure 1 shows the output of a state-of-the-art monocular depth estimator network enriched to model uncertainty. We can notice how our proposal effectively allows to detect wrong predictions (e.g., in the proximity of the person riding the bike).

2. Related work

In this section, we review the literature concerning self-supervised monocular depth estimation and techniques to estimate uncertainty in deep neural networks.

**Self-supervision for mono.** The advent of deep learning, together with the increasing availability of ground truth depth data, led to the development of frameworks [38, 40, 74, 15] achieving unpaired accuracy compared to previous approaches [58, 37, 14]. Nonetheless, the effort to collect large amounts of labeled images is high. Thus, to overcome the need for ground truth data, self-supervision in the form of image reconstruction represents a prevalent research topic right now. Frameworks leveraging on this paradigm belong to two (not mutually exclusive) categories, respectively supervised through monocular sequences or stereo pairs.

The first family of networks jointly learns to estimate the depth and relative pose between two images acquired by a moving camera. Seminal work in this direction is [82], extended by leveraging on point-cloud alignment [45], differentiable DVO [69], optical flow [78, 83, 11, 3], semantic [66] or scale consistency [5]. One of the shortcomings of these approaches is represented by moving objects appearing in the training images, addressed in [8, 75] employing instance segmentation and subsequent motion estimation of the segmented dynamic objects.

For the second category, pivotal are the works by Garg et al. [17] and Godard et al. [19]. Other methods improved efficiency [53, 50] to enable deployment on embedded devices, or accuracy by simulating a trinocular setup [56], jointly learning for semantic [79], using higher resolution [51], GANs [1], sparse inputs from visual odometry [2] or a teacher-student scheme [52]. Finally, approaches leveraging both kind of supervisions have been proposed in [80, 77, 41, 20].

**Weak-supervision for mono.** A trade-off between self and full supervision is represented by another family of approaches leveraging weaker annotations. In this case, labels can be sourced from synthetic datasets [46], used to train stereo networks for single view stereo [42] and label distillation [22] or in alternative to learn depth estimation and perform domain transfer when dealing with real images [4].

Another source of weak supervision consists of using noisy annotations obtained employing the raw output of a LiDAR sensor [35] or model-based algorithms. In this latter case, the use of conventional stereo algorithms such as SGM [23] to obtain proxy labels [65, 72], optionally together with confidence measures [64], allowed improving self-supervision from stereo pairs. Other works distilled noisy labels leveraging on structure from motion [32] or direct stereo odometry [76].

**Uncertainty estimation.** Estimating the uncertainty (or, complementary, confidence) of cues inferred from images is of paramount importance for their deployment in real computer vision applications. This aspect has been widely explored even before the spread of deep learning, for instance, when dealing with optical flow and stereo matching. Concerning optical flow, uncertainty estimation methods belong to two main categories: model-inherent and post-hoc. The former family [7, 36, 71] estimates uncertainty
scores based on the internal flow estimation model, \textit{i.e.}, energy minimization models, while the latter\cite{43,33,34} analyzing already estimated flow fields. Regarding stereo vision, confidence estimation has been inferred similarly. At first, from features extracted by the internal disparity estimation model, \textit{i.e.}, the cost volume\cite{24}, then by means of deep learning on already estimated disparity maps\cite{55,61,54,67,31}.

Uncertainty estimation has a long history in neural networks as well, starting with Bayesian neural networks\cite{44,10,73}. Different models are sampled from the distribution of weights to estimate mean and variance of the target distribution in an \textit{empirical} manner. In\cite{21,6}, sampling was replaced by variational inference. Additional strategies to sample from the distribution of weights are bootstrapped ensembles\cite{39} and Monte Carlo Dropout\cite{16}. A different strategy consists of estimating uncertainty in a \textit{predictive} manner. Purposely, a neural network is trained to infer the mean and variance of the distribution rather than a single value\cite{49}. This strategy is both effective and cheaper than empirical strategies, since it does not require multiple forward passes and can be adapted to self-supervised approaches as shown in\cite{32}. Recent works\cite{29,30} combined both in a joint framework.

Finally, Ilg\textit{et al}.\cite{27} conducted studies about uncertainty modelling for deep optical flow networks. Nonetheless, in addition to the different nature of our task (\textit{i.e.}, the ill-posed monocular depth estimation problem), our work differs for the supervision paradigm, traditional in their case and self-supervised in ours.

3. Depth-from-mono and uncertainty

In this section, we introduce how to tackle uncertainty modelling with self-supervised depth estimation frameworks. Given a still image \( I \) any depth-from-mono framework produces an output map \( d \) encoding the depth of the observed scene. When full supervision is available, to train such a network we aim at minimizing a loss signal \( L_{fs} \) obtained through a generic function \( F \) of inputs estimated \( d \) and ground truth \( d^\ast \) depth maps.

\begin{equation}
L_{fs} = F(d, d^\ast)
\end{equation}

When traditional supervision is not available, it can be replaced by self-supervision obtained through image reconstruction. In this case, the ground truth map \( d^\ast \) is replaced by a second image \( I^\dagger \). Then, by knowing camera intrinsics \( K \), \( K^\dagger \) and the relative camera pose \( (R|t) \) between the two images, a reconstructed image \( \hat{I} \) is obtained as a function \( \pi \) of intrinsics, pose, image \( I^\dagger \) and depth \( d \), enabling to compute a loss signal \( L_{ss} \) as a generic \( F \) of inputs \( \hat{I} \) and \( I \).

\begin{equation}
L_{ss} = F(\hat{I}, I) = F(\pi(I^\dagger, K^\dagger, K^\dagger t, K, d), I)
\end{equation}

\( I \) and \( I^\dagger \) can be acquired either by means of a single moving camera or with a stereo rig. In this latter case, \((R|t)\) is known beforehand thanks to the stereo calibration parameters, while for images acquired by a single camera it is usually learned jointly to depth, both up to a scale factor. A popular choice for \( F \) is a weighted sum between L1 and Structured Similarity Index Measure (SSIM)\cite{70}

\begin{equation}
F(\hat{I}, I) = \alpha \cdot \frac{1 - \text{SSIM}(\hat{I}, I)}{2} + (1 - \alpha) \cdot |\hat{I} - I|
\end{equation}

with \( \alpha \) commonly set to 0.85\cite{20}. In case of \( K \) frames used for supervision, coming for example by joint monocular and stereo supervision, for each pixel \( q \) the minimum among computed losses allows for robust reprojection\cite{20}

\begin{equation}
L_{ss}(q) = \min_{i \in [0..K]} F(\hat{I}_i(q), I(q))
\end{equation}

Traditional networks are deterministic, producing a single output typically corresponding to the mean value of the distribution of all possible outputs \( p(d^\ast|I,D) \), \( D \) being a dataset of images and corresponding depth maps. Estimating the variance of such distribution allows for modelling uncertainty on the network outputs, as shown in\cite{28,29} and depicted in Figure 2, a) in empirical way, b) by learning a predictive model or c) combining the two approaches.

First and foremost, we point out that the self-supervision provided to the network is \textit{indirect} with respect to its main

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure2}
\caption{Overview of uncertainty estimation implementations. Respectively a) empirical methods model uncertainty as the variance of predictions from a subset of all the possible instances of the same network, b) predictive are trained to estimate depth and uncertainty as mean and variance of a distribution and c) Bayesian methods are approximated\cite{48} by sampling multiple predictive models and summing single uncertainties with the variance of the depth predictions.}
\end{figure}
task. This means that the network estimates are not optimized with respect to the desired statistical distribution, i.e., depth $d^*$, but they are an input parameter of a function ($\pi$) optimized over a different statistical model, i.e., image $\mathcal{I}$. While this does not represent an issue for empirical methods, predictive methods like negative log-likelihood minimization can be adapted to this paradigm as done by Klodt and Vedaldi [32]. Nevertheless, we will show how this solution is sub-optimal when the pose is unknown, i.e., when $\pi$ is function of two unknown parameters.

3.1. Uncertainty by image flipping

A simple strategy to estimate uncertainty is inspired by the post-processing (Post) step proposed by Godard et al. [19]. Such a refinement consists of estimating two depth maps $d$ and $\tilde{d}$ for image $\mathcal{I}$ and its horizontally flipped counterpart $\tilde{\mathcal{I}}$. The refined depth map $d^r$ is obtained by averaging $d$ and $\tilde{d}$, i.e., back-flipped $\tilde{d}$. We encode the uncertainty for $d^r$ as the difference between the two

$$u_{\text{Post}} = |d - \tilde{d}|$$

i.e., the variance over a small distribution of outputs (i.e., two), as typically done for empirical methods outlined in the next section. Although this method requires $2 \times$ forwards at test time compared to the raw depth-from-mono model, as shown in Figure 3, it can be applied seamlessly to any pre-trained framework without any modification.

3.2. Empirical estimation

This class of methods aims at encoding uncertainty empirically, for instance, by measuring the variance between a set of all the possible network configurations. It allows to explain the model uncertainty, namely epistemic [29]. Strategies belonging to this category [27] can be applied to self-supervised frameworks straightforwardly.

**Dropout Sampling (Drop).** Early works estimated uncertainty in neural networks [44] by sampling multiple networks from the distribution of weights of a single architecture. Monte Carlo Dropout [63] represents a popular method to sample $N$ independent models without requiring multiple and independent trainings. At training time, connections between layers are randomly dropped with a probability $p$ to avoid overfitting. At test time, all connections are kept. By keeping dropout enabled at test time, we can perform multiple forwards sampling a different network every time. Empirical mean $\mu(d)$ and variance $\sigma^2(d)$ are computed, as follows, performing multiple ($N$) inferences:

$$\mu(d) = \frac{1}{N} \sum_{i=1}^{N} d_i$$

$$u_{\text{Drop}} = \sigma^2(d) = \frac{1}{N} \sum_{i=1}^{N} (d_i - \mu(d))^2$$

At test time, using the same number of network parameters, $N \times$ forwards are required.

**Bootstrapped Ensemble (Boot).** A simple, yet effective alternative to weights sampling is represented by training an ensemble of $N$ neural networks [39] randomly initializing $N$ instances of the same architecture and training them with bootstrapping, i.e., on random subsets of the entire training set. This strategy produces $N$ specialized models. Then, similarly to dropout sampling, we can obtain empirical mean $\mu(d)$ and variance $\sigma^2(d)$ in order to approximate the mean and variance of the distribution of depth values. It requires $N \times$ parameters to be stored, results on $N \times$ independent trainings, and a single forward pass for each stored configuration at test time.

**Snapshot Ensemble (Snap).** Although the previous method is compelling, obtaining ensembles of neural networks is expensive since it requires carrying out $N$ independent training. An alternative solution [25] consists of obtaining $N$ snapshots out of a single training by leveraging on cyclic learning rate schedules to obtain $C$ pre-converged models. Assuming an initial learning rate $\lambda_0$, we obtain $\lambda_t$ at any training iteration $t$ as a function of the total number of steps $T$ and cycles $C$ as in [25]

$$\lambda_t = \frac{\lambda_0}{2} \cdot \left( \cos \left( \pi \cdot \text{mod} \left( \frac{t}{C}, \frac{T}{C} \right) \right) + 1 \right)$$

Similarly to Boot and Drop, we obtain empirical mean $\mu(d)$ and variance $\sigma^2(d)$ by choosing $N$ out of the $C$ models obtained from a single training procedure.

3.3. Predictive estimation

This category aims at encoding uncertainty by learning a predictive model. This means that at test time these methods produce estimates that are function of network parameters and the input image and thus reason about the current observations, modelling aleatoric heteroscedastic uncertainty [29]. Since often learned from real data distribution, for instance as a function of the distance between the predictions and the ground truth or by maximizing log-likelihood,
these approaches need to be rethought to deal with self-supervised paradigms.

**Learned Reprojection (Repr).** To learn a function over the prediction error employing a classifier is a popular technique used for both stereo [55, 62] and optical flow [43]. However, given the absence of ground truth labels, we cannot apply this approach to self-supervised frameworks seamlessly. Nevertheless, we can drive one output of our network to mimic the behavior of the self-supervised loss function used to train it, thus learning ambiguities affecting the paradigm itself (e.g., occlusions, low texture and more). Indeed, the per-pixel loss signal is supposed to be high when the estimated depth is wrong. Thus, uncertainty $u_{\text{Repr}}$ is trained adding the following term to $L_{SS}$

$$L_{\text{Repr}} = \beta \cdot |u_{\text{Repr}} - \mathcal{F}(\hat{I}, I)|$$

(9)

Since multiple images $I^t$ may be used for supervision, i.e., when combining monocular and stereo, usually for each pixel $q$ the minimum reprojection signal is considered to train the network, thus $u_{\text{Repr}}$ is trained accordingly

$$L_{\text{Repr}}(q) = \beta \cdot |u_{\text{Repr}}(q) - \min_{i \in [0..K]} \mathcal{F}(\hat{I}_i(q), I(q))|$$

(10)

In our experiments, we set $\beta$ to 0.1 and stop $\mathcal{F}$ gradients inside $L_{\text{Repr}}$ for numerical stability. A similar technique appeared in [9], although not evaluated quantitatively.

**Log-Likelihood Maximization (Log).** Another popular strategy [49] consists of training the network to infer mean and variance of the distribution $p(d^*|I, D)$ of parameters $\Theta$. The network is trained by log-likelihood maximization (i.e., negative log-likelihood minimization)

$$\log p(d^*|w) = \frac{1}{N} \sum_q \log p(d^*(q)|\Theta(I, w))$$

(11)

$w$ being the network weights. As shown in [27], the predictive distribution can be modelled as Laplacian or Gaussian respectively in case of L1 or L2 loss computation with respect to $d^*$. In the former case, this means minimizing the following loss function

$$L_{\text{Log}} = \frac{|\mu(d) - d^*|}{\sigma(d)} + \log \sigma(d)$$

(12)

with $\mu(d)$ and $\sigma(d)$ outputs of the network encoding mean and variance of the distribution. The additional logarithmic term discourages infinite predictions for any pixel. Regarding numerical stability [29], the network is trained to estimate the log-variance in order to avoid zero values of the variance. As shown by Klodt and Vedaldi [32], in absence of ground truth $d^*$ one can model the uncertainty $u_{\text{Log}}$ according to photometric matching

$$\log \sigma(d^*) = \min_{i \in [0..K]} \mathcal{F}(\hat{I}_i(q), I(q)) + \log u_{\text{Log}}$$

(13)

Recall that $\mathcal{F}$ is computed over $\pi$ according to Equation 2. Although for stereo supervision this formulation is equivalent to traditional supervision, i.e. $\pi$ is function of a single unknown parameter $d$, in case of monocular supervision this formulation jointly explains uncertainty for depth and pose, both unknown variables in $\pi$. We will show how this approach leads to sub-optimal modelling and how to overcome this limitation with the next approach.

**Self-Teaching (Self).** In order to decouple depth and pose when modelling uncertainty, we propose to source a direct form of supervision from the learned model itself. By training a first network in a self-supervised manner, we obtain a network instance $T$ producing a noisy distribution $d_T$. Then, we train a second instance of the same model, namely $S$, to mimic the distribution sourced from $T$. Typically, teacher-student frameworks [81] applied to monocular depth estimation [52] deploy a complex architecture to supervise a more compact one. In contrast, in our approach the teacher $T$ and the student $S$ share the same architecture and for this reason we refer to it as Self-Teaching (Self). By assuming an L1 loss, we can model for instance negative log-likelihood minimization as

$$L_{\text{Self}} = \frac{|\mu(d_S) - d_T|}{\sigma(d_S)} + \log \sigma(d_S)$$

(14)

We will show how with this strategy i) we obtain a network $S$ more accurate than $T$ and ii) in case of monocular supervision, we can decouple depth from pose and achieve a much more effective uncertainty estimation. Figure 4 summarizes our proposal.

### 3.4. Bayesian estimation

Finally, in Bayesian deep learning [29], the model uncertainty can be explained by marginalizing over all possible $w$ rather than choosing a point estimate. According to Neal
an approximate solution can be obtained by sampling N models and by modelling mean and variance as

\[ p(d^*|I, D) \approx \sum_{i=1}^{N} p(d^*|\Theta(I, w_i)) \]  

If mean and variance are modelled for each \( w_i \) sampling, we can obtain overall mean and variance as reported in [29, 27]

\[ \mu(d) = \frac{1}{N} \sum_{i=1}^{N} \mu_i(d_i) \]  
\[ \sigma^2(d) = \frac{1}{N} \sum_{i=1}^{N} (\mu_i(d_i) - \mu(d))^2 + \sigma_i^2(d_i) \]

The implementation of this approximation is straightforward by combining empirical and predictive methods [29, 27]. Purposely, in our experiments we will pick the best empirical and predictive methods, e.g. combining \text{Boot} and \text{Self} (\text{Boot+Self}).

4. Experimental results

In this section, we exhaustively evaluate self-supervised strategies for joint depth and uncertainty estimation.

4.1. Evaluation protocol, dataset and metrics

At first, we describe all details concerning training and evaluation to ensure full reproducibility. Source code will be available at https://github.com/mattpoggi/mono-uncertainty.

**Architecture and training schedule.** We choose as baseline model Monodepth2 [20], thanks to the code made available and to its possibility to be trained seamlessly according to monocular, stereo, or both self-supervision paradigms. In our experiments, we train any variant of this method following the protocol defined in [20], on batches of 12 images resized to 192 × 640 for 20 epochs starting from pre-trained encoders on ImageNet [12]. Moreover, we always follow the augmentation and training practices described in [20]. Finally, to evaluate \text{Post} we use the same weights made publicly available by the authors. Regarding empirical methods, we set \( N \) to 8 and the number of cycles \( C \) for \text{Snap} to 20. We randomly extract 25% of the training set for each independent network in \text{Boot}. Dropout is applied after convolutions in the decoder only. About predictive models, a single output channel is added in parallel to depth prediction channel.

**Dataset.** We compare all the models on the KITTI dataset [18], made of 61 scenes (about 42K stereo frames) acquired in driving scenarios. The dataset contains images at an average resolution of 375 × 1242 and depth maps from a calibrated LiDAR sensor. Following standards in the field, we deploy the Eigen split [13] and set 80 meters as the maximum depth. For this purpose, we use the improved ground truth introduced in [68], much more accurate than the raw LiDAR data, since our aim is a strict evaluation rather than a comparison with existing monocular methods. Nevertheless, we report results on the raw LiDAR data using Garg’s crop [17] as well in the supplementary material.

**Depth metrics.** To assess depth accuracy, we report for the sake of page limit three out of seven standard metrics\(^1\) defined in [13]. Specifically, we report the absolute relative error (Abs Rel), root mean square error (RMSE), and the amount of inliers (\( \delta < 1.25 \)). We refer the reader to [13] or supplementary material for a complete description of these metrics. They enable a compact evaluation concerning both relative (Abs Rel and \( \delta < 1.25 \)) and absolute (RMSE) errors. Moreover, we also report the number of training iterations (#Trn), parameters (#Par), and forwards (#Fwd) required at testing time to estimate depth. In the case of monocular supervision, we scale depth as in [82].

**Uncertainty metrics.** To evaluate how significant the modelled uncertainties are, we use sparsification plots as in [27]. Given an error metric \( \epsilon \), we sort all pixels in each depth map in order of descending uncertainty. Then, we iteratively extract a subset of pixels (i.e., 2% in our experiments) and compute \( \epsilon \) on the remaining to plot a curve, that is supposed to shrink if the uncertainty properly encodes the errors in the depth map. An ideal sparsification (oracle) is obtained by sorting pixels in descending order of the \( \epsilon \) magnitude. In contrast, a random uncertainty can be modelled as a constant, giving no information about how to remove erroneous measurements and, thus, a flat curve. By plotting the difference between estimated and oracle sparsification, we can measure the Area Under the Sparsification Error (AUSE, the lower the better). Subtracting estimated sparsification from random one enables computing the Area Under the Random Gain (AURG, the higher the better). The former quantifies how close the estimate is to the oracle uncertainty, the latter how better (or worse, as we will see in some cases) it is compared to no modelling at all. We assume Abs Rel, RMSE or \( \delta \geq 1.25 \) (since \( \delta < 1.25 \) defines an accuracy score) as \( \epsilon \).

4.2. Monocular (M) supervision

**Depth.** Table 1a reports depth accuracy for Monodepth2 variants implementing the different uncertainty estimation strategies when trained with monocular supervision. We can notice how, in general, empirical methods fail at improving depth prediction on most metrics, with \text{Drop} having a large gap from the baseline. On the other hand, \text{Boot} and \text{Snap} slightly reduce RMSE. Predictive methods as well produce worse depth estimates, except the proposed \text{Self} method, which improves all the metrics compared to

\(^1\)Results for the seven metrics are available as supplementary material.
Table 1. Quantitative results for monocular (M) supervision.

<table>
<thead>
<tr>
<th>Method</th>
<th>Sup</th>
<th>#Trn</th>
<th>#Par</th>
<th>#Swd</th>
<th>Abs Rel</th>
<th>RMSE</th>
<th>δ ≥ 1.25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monodepth2 [20]</td>
<td>M</td>
<td>1×</td>
<td>1×</td>
<td>1×</td>
<td>0.090</td>
<td>3.942</td>
<td>0.914</td>
</tr>
<tr>
<td>Monodepth2-Post [20]</td>
<td>M</td>
<td>1×</td>
<td>1×</td>
<td>1×</td>
<td>0.088</td>
<td>3.841</td>
<td>0.917</td>
</tr>
<tr>
<td>Monodepth2-Drop</td>
<td>M</td>
<td>1×</td>
<td>1×</td>
<td>N×</td>
<td>0.101</td>
<td>4.146</td>
<td>0.892</td>
</tr>
<tr>
<td>Monodepth2-Boot</td>
<td>N×</td>
<td>1×</td>
<td>0.092</td>
<td>3.821</td>
<td>0.911</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monodepth2-Snap</td>
<td>M</td>
<td>1×</td>
<td>N×</td>
<td>1×</td>
<td>0.091</td>
<td>3.921</td>
<td>0.912</td>
</tr>
<tr>
<td>Monodepth2-Repr</td>
<td>M</td>
<td>1×</td>
<td>1×</td>
<td>0.092</td>
<td>3.936</td>
<td>0.912</td>
<td></td>
</tr>
<tr>
<td>Monodepth2-Log</td>
<td>M</td>
<td>1×</td>
<td>1×</td>
<td>0.091</td>
<td>4.052</td>
<td>0.912</td>
<td></td>
</tr>
<tr>
<td>Monodepth2-Self</td>
<td>M</td>
<td>(1+1)×</td>
<td>1×</td>
<td>0.087</td>
<td>3.826</td>
<td>0.920</td>
<td></td>
</tr>
<tr>
<td>Monodepth2-Boot+Log</td>
<td>M</td>
<td>N×</td>
<td>N×</td>
<td>1×</td>
<td>0.092</td>
<td>3.850</td>
<td>0.917</td>
</tr>
<tr>
<td>Monodepth2-Boot+Self</td>
<td>M</td>
<td>(1+N)×</td>
<td>N×</td>
<td>1×</td>
<td>0.088</td>
<td>3.799</td>
<td>0.918</td>
</tr>
<tr>
<td>Monodepth2-Snap+Log</td>
<td>M</td>
<td>1×</td>
<td>1×</td>
<td>0.092</td>
<td>3.961</td>
<td>0.911</td>
<td></td>
</tr>
<tr>
<td>Monodepth2-Snap+Self</td>
<td>M</td>
<td>(1+1)×</td>
<td>1×</td>
<td>0.088</td>
<td>3.832</td>
<td>0.919</td>
<td></td>
</tr>
</tbody>
</table>

a) Depth evaluation

<table>
<thead>
<tr>
<th>Method</th>
<th>Abs Rel</th>
<th>RMSE</th>
<th>δ ≥ 1.25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monodepth2-Post</td>
<td>0.084</td>
<td>0.030</td>
<td>0.022</td>
</tr>
<tr>
<td>Monodepth2-Drop</td>
<td>0.065</td>
<td>0.040</td>
<td>0.019</td>
</tr>
<tr>
<td>Monodepth2-Boot</td>
<td>0.058</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>Monodepth2-Snap</td>
<td>0.059</td>
<td>0.001</td>
<td>0.002</td>
</tr>
<tr>
<td>Monodepth2-Repr</td>
<td>0.051</td>
<td>0.008</td>
<td>0.013</td>
</tr>
<tr>
<td>Monodepth2-Log</td>
<td>0.039</td>
<td>0.020</td>
<td>0.033</td>
</tr>
<tr>
<td>Monodepth2-Self</td>
<td>0.030</td>
<td>0.026</td>
<td>0.045</td>
</tr>
<tr>
<td>Monodepth2-Boot+Log</td>
<td>0.029</td>
<td>0.028</td>
<td>0.049</td>
</tr>
<tr>
<td>Monodepth2-Boot+Self</td>
<td>0.028</td>
<td>0.021</td>
<td>0.040</td>
</tr>
<tr>
<td>Monodepth2-Snap+Log</td>
<td>0.032</td>
<td>0.022</td>
<td>0.045</td>
</tr>
<tr>
<td>Monodepth2-Snap+Self</td>
<td>0.031</td>
<td>0.026</td>
<td>0.045</td>
</tr>
</tbody>
</table>

b) Uncertainty evaluation

Table 2. Quantitative results for stereo (S) supervision. Evaluation on Eigen split [13] with improved ground truth [68].

<table>
<thead>
<tr>
<th>Method</th>
<th>Sup</th>
<th>#Trn</th>
<th>#Par</th>
<th>#Swd</th>
<th>Abs Rel</th>
<th>RMSE</th>
<th>δ ≥ 1.25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monodepth2 [20]</td>
<td>S</td>
<td>1×</td>
<td>1×</td>
<td>1×</td>
<td>0.088</td>
<td>3.942</td>
<td>0.914</td>
</tr>
<tr>
<td>Monodepth2-Post [20]</td>
<td>S</td>
<td>1×</td>
<td>1×</td>
<td>2×</td>
<td>0.088</td>
<td>3.841</td>
<td>0.917</td>
</tr>
<tr>
<td>Monodepth2-Drop</td>
<td>S</td>
<td>1×</td>
<td>1×</td>
<td>N×</td>
<td>0.101</td>
<td>4.146</td>
<td>0.892</td>
</tr>
<tr>
<td>Monodepth2-Boot</td>
<td>N×</td>
<td>1×</td>
<td>0.092</td>
<td>3.821</td>
<td>0.911</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monodepth2-Snap</td>
<td>S</td>
<td>1×</td>
<td>N×</td>
<td>1×</td>
<td>0.091</td>
<td>3.921</td>
<td>0.912</td>
</tr>
<tr>
<td>Monodepth2-Repr</td>
<td>S</td>
<td>1×</td>
<td>1×</td>
<td>0.092</td>
<td>3.936</td>
<td>0.912</td>
<td></td>
</tr>
<tr>
<td>Monodepth2-Log</td>
<td>S</td>
<td>1×</td>
<td>1×</td>
<td>0.091</td>
<td>4.052</td>
<td>0.912</td>
<td></td>
</tr>
<tr>
<td>Monodepth2-Self</td>
<td>S</td>
<td>(1+1)×</td>
<td>1×</td>
<td>0.087</td>
<td>3.826</td>
<td>0.920</td>
<td></td>
</tr>
<tr>
<td>Monodepth2-Boot+Log</td>
<td>S</td>
<td>N×</td>
<td>N×</td>
<td>1×</td>
<td>0.092</td>
<td>3.850</td>
<td>0.917</td>
</tr>
<tr>
<td>Monodepth2-Boot+Self</td>
<td>S</td>
<td>(1+N)×</td>
<td>N×</td>
<td>1×</td>
<td>0.088</td>
<td>3.799</td>
<td>0.918</td>
</tr>
<tr>
<td>Monodepth2-Snap+Log</td>
<td>S</td>
<td>1×</td>
<td>1×</td>
<td>0.092</td>
<td>3.961</td>
<td>0.911</td>
<td></td>
</tr>
<tr>
<td>Monodepth2-Snap+Self</td>
<td>S</td>
<td>(1+1)×</td>
<td>1×</td>
<td>0.088</td>
<td>3.832</td>
<td>0.919</td>
<td></td>
</tr>
</tbody>
</table>

a) Depth evaluation

<table>
<thead>
<tr>
<th>Method</th>
<th>Abs Rel</th>
<th>RMSE</th>
<th>δ ≥ 1.25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monodepth2-Post</td>
<td>0.010</td>
<td>0.036</td>
<td>0.042</td>
</tr>
<tr>
<td>Monodepth2-Drop</td>
<td>0.013</td>
<td>0.032</td>
<td>0.039</td>
</tr>
<tr>
<td>Monodepth2-Boot</td>
<td>0.023</td>
<td>0.029</td>
<td>0.034</td>
</tr>
<tr>
<td>Monodepth2-Snap</td>
<td>0.028</td>
<td>0.029</td>
<td>0.035</td>
</tr>
<tr>
<td>Monodepth2-Repr</td>
<td>0.024</td>
<td>0.036</td>
<td>0.037</td>
</tr>
<tr>
<td>Monodepth2-Log</td>
<td>0.025</td>
<td>0.036</td>
<td>0.037</td>
</tr>
<tr>
<td>Monodepth2-Self</td>
<td>0.028</td>
<td>0.036</td>
<td>0.037</td>
</tr>
<tr>
<td>Monodepth2-Boot+Log</td>
<td>0.025</td>
<td>0.036</td>
<td>0.037</td>
</tr>
<tr>
<td>Monodepth2-Boot+Self</td>
<td>0.022</td>
<td>0.035</td>
<td>0.035</td>
</tr>
<tr>
<td>Monodepth2-Snap+Log</td>
<td>0.021</td>
<td>0.037</td>
<td>0.037</td>
</tr>
<tr>
<td>Monodepth2-Snap+Self</td>
<td>0.023</td>
<td>0.035</td>
<td>0.037</td>
</tr>
</tbody>
</table>

b) Uncertainty evaluation

the baseline, even when post-processed. Regarding the Bayesian solutions, both Boot and Snap performs worse when combined with Log, while they are always improved by the proposed Self method.

**Uncertainty.** Table 1b resumes performance of modelled uncertainties at reducing errors on the estimated depth maps. Surprisingly, empirical methods rarely perform better than the Post solution. In particular, empirical methods alone fail at performing better than a random chance, except for Drop that, on the other hand, produces much worse depth maps. Predictive methods perform better, with Log and Self yielding the best results. Among them, our method outperforms Log by a notable margin. Combining empirical and predictive methods is beneficial, improving over single choices. In particular, Boot+Self achieves the best overall results.

**Summary.** In general Self, combined with empirical methods, performs better for both depth accuracy and uncertainty modelling when dealing with M supervision, thanks to disentanglement between depth and pose. We believe that empirical methods performance can be ascribed to depth scale, being unknown during training.

### 4.3. Stereo (S) supervision

**Depth.** On Table 2a we show the results of the same approaches when trained with stereo supervision. Again, Drop fails to improve depth accuracy, together with Repr among predictive methods. Boot produces the best improvement, in particular in terms of RMSE. Traditional Log improves this time over the baseline, according to RMSE and δ < 1.25 metrics while, Self consistently improves the baseline on all metrics, although it does not outperform Post, which requires two forward passes.

**Uncertainty.** Table 2b summarizes the effectiveness of modelled uncertainties. This time, only Drop performs worse than Post achieving negative AURG, thus being detrimental at sparsification, while other empirical methods achieve much better results. In these experiments, thanks to the known pose of the stereo setup, Log deals only with depth uncertainty and thus performs extremely well. Self, although allowing for more accurate depth as reported in Table 2a, ranks second this time. Considering Bayesian implementations, again, both Boot and Snap are always improved. Conversely, compared to the M case, Log this time consistently outperforms Self in any Bayesian formulation.

**Summary.** When the pose is known, the gap between Log and Self concerning depth accuracy is minor, with Self performing better when modelling only predictive uncertainty and Log slightly better with Bayesian formulations. For uncertainty estimation, Log consistently performs better. The behavior of empirical methods alone confirms our findings from the previous experiments: by knowing the scale, Boot and Snap model uncertainty much better. In contrast, Drop fails for this purpose.

### 4.4. Monocular-Stereo (MS) supervision

**Depth.** Table 3a reports the behavior of depth accuracy when monocular and stereo supervisions are combined. In this case, only Self consistently outperforms the baseline and is competitive with Post, which still requires two forward passes. Among empirical methods, Boot is the most effective. Regarding Bayesian solutions, those using Self
are, in general, more accurate on most metrics, yet surprisingly worse than Self alone.

**Uncertainty.** Table 3b shows the performance of the considered uncertainties. The behavior of all variants is similar to the one observed with stereo supervision, except for Log and Self. We can notice that Self outperforms Log, similarly to what observed with M supervision. It confirms that pose estimation drives Log to worse uncertainty estimation, while Self models are much better thanks to the training on proxy labels produced by the Teacher network. Concerning Bayesian solutions, in general, Boot and Snap are improved when combined with both Log and Self, with Self combinations typically better than their Log counterparts and equivalent to standalone Self.

**Summary.** The evaluation with monocular and stereo supervision confirms that when the pose is estimated alongside with depth, Self proves to be a better solution compared to Log and, in general, other approaches to model uncertainty. Finally, empirical methods alone behave as for experiments with stereo supervision, confirming that the knowledge of the scale during training is crucial to the proper behavior of Drop, Boot and Snap.

### 4.5. Sparsification curves

In order to further outline our findings, we report in Figure 5 the RMSE sparsification error curves, averaged over the test set, when training with M, S or MS supervision. The plots show that methods leveraging on Self (blue) are the best to model uncertainty when dealing with pose estimation, *i.e.* M and MS, while those using Log (green) are better when training on S. We report curves for Abs Rel and

![Figure 5. Sparsification Error curves. From left to right, average RMSE with M, S and MS supervisions. Best viewed with colors.](image-url)

### 4.6. Supplementary material

For the sake of the pages limit, we report more details about the experiments shown so far in the supplementary material. Specifically, i) complete depth evaluation with all seven metrics defined in [13], ii) depth and uncertainty evaluation with reduced depth range to 50 meters, iii) evaluation assuming the raw LiDAR data as ground truth, for compliance with previous works [20] and iv) sparsification curves for all metrics. We also provide additional qualitative results in the form of images and a video sequence, available at [www.youtube.com/watch?v=bxVPXqf4zt4](http://www.youtube.com/watch?v=bxVPXqf4zt4).

### 5. Conclusion

In this paper, we have thoroughly investigated for the first time in literature uncertainty modelling in self-supervised monocular depth estimation. We have reviewed and evaluated existing techniques, as well as introduced a novel Self-Teaching (Self) paradigm. We have considered up to 11 strategies to estimate the uncertainty on predictions of a depth-from-mono network trained in a self-supervised manner. Our experiments highlight how different supervision strategies lead to different winners among the considered methods. In particular, among empirical methods, only Dropout sampling performs well when the scale is unknown during training (M), while it is the only one failing when scale is known (S, MS). Empirical methods are affected by pose estimation, for which log-likelihood maximization gives sub-optimal results when the pose is unknown (M, MS). In these latter cases, potentially the most appealing for practical applications, the proposed Self technique results in the best strategy to model uncertainty. Moreover, uncertainty estimation also improves depth accuracy consistently, with any training paradigm.

**Acknowledgement.** We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.
References


[19] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In CVPR, 2017. 1, 2, 4


[41] Chenxu Luo, Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, Ram Nevatia, and Alan Yuille. Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding. PAMI, 2019. 2


[54] Matteo Poggi and Stefano Mattoccia. Learning from scratch a confidence measure. In BMVC, 2016. 3


