Robust Regression for Monocular Depth Estimation

Julian Lienen JULIAN.LIENEN@UPB.DE
Paderborn University, Germany

Nils Nommensen NILS.NOMMENSEN@TIB.EU
Ralph Ewerth RALPH.EWERTH@TIB.EU
L3S Research Center, Leibniz University Hannover and TIB Hannover, Germany

Eyke Hüllermeier EYKE@IFI.LMU.DE
University of Munich (LMU), Germany

Editors: Vineeth N Balasubramanian and Ivor Tsang

Abstract

Learning accurate models for monocular depth estimation requires precise depth annotation as e.g. gathered through LiDAR scanners. Because the data acquisition with sensors of this kind is costly and does not scale well in general, less advanced depth sources, such as time-of-flight cameras, are often used instead. However, these sensors provide less reliable signals, resulting in imprecise depth data for training regression models. As shown in idealized environments, the noise produced by commonly used RGB-D sensors violates standard statistical assumptions of regression methods, such as least squares estimation. In this paper, we investigate whether robust regression methods, which are more tolerant toward violations of statistical assumptions, can mitigate the effects of low-quality data. As a viable alternative to established approaches of that kind, we propose the use of so-called superset learning, where the original data is replaced by (less precise but more reliable) set-valued data. To evaluate and compare the methods, we provide an extensive empirical study on common benchmark data for monocular depth estimation. Our results clearly show the superiority of robust variants over conventional regression.

Keywords: Robust regression, monocular depth estimation, superset learning, data imprecisiation

1. Introduction

In many computer vision applications, such as 3D scene understanding or autonomous driving, the estimation of depth in visual perception is of crucial importance. Often, signals are only observed in the form of monocular images used as input to predict pixel-wise depth. Due to its ill-posed nature, the estimation of depth based on single images is a complex task, which has recently been tackled by machine learning methods, more specifically by deep neural networks trained on large amounts of data samples.

Various data sets provide single images in different scenes along with depth maps gathered from sensors, which are made available as supervision for training monocular depth estimation models. As the acquisition of data with highly accurate depth sensors, e.g., through laser-based LiDAR systems, is costly, most high-volume metric depth data sets were constructed based on less accurate RGB-D sensors, such as infrared (IR) or time-of-flight (TOF) cameras. As a prominent sensor of this kind, Kinect V1 has been employed to

construct the widely used NYUD-v2 data set (Silberman et al., 2012), especially for depth in indoor scenes.

Despite their popularity and applicability, data sets constructed with such sensors incorporate a considerable degree of noise. As studied in idealized environments, the distortion of commonly used sensors increases with higher spatial depth (Khoshelham and Elberink, 2012; Wasenmüller and Stricker, 2016; Ahn et al., 2019). While this can also be observed for laser-based sensors (Rosenberger et al., 2018), the problem is especially severe for less sophisticated IR or TOF sensors. As a prominent example, studies analyzing Kinect V1 sensors yield an exponentially increasing standard deviation for higher depth values to be measured, while an increasing offset of the sensed value to the underlying true depth value can be observed (Nguyen et al., 2012; Wasenmüller and Stricker, 2016). Moreover, due to physical properties of the sensors, e.g., interference of emitted rays, the error terms for each individual data term can not be assumed to be independent of other observed signals.

These properties are in conflict with standard statistical model assumptions of conventional regression methods, such as least squares that has also been considered as an optimization criterion in the domain of depth estimation (Carvalho et al., 2018). For instance, it is often assumed to observe noise with constant variance (*homoscedasticity*), and that errors are independent between samples (no *autocorrelation*). Provided such assumptions, traditional methods deliver efficient estimators with several appealing asymptotic guarantees (Dougherty, 2011).

Obviously, these assumptions are violated for most non-synthetic depth estimation data sets. Although several alternatives were suggested to address the aforementioned issues by weaker model assumptions (e.g., as in (Barron, 2019; Irie et al., 2019; Ranftl et al., 2020)), the explicit consideration of robustness in the modeling of monocular depth estimation has received rather little attention so far. This work aims to fill this gap by providing an overview of existing robust regression methods and investigating their effectiveness in the context of depth estimation.

In addition to established methods for robust regression, we also propose to realize the recent idea of "data imprecisiation" to achieve robustness in depth estimation. Here, precise but possibly distorted (biased or noisy) data is turned into imprecise (set-valued) but probably more correct and reliable data, and a model is then trained on the modified data using so-called superset learning (Hüllermeier, 2014).

An exhaustive empirical evaluation demonstrates the effectiveness of robust variants over conventional regression methods on popular depth estimation benchmarks, and especially confirms the adequacy of the superset modelling approach in cases of erroneous and misleading training information.

2. Robust Regression

In this section, we survey related work on robust regression, specifically focusing on losses that have been used in the domain of depth estimation.

2.1. Standard Regression Methods

In the setting of regression, one commonly assumes a stochastic dependency of the form $y = f(\boldsymbol{x}) + \epsilon$, i.e., samples $y \in \mathcal{Y} = \mathbb{R}$ of the target (output) variable are functions of the

input (instance) $\boldsymbol{x} \in \mathcal{X}$ afflicted with (additive) random errors $\varepsilon \in \mathbb{R}$. In the context of depth estimation, instances $\boldsymbol{x}$ could be descriptions of the pixels of an image, and outputs y the corresponding depth values. Given training data in the form of a set of input/output pairs $(\boldsymbol{x}_i, y_i)$, $i = 1, \ldots, n$, the task is to learn a function $\hat{f} : \mathcal{X} \to \mathcal{Y}$ that allows for predicting the target value for any query instance given as an input. Typically, this is accomplished by finding a function that minimizes a certain loss $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ on the training data, i.e., that minimizes the training error $\sum_{i=1}^{n} \mathcal{L}(y_i, \hat{f}(\boldsymbol{x}_i))$ (or a regularized version of this error).

If such assumptions are violated in practice, and outcomes are observed with a high degree of noise, one may want to weaken the assumptions, leading to more robust estimators. Long-standing research has been conducted on achieving robustness in classical statistics, leading to several approaches that improve estimation performance on data that does not comply with strong model assumptions. For instance, generalized versions of least squares regression have been suggested to cope with heteroscedasticity (Kariya and Kurata, 2004), e.g., by weighting the residuals according to the inverse of the variance of the error. Likewise, alternative loss functions, for example the absolute ($\mathcal{L}_1$) instead of the squared ($\mathcal{L}_2$) error, have been considered to alleviate sensitivity to outliers. Often, however, the minimization of such losses comes with other issues of practical relevance, such as unstable or ambiguous solutions (Dodge, 1987).

A famous class of extremum estimation methods are the so-called M-estimators (Huber, 1981), which generalize the idea of maximum likelihood estimation by providing an interface to inject more robust cost functions as optimization criterion. As one of such functions, Huber (Huber, 1981) introduced a robust loss that combines the squared and the absolute loss to diminish the sensitivity to outliers.

2.2. Robustness in Depth Estimation

In the domain of (supervised) monocular depth estimation, a plethora of different loss functions has been suggested to induce regression models, ranging from $\mathcal{L}_1$- (Ma and Karaman, 2018; Ranftl et al., 2020) and $\mathcal{L}_2$-based (Carvalho et al., 2018; Ranftl et al., 2020) losses to model-specific measures (Kendall and Gal, 2017; Wu et al., 2019; Bhat et al., 2021). Also, several loss augmentations have been proposed, e.g., to consider smoothness in the prediction (Li and Snavely, 2018) or to treat targets in a different representation (Fu et al., 2018; Li and Snavely, 2018). Although ablation studies often compare loss functions and their effects (e.g., as in (Carvalho et al., 2018; Ranftl et al., 2020)), to the best of our knowledge, an explicit investigation of the robustness of losses in the context of depth estimation is still missing.

As one of the earlier approaches to achieve robustness, the previously mentioned Huber-loss has been applied in the domain of depth estimation, although in a reversed form (Laina et al., 2016; Carvalho et al., 2018). Its original (robust) form as used for depth estimation is given by

$$\mathcal{L}_{\text{Huber}}(y, \hat{y}) := \begin{cases} \frac{(y-\hat{y})^2 + c^2}{2c} & \text{if } |y - \hat{y}| \leq c \\ |y - \hat{y}| & \text{otherwise} \end{cases}, \tag{1}$$

where y is the observed value, $\hat{y}$ the model prediction, and the parameter c is typically defined as 20% of the maximum residual in each batch calculation. The Huber loss inherits

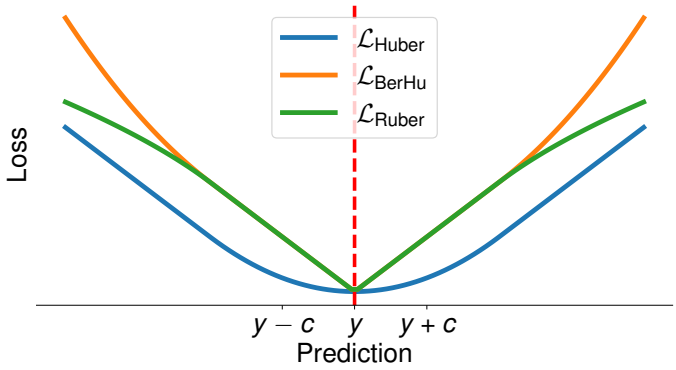

Figure 1: Variants of the Huber loss as used in the domain of monocular depth estimation.

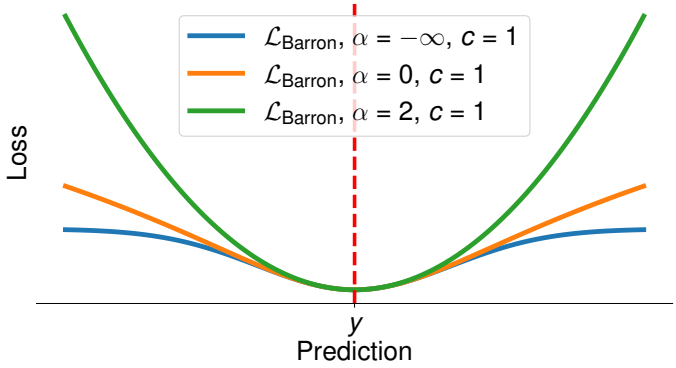

Figure 2: Special cases of $\mathcal{L}_{\text{Barron}}$ as presented in (Barron, 2019).

the advantage of $\mathcal{L}_1$ to deemphasize the influence of outliers while overcoming the non-differentiability of this loss at zero. As the more popular method in the depth estimation domain, let us denote the BerHu loss as the reversed version of $\mathcal{L}_{\text{Huber}}$ by $\mathcal{L}_{\text{BerHu}}$.

The loss formulation has also been adopted by smoothening the $\mathcal{L}_1$ part for further robustness, leading to the so-called Ruber loss (Irie et al., 2019), which is defined as

$$\mathcal{L}_{\text{Ruber}}(y, \hat{y}) := \begin{cases} |y - \hat{y}| & \text{if } |y - \hat{y}| \leq c \\ \sqrt{2c|y - \hat{y}| - c^2} & \text{otherwise} \end{cases} . \tag{2}$$

In their work, the authors show improved robustness, along with the optimization of the parameter c in a data-driven manner. Fig. 1 illustrates the Huber-like losses as used within the domain of depth estimation.

As one loss coming from a related field, namely flow prediction, the so-called "generalized Charbonnier" loss (Sun et al., 2010) with a smoothed $\mathcal{L}_1$ loss term as special case showed promising robustness properties for the problem of depth estimation (Chen and Koltun, 2014). Closely related to this, Barron (2019) suggests a more expressive robust loss variant,

which even extends the Charbonnier loss. It is given by

$$\mathcal{L}_{\text{Barron}}(y, \hat{y}) := \frac{|\alpha - 2|}{\alpha} \left(\left(\frac{((y - \hat{y})/c)^2}{|\alpha - 2|} + 1 \right)^{\alpha/2} - 1 \right), \tag{3}$$

where $\alpha \in \mathbb{R}$ and $c \in \mathbb{R}_+$ are hyperparameters to control the robustness and scale respectively. Special cases of the loss are depicted in Fig. 2. Interestingly, this loss delivers the generalized Charbonnier loss, as well as $\mathcal{L}_2$ and a smoothed version of $\mathcal{L}_1$ as special cases.

Current state-of-the-art methods often employ a scale-invariant version of the $\mathcal{L}_2$ loss in log-space (Eigen et al., 2014), which, for a set of n observations $y_1, \ldots, y_n$, is given by

$$\mathcal{L}_{\text{SIError}}(y, \hat{y}) := \frac{1}{n} \sum_{i=1}^{n} g_i^2 - \frac{\lambda}{n^2} \left(\sum_{i=1}^{n} g_i \right)^2, \tag{4}$$

where g_i is the residual of the i^{th} instance in log space, i.e., $g_i = \log y_i - \log y_i^*$ and $\lambda \in [0, 1]$ is a hyperparameter. This variant has further been augmented by an additional scaling parameter α (Lee et al., 2019; Bhat et al., 2021). We refer to the scaled variant of this loss as $\mathcal{L}_{\text{ScaledSIError}}$. Although not specifically designed to cope with outliers, its depth interpretation in log space diminishes the severity of heteroscedasticity in least squares optimization, and it has been shown to yield state-of-the-art generalization performance (Bhat et al., 2021).

As another robust loss formulation, this time applied in the disparity space, Ranftl et al. (2020) propose a loss variant that trims an $\mathcal{L}_1$ loss by disregarding the 20% largest residuals in each image, which we refer to as $\mathcal{L}_{\text{trim}}$. This is in contrast to M-estimators as the weighted least squares method, where residuals with a high variance are down-weighted.

3. Superset Learning

As an alternative to cope with low-quality data, we advocate the idea of "data imprecisiation", which in turn is grounded in the framework of superset learning. In the following, we give a brief introduction to superset learning in general, followed by two concrete proposals for robust depth estimation.

3.1. Background on Superset Learning

Recall that, in learning a depth estimator given images with their corresponding depth maps, one typically considers pixels as individual training instances attached with single values from a target space $\mathcal{Y}$, in the case of depth regression usually with $\mathcal{Y} = \mathbb{R}_+$. Given this ground truth data, the task is to learn a model (hypothesis) predicting values $\hat{y} \in \mathcal{Y}$ that fit the training data as much as possible (but not too much to avoid overfitting). To measure the optimality of the prediction, losses of the form $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ are employed, like those presented before.

In superset learning, we consider the case where data is not necessarily observed precisely. Instead of precise outcomes $y \in \mathcal{Y}$ provided as supervision, we only assume that *subsets* $Y \subseteq \mathcal{Y}$ of the output space are given as training information. Thus, a single observation is of the form $(\boldsymbol{x}, Y) \in \mathcal{X} \times 2^{\mathcal{Y}}$. The set-valued data is supposed to cover the

underlying precise but unobserved data in the sense that $y \in Y$ (hence the name "superset learning").[1]

Provided data of that kind, Hüllermeier (2014) proposed an approach to superset learning motivated by the idea of performing model identification and "data disambiguation" at the same time. To this end, the underlying loss function $\mathcal{L} : \mathcal{Y}^2 \to \mathbb{R}_+$ is extended to the *optimistic superset loss* (OSL) $\mathcal{L}^* : 2^{\mathcal{Y}} \times \mathcal{Y} \longrightarrow \mathbb{R}_+$ defined by the map

$$(Y, \hat{y}) \mapsto \min \big\{ \mathcal{L}(y, \hat{y}) \,|\, y \in Y \big\}. \tag{5}$$

More recently, the same loss has also been introduced under the notion of *infimum loss* (Cabannes et al., 2020). Superset learning then seeks to perform generalized risk minimization, i.e., to minimize the OLS loss (or a regularized version thereof) instead of the original loss $\mathcal{L}$ on the training data.

3.2. From Set-valued to Fuzzy Data

The OSL (5) can be generalized further to the case where data is characterized in terms of fuzzy sets (Klir and Folger, 1988). The latter generalize conventional sets in the sense of allowing gradual membership of elements, where the degree of membership is typically specified in terms of a real number in the unit interval. Thus, a fuzzy subset $\tilde{Y}$ of $\mathcal{Y}$ can be identified with a membership function of the form $\tilde{Y} : \mathcal{Y} \longrightarrow [0, 1]$, where $\tilde{Y}(y) = 1$ indicates full membership of y, $\tilde{Y}(y) = 0$ no membership, and $0 < \tilde{Y}(y) < 1$ that y belongs to the fuzzy set to a certain degree (Klir and Folger, 1988). The fuzzy-version of the OSL loss (which we refer to as FOSL) is obtained as a generalization of $\mathcal{L}^*$, using a reduction scheme based on a standard level-cut representation of fuzzy sets:

$$\begin{aligned} \mathcal{L}^{**} : \mathcal{F}(\mathcal{Y}) \times \mathcal{Y} &\longrightarrow \mathbb{R}_+ \,, \\ \big(\tilde{Y}, \hat{y}\big) &\mapsto \int_0^1 \mathcal{L}^* \Big([\tilde{Y}]_\alpha, \hat{y} \Big) \, d\alpha \,, \end{aligned} \tag{6}$$

where $\mathcal{F}(\mathcal{Y})$ denotes the set of all fuzzy subsets of $\mathcal{Y}$ and $[\tilde{Y}]_\alpha := \{y \,|\, \tilde{Y}(y) \geq \alpha\}$ is the α-cut of $\tilde{Y}$.

3.3. Data Imprecisiation

In addition to learning from genuinely imprecise data, the framework of superset learning can also be used for learning from standard (precise) data, which — via a process of "imprecisiation" — is deliberately turned into imprecise data (Hüllermeier, 2014). Different effects can be achieved in this way. In particular, data imprecisiation offers a means to control the influence of individual observations on the overall result of the learning process: the more imprecise an observation is made, the less it will influence the model induced from the data (Lienen and Hüllermeier, 2021).

Indeed, the optimistic superset loss (5), and likewise the fuzzy version (6), is a relaxation of the original loss $\mathcal{L}$ in the sense that $\mathcal{L}^* \leq \mathcal{L}$. More specifically, the larger the set Y, the smaller the loss: $Y \supseteq Y'$ implies $\mathcal{L}^*(Y, \hat{y}) \leq \mathcal{L}^*(Y', \hat{y})$ for all $\hat{y} \in \mathcal{Y}$. Thus, the loss

1. Note that the precise data y may already be corrupted with noise.

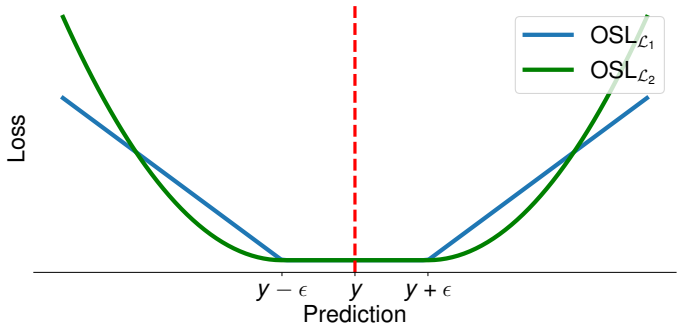

Figure 3: ϵ-insensitive OSL variants for interval data.

$\mathcal{L}(y, \hat{y})$ incurred for a prediction $\hat{y}$ can be weakened by replacing the original observation y with a (fuzzy) subset around y, and the larger the subset, the smaller the loss. Therefore, "imprecisiating" a data point by replacing the original (precise) observation y with a (fuzzy) set-valued outcome Y can be seen as a means for reducing the influence of possibly noisy or unreliable data, and hence for making learning more robust. In the following, we shall discuss two concrete approaches of that kind in the context of regression for depth estimation.

3.4. Interval Data

As already said, the values produced by depth sensors are often quite noisy, and the assumptions of a precise noise model do normally not apply. A somewhat crude but robust alternative is to model the information about the underlying true depth in terms of a tolerance interval around the precise measurement y. Thus, the learning algorithm is merely provided with the information that the sought depth is most likely an element of this interval. Depending on the length of the interval, this information might be relatively weak.

More importantly, however, it is also most likely *correct*. Therefore, compared to more precise but presumably wrong information, it is less likely to bias the learner in a wrong direction. In fact, because the loss is 0 as long as the learner predicts any value inside the interval, it is completely free to choose the value that appears most plausible (in light of the other observations and its underlying model assumptions), without incurring any penalty. As confirmed by empirical studies (Cabannes et al., 2020), this provides the learner with an opportunity to disambiguate the data and increases robustness toward misleading observations.

More specifically, we model the data in terms of ϵ-intervals $Y = [y-\epsilon, y+\epsilon]$. Interestingly, we thus establish a close connection to the well-known method of support vector regression (SVR) (Schölkopf and Smola, 2001). In fact, the OSL extension of the $\mathcal{L}_1$ loss obtained for data of that kind exactly coincides with the ϵ-insensitive loss used in SVR. Fig. 3 depicts this loss as well as the OSL extension of the $\mathcal{L}_2$ loss.

In the approach realized in this paper, all intervals are centered around the original observations and share the same length 2ϵ. We consider ϵ as a hyperparameter that is tuned on a validation set. Let us note, however, that intervals could in principle also be customized for each observation individually. This way, different types of domain knowledge

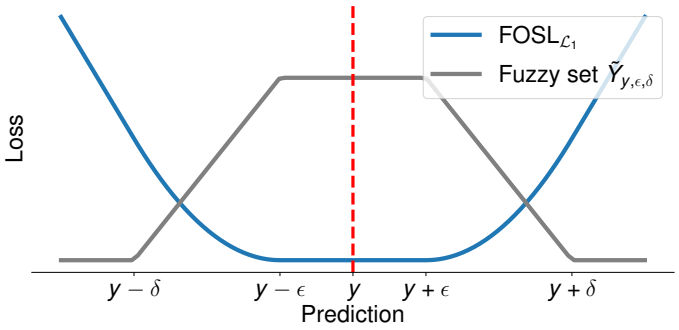

Figure 4: The FOSL variant based on $\mathcal{L}_1$ for a trapezoidal fuzzy superset $\tilde{Y}_{y,\epsilon,\delta}$.

could be incorporated, for example that measurements in a certain region of an image are more reliable than in another region, or that some measurement have a stronger tendency to over- than to underestimate the true depth.

3.5. Fuzzy data

Going beyond a distinction between plausible and implausible values, as purported by an interval, fuzzy sets allow for modeling data in a more elaborate way. As an interesting special case, the Huber loss is reproduced as the OSL-extension of the $\mathcal{L}_1$ loss when replacing precise measurements y by *triangular* fuzzy sets $\tilde{Y}_{y,\delta}(z) = \max\{0, 1 - |y - z|/\delta\}$.

Even more appropriate for the case of robust depth estimation is the FOSL loss obtained for *trapezoidal* fuzzy sets of the form

$$\tilde{Y}_{y,\epsilon,\delta}(z) = \begin{cases} \frac{z-y+\delta}{\delta-\epsilon} & \text{if } y-\delta \leq z \leq y-\epsilon \\ 1 & \text{if } y-\epsilon \leq z \leq y+\epsilon \\ \frac{y+\delta-z}{\delta-\epsilon} & \text{if } y+\epsilon \leq z \leq y+\delta \\ 0 & \text{otherwise} \end{cases}, \tag{7}$$

which combine attenuation properties of the Huber-loss with the relaxation effects produced by the ϵ-insensitivity of the SVR loss. More specifically, by incorporating $\tilde{Y}_{y,\epsilon,\delta}$ in (5) we obtain

$$\mathcal{L}^{**}\big(\tilde{Y}_{y,\epsilon,\delta}, \hat{y}\big) := \begin{cases} 0 & \text{if } \hat{y} \in [y-\epsilon, y+\epsilon] \\ \frac{(y\pm\epsilon-\hat{y})^2}{2(\delta-\epsilon)} & \text{if } \hat{y} \in (y\pm\delta, y\pm\epsilon) \\ |y \pm \epsilon - \hat{y}| - \frac{\delta-\epsilon}{2} & \text{otherwise} \end{cases}, \tag{8}$$

where $\epsilon, \delta \in \mathbb{R}_+$ with $\epsilon \geq \delta$ are hyperparameters. Similar to the interval-based loss, ϵ and δ can be optimized on validation data. Fig. 4 shows the resulting loss function.

4. Evaluation

To demonstrate the effectiveness of robust methods for depth estimation, we conduct an extensive empirical evaluation on common indoor benchmark data. First, we give an overview over the data, baselines, metrics, and implementation details, followed by the presentation of the results.

4.1. Experimental Settings

4.1.1. Datasets

In our studies, we consider two sources for training a depth predictor. First, we use *NYUD-v2* (Silberman et al., 2012) as a homogeneous indoor[2] data set based on the Kinect V1 sensor, which has been studied broadly and for which approximations of the sensor noise are provided (e.g., as in (Nguyen et al., 2012; Wasenmüller and Stricker, 2016)). Second, as a data set that unifies multiple sources with individual error terms, we consider *SunRGBD* (Song et al., 2015) as an additional heterogeneous source to learn from. This data set uses four different sensors, namely Kinect V1, Kinect V2, RealSense, and Xtion (cf. (Song et al., 2015) for more detailed descriptions).

For *NYUD-v2*, we use a subset of 10k preprocessed instances as also used in (Bhat et al., 2021). For training, we rescale each input image and depth map to the size of 224×224, while we evaluate on the Eigen split of 654 test samples using the commonly applied cropping in the original resolution (480×640).

To train models on *SunRGBD*, we use the original training and test splits as provided by the authors of the data set. While the training set consists of $10,355$ indoor RGB-D images, the test split comprises 2860 images. The resolutions are kept the same as for *NYUD-v2*.

Since both data sets involve noisy depth sensors, models reconstructing the sensor noise observed in the training data benefit from the evaluation on the corresponding test sets when constructed on the same base. Rather, we aim to measure the model performances on the basis of highly accurate signals. To this end, we evaluate the induced models on the LiDAR-based dataset *iBims-1* (Koch et al., 2018) and *DIODE* (Vasiljevic et al., 2019).

iBims-1 makes use of a digital single-lens reflex camera attached with a high-precision laser scanner to acquire images along with their pixel-wise depth, approximately matching the depth value distribution of *NYUD-v2*. The data set consists of 100 indoor RGB-D image of resolution 480×640. Within our studies, we use this data set as validation set to optimize model hyperparameters.

For the final model assessment, we use the provided indoor validation set of *DIODE*, consisting of 335 high-quality RGB-D images of resolution 768×1024, which provides a diverse set of indoor scenes used to measure the generalization performance of the assessed models. To compute metrics on the test data, we upscaled all model predictions to the original size.

4.1.2. Baselines

As baselines, we consider the loss functions discussed before. That is, we depart from $\mathcal{L}_1$ and $\mathcal{L}_2$ as the most obvious choices to train regression models. Beyond that, as used within the domain of depth estimation, we consider the Huber-loss variants $\mathcal{L}_{\text{Huber}}$, $\mathcal{L}_{\text{BerHu}}$, and $\mathcal{L}_{\text{Ruber}}$. Moreover, $\mathcal{L}_{\text{Barron}}$ and $\mathcal{L}_{\text{trim}}$ as losses explicitly approaching robustness are also included. As a loss used to train current SOTA models, we further evaluate models learned with $\mathcal{L}_{\text{ScaledSIError}}$.

2. Here, we focus on indoor data sets as the test data in such scenes is usually more precise compared to outdoor scenery.

Apart from that, in order to seek to improve conventional $\mathcal{L}_2$ optimization, we apply the weighted least squares criterion, which we refer to as $\mathcal{L}_{\text{WeightedL2}}$. Here, we use an approximation of the standard deviation of the Kinect V1 sensors as provided in (Nguyen et al., 2012), namely $\sigma(x) = 0.0012 + 0.0012(x - 0.4)^2$ for weighting.

To demonstrate the effectiveness of the superset modelling approaches, we provide results for both the interval- and fuzzy set-based modelling approach. For the former, we investigate variants based on $\mathcal{L}_1$ and $\mathcal{L}_2$, denoted by $\text{OSL}_{\mathcal{L}_1}$ and $\text{OSL}_{\mathcal{L}_2}$, respectively. For the latter, we consider the FOSL variant on the basis of $\mathcal{L}_1$ as $\text{FOSL}_{\mathcal{L}_1}$.

4.1.3. Metrics

In order to measure the performance of the individual models, we present the results for 6 regression methods as commonly reported in the field of depth estimation. The error metrics are defined for ground truth depth values $y \in \mathbb{R}_+$ and model predictions $\hat{y} \in \mathbb{R}_+$ for an image as follows:

- Absolute relative error (REL): $\frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i}$
- Average $\log_{10}$ error: $\frac{1}{n}\sum_{i=1}^{n}|\log_{10}(y_i) - \log_{10}(\hat{y}_i)|$
- Root mean squared error (RMS): $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$
- Threshold accuracies δ_i: percentage of $\hat{y}$ s.t. $\max\left(\frac{y}{\hat{y}}, \frac{\hat{y}}{y}\right) = \delta < 1.25^i$

The final results are averaged over all test images. We provide results for more metrics in the supplement.

4.1.4. Implementation Details

For our experiments, we use a simple U-Net architecture employing an EfficientNetB0 encoder pretrained on ImageNet. For the decoder part, we use a stack of repeating convolutional, BatchNormalization, ReLU, and bilinear upsampling layers. In total, the model comprises approximately 15 million parameters and is kept the same across all experiments.

To provide a fair comparison of all losses incorporating several hyperparameters, we optimized hyperparameters for both the optimizer (Adam in our case) and the individual losses within a random search with 20 trials. Each model is trained for 25 epochs with a batch size of 16. As mentioned before, *iBims-1* was used to calculate the validation scores. The model providing the lowest validation score throughout the runs was considered for the final testing. For statistical significance of the results, we conducted each experiment three times with different seeds.

To allow for reproducing our results, a more comprehensive overview about implementation details and a detailed model description is provided in the supplement.

4.2. Homogeneous Depth Sensor: NYUD-v2

In the first experiment, we assess models trained on subsets of *NYUD-v2*. As discussed before, this data set annotated by Kinect V1 depths incorporates a relatively high degree of noise and violates classical statistical assumptions. To assess the robustness of the different

Table 1: Averaged results and standard deviations on models trained on subsets of various sizes of *NYUD-v2* on *DIODE* and *NYUD-v2*. The best results indicated in bold per number of instances and metric.

# Insts.	Loss	*DIODE*						*NYUD-v2*		
		REL (↓)	$\log_{10}$ (↓)	RMS (↓)	δ_1 (↑)	δ_2 (↑)	δ_3 (↑)	REL (↓)	RMS (↓)	δ_1 (↑)
2k	$\mathcal{L}_2$	0.492 ± 0.030	0.223 ± 0.001	1.839 ± 0.015	0.316 ± 0.007	0.547 ± 0.006	0.703 ± 0.007	0.375 ± 0.045	1.015 ± 0.091	0.463 ± 0.039
	$\mathcal{L}_1$	0.463 ± 0.025	0.228 ± 0.001	1.891 ± 0.035	0.306 ± 0.005	0.534 ± 0.001	0.694 ± 0.002	0.327 ± 0.024	0.934 ± 0.043	0.512 ± 0.014
	$\mathcal{L}_{\text{Huber}}$	0.433 ± 0.021	0.230 ± 0.016	1.873 ± 0.084	0.293 ± 0.031	0.542 ± 0.030	0.703 ± 0.031	0.281 ± 0.004	0.826 ± 0.005	0.554 ± 0.003
	$\mathcal{L}_{\text{BerHu}}$	0.440 ± 0.016	0.225 ± 0.003	1.854 ± 0.012	0.304 ± 0.008	0.548 ± 0.009	0.713 ± 0.003	0.284 ± 0.017	0.851 ± 0.021	0.553 ± 0.019
	$\mathcal{L}_{\text{Ruber}}$	0.434 ± 0.008	0.232 ± 0.008	1.878 ± 0.034	0.297 ± 0.016	0.536 ± 0.022	0.700 ± 0.022	0.285 ± 0.024	0.835 ± 0.044	0.571 ± 0.021
	$\mathcal{L}_{\text{Barron}}$	0.450 ± 0.010	0.224 ± 0.002	1.850 ± 0.018	0.313 ± 0.010	0.553 ± 0.005	0.719 ± 0.006	0.310 ± 0.024	0.883 ± 0.057	0.531 ± 0.038
	$\mathcal{L}_{\text{trim}}$	0.451 ± 0.026	0.229 ± 0.006	1.878 ± 0.026	0.328 ± 0.004	0.553 ± 0.001	0.702 ± 0.008	0.362 ± 0.034	1.045 ± 0.151	0.481 ± 0.015
	$\mathcal{L}_{\text{ScaledSIError}}$	**0.427** ± 0.001	0.225 ± 0.007	1.825 ± 0.024	0.300 ± 0.020	0.554 ± 0.019	**0.721** ± 0.009	**0.258** ± 0.024	**0.763** ± 0.046	**0.613** ± 0.031
	$\mathcal{L}_{\text{WeightedL2}}$	0.486 ± 0.026	0.221 ± 0.001	1.837 ± 0.007	0.320 ± 0.007	0.551 ± 0.003	0.708 ± 0.003	0.371 ± 0.035	1.007 ± 0.068	0.465 ± 0.025
	$\text{OSL}_{\mathcal{L}_1}$	0.454 ± 0.028	**0.216** ± 0.005	**1.803** ± 0.028	**0.332** ± 0.017	**0.562** ± 0.014	0.712 ± 0.006	0.325 ± 0.035	0.867 ± 0.062	0.532 ± 0.031
	$\text{OSL}_{\mathcal{L}_2}$	0.472 ± 0.057	0.219 ± 0.007	1.815 ± 0.032	0.317 ± 0.020	0.548 ± 0.016	0.703 ± 0.012	0.361 ± 0.057	0.981 ± 0.093	0.495 ± 0.037
	$\text{FOSL}_{\mathcal{L}_1}$	0.448 ± 0.005	0.229 ± 0.008	1.875 ± 0.045	0.302 ± 0.018	0.535 ± 0.016	0.701 ± 0.005	0.282 ± 0.012	0.832 ± 0.025	0.561 ± 0.016
10k	$\mathcal{L}_2$	0.446 ± 0.007	0.227 ± 0.004	1.859 ± 0.011	0.307 ± 0.008	0.545 ± 0.006	0.706 ± 0.006	0.301 ± 0.015	0.876 ± 0.025	0.525 ± 0.003
	$\mathcal{L}_1$	0.432 ± 0.004	0.228 ± 0.012	1.851 ± 0.052	0.308 ± 0.019	0.548 ± 0.023	0.709 ± 0.020	0.252 ± 0.019	0.741 ± 0.035	0.625 ± 0.018
	$\mathcal{L}_{\text{Huber}}$	0.441 ± 0.008	0.231 ± 0.012	1.868 ± 0.041	0.313 ± 0.017	0.554 ± 0.018	0.713 ± 0.012	0.260 ± 0.004	0.754 ± 0.026	0.628 ± 0.011
	$\mathcal{L}_{\text{BerHu}}$	0.431 ± 0.002	0.229 ± 0.003	1.857 ± 0.010	0.314 ± 0.005	0.554 ± 0.004	0.714 ± 0.004	0.222 ± 0.005	0.688 ± 0.012	0.672 ± 0.010
	$\mathcal{L}_{\text{Ruber}}$	0.427 ± 0.013	0.226 ± 0.002	1.843 ± 0.004	0.311 ± 0.005	0.553 ± 0.005	0.721 ± 0.006	0.231 ± 0.015	0.690 ± 0.024	0.664 ± 0.025
	$\mathcal{L}_{\text{Barron}}$	0.458 ± 0.012	0.226 ± 0.009	1.857 ± 0.040	0.304 ± 0.020	0.545 ± 0.019	0.708 ± 0.016	0.289 ± 0.032	0.815 ± 0.060	0.569 ± 0.043
	$\mathcal{L}_{\text{trim}}$	0.430 ± 0.011	0.234 ± 0.005	1.880 ± 0.020	0.290 ± 0.014	0.537 ± 0.015	0.701 ± 0.012	0.247 ± 0.026	0.747 ± 0.069	0.615 ± 0.043
	$\mathcal{L}_{\text{ScaledSIError}}$	**0.411** ± 0.010	0.237 ± 0.011	1.875 ± 0.045	0.301 ± 0.027	0.546 ± 0.029	0.713 ± 0.019	**0.196** ± 0.003	**0.649** ± 0.018	**0.702** ± 0.011
	$\mathcal{L}_{\text{WeightedL2}}$	0.433 ± 0.013	0.225 ± 0.002	1.846 ± 0.016	0.314 ± 0.005	0.550 ± 0.009	0.711 ± 0.009	0.278 ± 0.016	0.811 ± 0.033	0.564 ± 0.022
	$\text{OSL}_{\mathcal{L}_1}$	0.417 ± 0.009	0.211 ± 0.002	1.771 ± 0.012	0.334 ± 0.011	0.579 ± 0.008	0.735 ± 0.001	0.279 ± 0.010	0.784 ± 0.017	0.618 ± 0.007
	$\text{OSL}_{\mathcal{L}_2}$	0.423 ± 0.007	**0.208** ± 0.003	**1.757** ± 0.021	**0.339** ± 0.009	**0.582** ± 0.006	**0.736** ± 0.006	0.305 ± 0.020	0.841 ± 0.024	0.558 ± 0.010
	$\text{FOSL}_{\mathcal{L}_1}$	0.413 ± 0.008	0.233 ± 0.013	1.876 ± 0.058	0.331 ± 0.021	0.557 ± 0.027	0.716 ± 0.030	0.229 ± 0.021	0.718 ± 0.032	0.662 ± 0.014

losses, we perform a cross-data set generalization study: While training on the noisy *NYUD-v2* data, we measure the performance of the models on the high-quality *DIODE* data set with the help of *iBims-1* as validation data in the hyperparameter optimization. Thereby, we consider varying amounts of training data being used to investigate the effect of more instances that might provide more stable estimates. Along with that, we further report the results on the Eigen test split for comparison. However, one notes that these test examples are gathered in the same way as the training data and thus incorporate the same noise that we approach to dump.

As can be seen in Table 1, the scaled SI error outperforms the other losses with regard to the *NYUD-v2* test data. However, when considering the cleaner *DIODE* benchmark data, the scaled SI loss shows less robust behavior, often not even improving baselines such as $\mathcal{L}_2$ itself. This demonstrates the inappropriateness to assess depth models on such noisy benchmark data.

On the contrary, most of the more robust loss variants improve over the conventional $\mathcal{L}_2$ loss, especially, when there is more training data provided. Notably, with only few exceptions, the superset loss variants outperform the baselines $\mathcal{L}_2$ and $\mathcal{L}_1$ in almost all cases, often even significantly. $\text{OSL}_{\mathcal{L}_1}$ turns out to work reasonably well when a small number of instances is provided, whereas $\text{OSL}_{\mathcal{L}_2}$ improves over the other methods for higher numbers of instances. Interestingly, albeit comprising it as a special case, $\text{FOSL}_{\mathcal{L}_1}$ often provides slightly inferior performance compared to $\text{OSL}_{\mathcal{L}_1}$. As the latter involves two loss-specific hyperparameters, this is most likely because of misleading draws in the hyperparameter optimization due to the larger hyperparameter space. While this shows the appealing property of $\text{OSL}_{\mathcal{L}_1}$ only having a single parameter to tune, spending more computational budget could leverage the increasing expressiveness of $\text{FOSL}_{\mathcal{L}_1}$ further.

Table 2: Averaged results and standard deviations of models trained on 2k instances from *NYUD-v2* on *DIODE* for varying noise levels. As before, best results per noise level and metric are indicated in bold.

Noise $\hat{\epsilon}$	Loss	REL (↓)	$\log_{10}$ (↓)	RMS (↓)	δ_1 (↑)	δ_2 (↑)	δ_3 (↑)
0.5	$\mathcal{L}_2$	0.851 ± 0.071	0.258 ± 0.013	2.090 ± 0.107	0.180 ± 0.022	0.398 ± 0.036	0.622 ± 0.033
	$\mathcal{L}_1$	0.477 ± 0.022	0.226 ± 0.003	1.859 ± 0.011	0.314 ± 0.010	0.541 ± 0.011	0.695 ± 0.011
	$\mathcal{L}_{\text{Huber}}$	0.576 ± 0.079	0.221 ± 0.008	1.847 ± 0.057	0.294 ± 0.038	0.540 ± 0.036	0.698 ± 0.016
	$\mathcal{L}_{\text{BerHu}}$	0.448 ± 0.003	0.220 ± 0.006	1.832 ± 0.041	**0.325** ± 0.006	0.558 ± 0.005	0.716 ± 0.008
	$\mathcal{L}_{\text{Ruber}}$	0.440 ± 0.009	0.225 ± 0.004	1.854 ± 0.014	0.315 ± 0.014	0.544 ± 0.008	0.706 ± 0.004
	$\mathcal{L}_{\text{Barron}}$	0.707 ± 0.045	0.234 ± 0.007	1.913 ± 0.050	0.233 ± 0.019	0.474 ± 0.030	0.671 ± 0.015
	$\mathcal{L}_{\text{trim}}$	0.461 ± 0.006	0.271 ± 0.004	1.874 ± 0.012	0.273 ± 0.003	0.503 ± 0.003	0.680 ± 0.002
	$\mathcal{L}_{\text{ScaledSIError}}$	0.606 ± 0.147	0.471 ± 0.221	2.512 ± 0.426	0.127 ± 0.101	0.252 ± 0.193	0.370 ± 0.259
	$\mathcal{L}_{\text{WeightedL2}}$	0.731 ± 0.009	0.241 ± 0.002	1.955 ± 0.013	0.225 ± 0.007	0.453 ± 0.007	0.661 ± 0.004
	$\text{OSL}_{\mathcal{L}_1}$	0.438 ± 0.037	**0.217** ± 0.004	**1.806** ± 0.025	0.323 ± 0.004	**0.561** ± 0.007	**0.719** ± 0.008
	$\text{OSL}_{\mathcal{L}_2}$	0.782 ± 0.072	0.243 ± 0.012	1.999 ± 0.083	0.201 ± 0.026	0.428 ± 0.038	0.649 ± 0.029
	$\text{FOSL}_{\mathcal{L}_1}$	**0.425** ± 0.015	0.224 ± 0.006	1.848 ± 0.043	0.310 ± 0.014	0.551 ± 0.012	0.709 ± 0.006
1.0	$\mathcal{L}_2$	1.466 ± 0.216	0.344 ± 0.039	3.167 ± 0.106	0.136 ± 0.042	0.279 ± 0.079	0.457 ± 0.089
	$\mathcal{L}_1$	0.482 ± 0.012	0.217 ± 0.002	1.814 ± 0.010	0.337 ± 0.006	**0.565** ± 0.005	0.707 ± 0.006
	$\mathcal{L}_{\text{Huber}}$	1.036 ± 0.039	0.282 ± 0.005	2.300 ± 0.021	0.142 ± 0.013	0.333 ± 0.017	0.566 ± 0.008
	$\mathcal{L}_{\text{BerHu}}$	0.484 ± 0.029	**0.216** ± 0.003	1.833 ± 0.020	0.322 ± 0.005	0.554 ± 0.006	0.712 ± 0.003
	$\mathcal{L}_{\text{Ruber}}$	0.453 ± 0.014	0.222 ± 0.000	1.847 ± 0.011	0.313 ± 0.010	0.554 ± 0.001	0.717 ± 0.005
	$\mathcal{L}_{\text{Barron}}$	1.063 ± 0.127	0.289 ± 0.018	2.367 ± 0.157	0.135 ± 0.024	0.320 ± 0.041	0.549 ± 0.040
	$\mathcal{L}_{\text{trim}}$	0.547 ± 0.018	0.366 ± 0.002	1.955 ± 0.008	0.213 ± 0.008	0.315 ± 0.010	0.489 ± 0.001
	$\mathcal{L}_{\text{ScaledSIError}}$	0.654 ± 0.066	0.460 ± 0.184	2.546 ± 0.387	0.128 ± 0.119	0.240 ± 0.204	0.341 ± 0.253
	$\mathcal{L}_{\text{WeightedL2}}$	0.785 ± 0.093	0.253 ± 0.007	2.029 ± 0.066	0.203 ± 0.036	0.422 ± 0.040	0.639 ± 0.022
	$\text{OSL}_{\mathcal{L}_1}$	0.457 ± 0.027	**0.216** ± 0.003	**1.812** ± 0.033	0.327 ± 0.009	0.560 ± 0.011	0.715 ± 0.007
	$\text{OSL}_{\mathcal{L}_2}$	1.113 ± 0.156	0.286 ± 0.020	2.375 ± 0.178	0.168 ± 0.015	0.345 ± 0.036	0.567 ± 0.043
	$\text{FOSL}_{\mathcal{L}_1}$	**0.449** ± 0.013	**0.216** ± 0.003	1.822 ± 0.022	**0.338** ± 0.013	0.562 ± 0.008	**0.718** ± 0.004

Nevertheless, the improvements in robustness can only be observed with relatively small margins. To highlight the effects of noisy data in a more exposing way, we show the results of an additional experiment injecting artificial noise into the original training data. To do so, we sample each observed training depth z from a normal distribution $\mathcal{N}(z, \hat{\sigma})$ with $\hat{\sigma}(x) := 0.01x^2 + \hat{\epsilon}$, where $\hat{\epsilon} \in \{0.5, 1.0\}$ is a parameter controlling the noise level. Note that we incorporate a heteroscedastic noise that increases with higher depth values.

The results in Table 2 demonstrate the incapability of conventional least squares optimization to provide reliable estimators under high noise. Accordingly, $\mathcal{L}_{\text{ScaledSIError}}$ does not lead to models learned in a robust manner either. As opposed to that, most of the conventional robust methods, especially the superset losses, turn out to be appropriate choices. Noteworthy, $\text{OSL}_{\mathcal{L}_1}$ provides the best performance for $\hat{\epsilon} = 0.5$, whereas $\text{FOSL}_{\mathcal{L}_1}$ proves its robustness capabilities for a higher noise of $\hat{\epsilon} = 1.0$.

4.3. Heterogeneous Depth Sensors: SunRGBD

While a single sensor was used to construct *NYUD-v2*, one may be interested in the case where multiple data sources are combined. In fact, this aggravates the problem of heterogeneous errors violating classical statistical assumptions as discussed before. In the following,

Table 3: Averaged results and standard deviations on models trained on 2k instances and the complete data set of *SunRGBD* on *DIODE*. The best model is indicated in bold per number of instances and metric.

# Insts.	Loss	*DIODE* REL (↓)	*DIODE* $\log_{10}$ (↓)	*DIODE* RMS (↓)	*DIODE* δ_1 (↑)	*DIODE* δ_2 (↑)	*DIODE* δ_3 (↑)	*SunRGBD* REL (↓)	*SunRGBD* RMS (↓)	*SunRGBD* δ_1 (↑)
2k	$\mathcal{L}_2$	0.512 ± 0.062	0.264 ± 0.043	1.922 ± 0.061	0.292 ± 0.033	0.515 ± 0.044	0.671 ± 0.039	0.432 ± 0.026	**1.135** ± 0.021	0.423 ± 0.010
	$\mathcal{L}_1$	0.432 ± 0.013	0.222 ± 0.006	1.837 ± 0.041	0.323 ± 0.006	0.554 ± 0.010	0.717 ± 0.011	0.423 ± 0.029	1.196 ± 0.052	0.415 ± 0.021
	$\mathcal{L}_{\text{Huber}}$	0.440 ± 0.016	0.218 ± 0.004	1.812 ± 0.028	0.328 ± 0.011	0.571 ± 0.011	0.726 ± 0.009	0.448 ± 0.028	1.175 ± 0.052	0.416 ± 0.019
	$\mathcal{L}_{\text{BerHu}}$	0.429 ± 0.017	0.220 ± 0.010	1.810 ± 0.051	0.316 ± 0.025	0.561 ± 0.022	0.726 ± 0.019	0.445 ± 0.014	1.222 ± 0.031	0.404 ± 0.009
	$\mathcal{L}_{\text{Ruber}}$	0.421 ± 0.011	0.218 ± 0.006	1.796 ± 0.035	0.321 ± 0.013	0.570 ± 0.008	0.729 ± 0.010	0.449 ± 0.011	1.218 ± 0.024	0.402 ± 0.008
	$\mathcal{L}_{\text{Barron}}$	0.463 ± 0.005	0.219 ± 0.005	1.823 ± 0.030	0.325 ± 0.012	0.560 ± 0.009	0.718 ± 0.011	0.480 ± 0.029	1.239 ± 0.059	0.403 ± 0.015
	$\mathcal{L}_{\text{trim}}$	0.500 ± 0.026	0.223 ± 0.003	1.863 ± 0.055	0.334 ± 0.014	0.565 ± 0.006	0.703 ± 0.009	**0.419** ± 0.044	1.150 ± 0.032	**0.429** ± 0.017
	$\mathcal{L}_{\text{ScaledSIError}}$	0.419 ± 0.004	0.228 ± 0.004	1.836 ± 0.019	0.309 ± 0.014	0.550 ± 0.011	0.714 ± 0.009	0.446 ± 0.009	1.224 ± 0.024	0.399 ± 0.006
	$\mathcal{L}_{\text{WeightedL2}}$	0.462 ± 0.018	0.223 ± 0.006	1.839 ± 0.038	0.328 ± 0.008	0.551 ± 0.008	0.708 ± 0.012	0.422 ± 0.026	1.159 ± 0.035	0.425 ± 0.013
	$\text{OSL}_{\mathcal{L}_1}$	0.424 ± 0.014	**0.206** ± 0.003	**1.730** ± 0.019	**0.346** ± 0.008	**0.591** ± 0.006	**0.734** ± 0.008	0.468 ± 0.011	1.192 ± 0.035	0.403 ± 0.008
	$\text{OSL}_{\mathcal{L}_2}$	0.471 ± 0.039	0.209 ± 0.004	1.756 ± 0.022	0.327 ± 0.010	0.577 ± 0.005	0.731 ± 0.013	0.464 ± 0.033	1.194 ± 0.059	0.408 ± 0.020
	$\text{FOSL}_{\mathcal{L}_1}$	**0.413** ± 0.009	0.219 ± 0.002	1.801 ± 0.008	0.318 ± 0.009	0.562 ± 0.078	0.728 ± 0.001	0.435 ± 0.013	1.220 ± 0.024	0.402 ± 0.009
Full	$\mathcal{L}_2$	0.418 ± 0.014	0.207 ± 0.003	1.733 ± 0.023	0.342 ± 0.003	0.585 ± 0.009	0.750 ± 0.011	0.470 ± 0.002	1.238 ± 0.007	0.398 ± 0.003
	$\mathcal{L}_1$	0.408 ± 0.008	0.219 ± 0.004	1.787 ± 0.026	0.304 ± 0.010	0.574 ± 0.008	0.746 ± 0.006	0.462 ± 0.014	1.250 ± 0.015	0.394 ± 0.006
	$\mathcal{L}_{\text{Huber}}$	0.394 ± 0.010	0.208 ± 0.009	1.715 ± 0.050	0.345 ± 0.030	0.604 ± 0.023	0.766 ± 0.014	0.483 ± 0.007	1.260 ± 0.021	0.388 ± 0.008
	$\mathcal{L}_{\text{BerHu}}$	0.391 ± 0.008	0.210 ± 0.009	1.720 ± 0.048	0.332 ± 0.026	0.603 ± 0.016	0.767 ± 0.014	0.485 ± 0.015	1.283 ± 0.014	0.386 ± 0.002
	$\mathcal{L}_{\text{Ruber}}$	0.376 ± 0.003	0.203 ± 0.002	1.679 ± 0.011	0.338 ± 0.011	0.617 ± 0.009	0.788 ± 0.001	0.489 ± 0.012	1.278 ± 0.017	0.388 ± 0.002
	$\mathcal{L}_{\text{Barron}}$	0.415 ± 0.009	0.208 ± 0.002	1.727 ± 0.017	0.335 ± 0.005	0.590 ± 0.002	0.762 ± 0.004	0.480 ± 0.010	1.263 ± 0.016	0.393 ± 0.003
	$\mathcal{L}_{\text{trim}}$	0.426 ± 0.011	0.218 ± 0.009	1.803 ± 0.049	0.325 ± 0.019	0.565 ± 0.015	0.734 ± 0.013	0.483 ± 0.023	1.316 ± 0.072	0.391 ± 0.010
	$\mathcal{L}_{\text{ScaledSIError}}$	0.377 ± 0.006	0.204 ± 0.007	1.690 ± 0.037	0.345 ± 0.014	0.609 ± 0.011	0.768 ± 0.008	0.471 ± 0.012	1.269 ± 0.017	0.387 ± 0.005
	$\mathcal{L}_{\text{WeightedL2}}$	0.418 ± 0.012	0.208 ± 0.007	1.748 ± 0.045	0.345 ± 0.013	0.581 ± 0.014	0.745 ± 0.015	**0.463** ± 0.005	1.226 ± 0.011	0.401 ± 0.004
	$\text{OSL}_{\mathcal{L}_1}$	**0.372** ± 0.019	**0.187** ± 0.007	**1.598** ± 0.050	**0.364** ± 0.003	**0.632** ± 0.011	**0.796** ± 0.012	0.475 ± 0.011	1.229 ± 0.014	0.401 ± 0.006
	$\text{OSL}_{\mathcal{L}_2}$	0.425 ± 0.054	0.195 ± 0.006	1.659 ± 0.037	0.354 ± 0.019	0.607 ± 0.015	0.765 ± 0.020	0.480 ± 0.016	**1.199** ± 0.016	**0.403** ± 0.006
	$\text{FOSL}_{\mathcal{L}_1}$	0.381 ± 0.004	0.210 ± 0.003	1.706 ± 0.018	0.324 ± 0.024	0.599 ± 0.014	0.769 ± 0.008	0.483 ± 0.016	1.242 ± 0.018	0.392 ± 0.003

we study the performance of models trained on *SunRGBD* for a varying number of training instances. In the appendix, we present further results on the test split of *SunRGBD*.

With less conformant error terms, the optimization with weaker model assumptions turns out to be reasonable. Table 3 shows the results for models trained on a subset of 2k instances and the full data set. As expected, the optimization based on $\mathcal{L}_2$ turns out to be misleading, especially for a small number of instances. Here, all robust variants turn out to work significantly better, most notably the OSL-based methods. $\text{OSL}_{\mathcal{L}_1}$ delivers the best performance with regard to the presented metrics. All superset losses improve over their respective baselines $\mathcal{L}_1$ and $\mathcal{L}_2$.

For a larger number of training examples, this trend continues, with $\text{OSL}_{\mathcal{L}_1}$ also providing the best performance for all reported metrics. Although less drastically, $\text{OSL}_{\mathcal{L}_2}$ and $\text{FOSL}_{\mathcal{L}_1}$ still outperform $\mathcal{L}_2$ and $\mathcal{L}_1$ respectively, further confirming the adequacy of an imprecisiation-based modeling. All in all, these results are in accordance with the initial motivation: By weakening the assumptions about the given data, we can leverage more robust loss alternatives for more accurate depth estimators.

5. Conclusion

We motivated the use of robust regression in depth estimation and revisited related loss functions, either applied in classical regression or specifically tailored to the domain of monocular depth estimation. Moreover, as an alternative to established approaches, we proposed the idea of "data imprecisiation" combined with superset learning. Instead of assuming precise but unreliable depth sensor signals as ground truth, the idea is to replace

these targets by (fuzzy) intervals, leading to an imprecise but more reliable representation of the ground truth.

In an extensive empirical evaluation, we could demonstrate the effectiveness of robust losses compared to conventional approaches such as OLS. Especially in the regime of little data with high noise, the superset learning approach turns out to achieve state-of-the-art performance.

Motivated by these results, we plan to further elaborate on the modeling of data to further improve robustness. In particular, going beyond a global (homogeneous) imprecisiation, we plan to investigate modeling on a per-instance basis, e.g., by distinguishing the reliability of instances based on the depth value itself or the position in the image.

Acknowledgments

This work was supported by the German Research Foundation (DFG) under Grant No. 420493178. Moreover, computational resources were provided by the Paderborn Center for Parallel Computing (PC^2).

References

Min Sung Ahn, Hosik Chae, Donghun Noh, Hyunwoo Nam, and Dennis W. Hong. Analysis and noise modeling of the Intel RealSense D435 for mobile robots. In *Proc. of the 16th International Conference on Ubiquitous Robots, UR*, pages 707–711. IEEE, 2019.

Jonathan T. Barron. A general and adaptive robust loss function. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 4331–4339. Computer Vision Foundation / IEEE, 2019.

Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. AdaBins: Depth estimation using adaptive bins. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 4009–4018, June 2021.

Vivien Cabannes, Alessandro Rudi, and Francis R. Bach. Structured prediction with partial labelling through the infimum loss. In *Proc. of the 37th International Conference on Machine Learning, ICML*, volume 119, pages 1230–1239. PMLR, 2020.

Marcela Carvalho, Bertrand Le Saux, Pauline Trouvé-Peloux, Andrés Almansa, and Frédéric Champagnat. On regression losses for deep depth estimation. In *Proc. of the IEEE International Conference on Image Processing, ICIP*, pages 2915–2919. IEEE, 2018.

Qifeng Chen and Vladlen Koltun. Fast MRF optimization with application to depth reconstruction. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 3914–3921. IEEE Computer Society, 2014.

Yadolah Dodge. An introduction to L1-norm based statistical data analysis. *Computational Statistics & Data Analysis*, 5(4):239–253, 1987. ISSN 0167-9473.

Christopher Dougherty. *Introduction to econometrics*. Oxford University Press, 2011.

David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems, NIPS*, pages 2366–2374, 2014.

Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 2002–2011. IEEE Computer Society, 2018.

Peter J. Huber. *Robust Statistics*. Wiley Series in Probability and Statistics. Wiley, 1981.

E. Hüllermeier. Learning from imprecise and fuzzy observations: Data disambiguation through generalized loss minimization. *Int. J. Approx. Reason.*, 55(7):1519–1534, 2014.

Go Irie, Takahito Kawanishi, and Kunio Kashino. Robust learning for deep monocular depth estimation. In *Proc. of the IEEE International Conference on Image Processing, ICIP*, pages 964–968. IEEE, 2019.

Takeaki Kariya and Hiroshi Kurata. *Generalized least squares*. John Wiley & Sons, 2004.

Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, NeurIPS*, pages 5574–5584, 2017.

Kourosh Khoshelham and Sander Oude Elberink. Accuracy and resolution of Kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454, 2012.

George J. Klir and Tina A. Folger. *Fuzzy sets, uncertainty and information*. Prentice Hall, 1988.

Tobias Koch, Lukas Liebel, Friedrich Fraundorfer, and Marco Körner. Evaluation of CNN-based single-image depth estimation methods. In *Proc. of the European Conference on Computer Vision, ECCV, Workshops Part III*, volume 11131 of *LNCS*, pages 331–348. Springer, 2018.

Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *Proc. of the 4th International Conference on 3D Vision, 3DV*, pages 239–248. IEEE Computer Society, 2016.

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. *CoRR*, abs/1907.10326, 2019.

Zhengqi Li and Noah Snavely. MegaDepth: Learning single-view depth prediction from internet photos. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 2041–2050. IEEE Computer Society, 2018.

Julian Lienen and Eyke Hüllermeier. Instance weighting through data imprecisiation. *Int. J. Approx. Reason.*, 134:1–14, July 2021. ISSN 0888-613X.

Fangchang Ma and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *Proc. of the IEEE International Conference on Robotics and Automation, ICRA*, pages 1–8. IEEE, 2018.

Chuong V. Nguyen, Shahram Izadi, and David R. Lovell. Modeling Kinect sensor noise for improved 3D reconstruction and tracking. In *Proc. of the 2nd International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pages 524–530. IEEE Computer Society, 2012.

René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.

Philipp Rosenberger, Martin Holder, Marina Zirulnik, and Hermann Winner. Analysis of real world sensor behavior for rising fidelity of physically based Lidar sensor models. In *Proc. of the IEEE Intelligent Vehicles Symposium, IV, Changshu*, pages 611–616. IEEE, 2018.

B. Schölkopf and AJ. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *Proc. of the 12th European Conference on Computer Vision, ECCV, Part V*, volume 7576 of *LNCS*, pages 746–760. Springer, 2012.

Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 567–576. IEEE Computer Society, 2015.

Deqing Sun, Stefan Roth, and Michael J. Black. Secrets of optical flow estimation and their principles. In *Proc. of the 23rd IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 2432–2439. IEEE Computer Society, 2010.

Igor Vasiljevic, Nicholas I. Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z. Dai, Andrea F. Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R. Walter, and Gregory Shakhnarovich. DIODE: A dense indoor and outdoor depth dataset. *CoRR*, abs/1908.00463, 2019.

Oliver Wasenmüller and Didier Stricker. Comparison of Kinect V1 and V2 depth images in terms of accuracy and precision. In *Proc. of the Asian Conference on Computer Vision, ACCV, Workshops, Part II*, volume 10117 of *LNCS*, pages 34–45. Springer, 2016.

Yicheng Wu, Vivek Boominathan, Huaijin G. Chen, Aswin C. Sankaranarayanan, and Ashok Veeraraghavan. PhaseCam3D - Learning phase masks for passive single view depth estimation. In *Proc. of the IEEE International Conference on Computational Photography, ICCP*, pages 1–12. IEEE, 2019.