International Journal of Computer Vision manuscript No.
(will be inserted by the editor)

IEBins: Iterative Elastic Bins for Monocular Depth Estimation and Completion

Shuwei Shao[1], **Zhongcai Pei**[1], **Weihai Chen**[1,*], **Peter C. Y. Chen**[2], **Zhengguo Li**[3,*]

Received: date / Accepted: date

Abstract Monocular depth estimation and completion are fundamental aspects of geometric computer vision, serving as essential techniques for various downstream applications. In recent developments, several methods have reformulated these two tasks as a *classification-regression* problem, deriving depth with a linear combination of predicted probabilistic distribution and bin centers. In this paper, we introduce an innovative concept termed **iterative elastic bins (IEBins)** for the classification-regression-based monocular depth estimation and completion. The IEBins involves the idea of iterative division of bins. In the initialization stage, a coarse and uniform discretization is applied to the entire depth range. Subsequent update stages then iteratively identify and uniformly discretize the target bin, by leveraging it as the new depth range for further refinement. To mitigate the risk of error accumulation during iterations, we propose a novel elastic target bin, replacing the original one. The width of this elastic bin is dynamically adapted according to the depth uncertainty. Furthermore, we develop dedicated frameworks to instantiate the IEBins. Extensive experiments on the KITTI, NYU-Depth-v2, SUN RGB-D, ScanNet and DIODE datasets indicate that our method outperforms prior state-of-the-art monocular depth estimation and completion competitors.

Keywords Monocular Depth Estimation, Depth Completion, Classification-regression, Iterative Refinement

1 Introduction

Monocular depth estimation and completion stand as enduring and pivotal pillars of geometric computer vision, boasting wide-ranging applications in fields such as robotics (Jia et al. 2023), 3D reconstruction (Izadi et al. 2011), and scene understanding (Chen et al. 2019a). Monocular depth estimation aims to acquire the depth map from a single RGB image, presenting a challenging scenario due to its inherent ill-posed nature and the scale ambiguity issue. This ambiguity arises because a 2D image can correspond to an infinite number of 3D scenes. The mainstream of depth completion is to complete the sparse depth map that provides accurate sparse depth priors of the surrounding environments under the guidance of RGB image. In recent times, an increasing number of learning-based approaches have emerged, driving the evolution of monocular depth estimation and completion (Bhat et al. 2021; Eigen et al. 2014; Hu et al. 2021; Yan et al. 2023; Yuan et al. 2022; Zhang et al. 2023). Without loss of integrity, these methods can be categorized into three groups: *regression*, *classification* and *classification-regression*.

The **regression** approach, identified as the most fundamental and straightforward formulation (Eigen et al. 2014; Liu et al. 2021a; Ma & Karaman 2018; Shao et al. 2022; Yan et al. 2022; Yang et al. 2021; Yuan et al. 2022), directly generates continuous pixel-wise depth supervised by a regression loss. Despite its widespread use as a foundational paradigm, models based on regression tend to fall short in delivering satisfactory results (Fu et al. 2018). The **classification** approach, proposed in (Fu et al. 2018) and (Cao et al. 2017), formulates monocular depth estimation as per-pixel classification, aiming to predict the optimal depth interval. To achieve this, it discretizes the entire depth range into multiple intervals (bins) in either a uniform (Fig. 1 (a)) or a log-uniform space (Fig. 1 (b)). The final depth prediction is determined by the bin center (depth candidate) corresponding to the classified target bin. While the classification markedly enhances model performance benefiting from the fact that predicting depth intervals is easier than predicting exact depth values, it often results in poor visual quality characterized by discontinuity artifacts (Bhat et al. 2021).

Recently, some methods (Agarwal & Arora 2023; Bhat et al. 2021, 2022; Imran et al. 2019; Johnston & Carneiro 2020; Li et al. 2022) have reframed monocular depth estimation and completion as a per-pixel **classification-regression** problem. They generally involve learning probabilistic distribution for each pixel and using a linear combination with depth candidates to acquire the final depth prediction. In theory, this formulation captures the promise of achieving sub-pixel depth estimation, thus avoiding the discontinuity artifacts induced by using classification only. To be specific, Bhat et al. (2021) introduced an adaptive bins division (Fig.1 (c)) in monocular depth estimation based on the observation that

[1] School of Automation Science and Electrical Engineering, Beihang University, Beijing, China
[2] Department of Mechanical Engineering, National University of Singapore, Singapore
[3] VI department, Institute for Infocomm Research, A*STAR, Singapore
*corresponding authors: Weihai Chen and Zhengguo Li

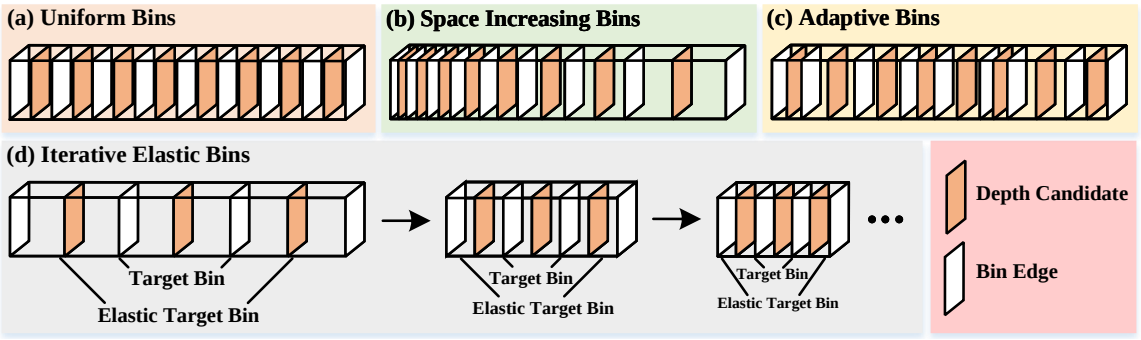

Fig. 1 Illustration of different bin types: uniform bins (Fu et al. 2018), space increasing bins (Fu et al. 2018), adaptive bins (Bhat et al. 2021) and the proposed iterative elastic bins.

significant depth distribution fluctuations across diverse scenes. Building upon this, Li et al. (2022) and Bhat et al. (2022) enhanced the adaptive bins strategy by either disentangling the generation of bins from probabilistic distribution prediction or gradually performing local predictions of depth distributions. In terms of depth completion, Imran et al. (2019) leveraged the classification-regression representation to alleviate the depth mixing issue at object boundaries. On top of that, Lee et al. (2021a) added an additional residual prediction step.

In this paper, we introduce a novel concept referred to as **iterative elastic bins (IEBins, Fig. 1 (d))**, tailored for the classification-regression-based monocular depth estimation and completion. The IEBins refines the binning structure in an iterative manner and uses multiple small numbers of bins, rather than adhering to the single standard number of bins. During the initialization stage, the entire depth range undergoes a coarse and uniform discretization. In each subsequent update stage, there is an iterative process to identify and uniformly discretize the target bin (the bin where the predicted depth is situated) by using it as the new depth range for ongoing refinement. Unfortunately, incorrect depth predictions at some stages may cause the depth ground-truth to lie outside the target bin, leading to optimization instability and reduced accuracy. To account for the error accumulation during iterations, we elastically adjust the width of the target bin based on the depth uncertainty, which is called elastic target bin. The depth uncertainty indicates possible errors in depth prediction. Taking inspiration from (Hu et al. 2022), we utilize the variance of the probabilistic distribution to measure the uncertainty. Notably, each subsequent update stage for depth completion is slightly different from that of monocular depth estimation. We tightly couple the IEBins with the non-local spatial propagation network (NLSPN) (Park et al. 2020) refinement in depth completion. The IEBins is capable of delivering the depth uncertainty to help NLSPN modulate the affinity map while the NLSPN is able to perform propagation refinement on the depth prediction from IEBins, allowing it to search for more suitable target bins. Furthermore, we develop dedicated frameworks to instantiate the IEBins. The frameworks are composed of two main components: a feature extractor and an iterative optimizer. For monocular depth estimation, the iterative optimizer is built upon the gated recurrent unit (GRU) with powerful capabilities in modeling temporal context. As for depth completion, the iterative optimizer is composed of convolutional layers and non-local spatial propagation network.

The IEBins approach was first proposed in our NeurIPS 2023 conference paper (Shao et al. 2023b), where we introduce IEBins and develop an IEBins-based framework for monocular depth estimation. In this full version, we: i) extend IEBins into depth completion and develop a dedicated completion framework by seamlessly integrating IEBins with the non-local spatial propagation network, achieving superior performance over prior competing approaches; ii) make a more comprehensive evaluation to validate the proposed method, including zero-shot generalization study on the ScanNet and DIODE datasets, further ablation study in depth estimation, quantitative and qualitative comparisons, sparsity study and ablation study in depth completion, alongside failure cases analysis; iii) show the benefits of improvements in downstream SLAM, both quantitatively and qualitatively, by incorporating different depth estimation methods into ORB-SLAM2 (Mur-Artal & Tardós 2017).

To summarize, our main contributions are three-fold:

- We introduce an innovative iterative elastic bins strategy for the classification-regression-based monocular depth estimation and completion.
- We develop new frameworks based on IEBins for monocular depth estimation and completion, which consist of a feature extractor and an iterative optimizer.
- Extensive experiments on the KITTI (Geiger et al. 2013), NYU-Depth-v2 (Silberman et al. 2012), SUN RGB-D (Song et al. 2015), ScanNet (Dai et al. 2017a) and DIODE (Vasiljevic et al. 2019) datasets indicate that our method surpasses prior state-of-the-art monocular depth estimation and completion competitors.

2 Related work

Monocular depth estimation. In recent years, learning-based monocular depth estimation has made significant strides. Saxena et al. (2005) pioneered the use of Markov Random Fields to capture critical local and global image features in depth estimation. Eigen et al. (2014) later introduced convolutional neural network (CNN)-based architectures to perform coarse-to-fine depth estimation. Since then, there has been extensive exploration of CNNs in this realm. In particular, Laina et al. (2016) applied the residual CNN to improve the optimization process. More recently, the Transformer architecture (Vaswani et al. 2017) has garnered considerable interests in computer vision (Dosovitskiy et al. 2021; Liu et al. 2021c; Peng et al. 2021). Capitalizing on the success of visual transformer in other tasks (Gao et al. 2021; Jiang et al.

2021; Yuan et al. 2021; Zheng et al. 2021), Yang et al. (2021), Ranftl et al. (2021) and Yuan et al. (2022) made a pivotal shift from CNN to Transformer, resulting in further performance improvements. Nevertheless, these methods encounter suboptimal solutions arising from the inherent drawback of regression (Fu et al. 2018). Fu et al. (2018) and Cao et al. (2017) instead formulated monocular depth estimation as a classification problem, in which they discretized the entire depth range into multiple bins and turned to predict the most likely depth interval. In a complementary effort, Diaz & Marathe (2019) softened the classification labels during training. Furthermore, Bhat et al. (2021), Johnston & Carneiro (2020), Li et al. (2022), Bhat et al. (2022) and Agarwal & Arora (2023) reframed monocular depth estimation from the perspective of classification-regression, aiming to alleviate the discontinuity artifacts associated with depth discretization. Among them, Bhat et al. (2021) introduced an adaptive bins strategy to dynamically generate bins from image to image. By contrast, we propose an iterative elastic bins paradigm using multiple small numbers of bins rather than a single standard number of bins like (Bhat et al. 2021). Moreover, the elastic nature of bins based on uncertainty inherently introduces the adaptive nature of bins. In a similar vein, Bhat et al. (2022) developed the step-wise binning structure refinement. However, Bhat et al. (2022) expands the number of bins at each stage by dividing all bins from the preceding stage while the number of bins in the proposed IEBins remains unchanged at different stages and only the target bin is located and divided.

Depth completion. With the rapid advancements in deep learning, image guided depth completion has experienced substantial progress in recent times. In a pioneering effort, Ma et al. (2019); Ma & Karaman (2018) leveraged an encoder-decoder architecture to complete the sparse depth map, operating within supervised and self-supervised frameworks. To alleviate the depth mixing issue at object boundaries, Imran et al. (2019) employed the classification-regression representation. On this basis, Lee et al. (2021a) incorporated an extra step for residual prediction and developed the plane-residual representation. Cheng et al. (2018) instead aimed to preserve the precise measurements from sparse depth input. They introduced the spatial propagation network (SPN) (Liu et al. 2017) into depth completion and implemented the convolutional spatial propagation network (CSPN), strategically positioned at the end of the encoder-decoder network to refine its prediction. Expanding upon the groundwork laid by CSPN, many subsequent works have emerged. Cheng et al. (2020) developed CSPN++ by using adaptive convolutional kernel sizes and number of iterations for propagation. Park et al. (2020) employed deformable kernels (Dai et al. 2017b) to effectively avoid irrelevant local neighbors during propagations. Lin et al. (2022) estimated independent affinity matrices to strengthen the representation capabilities and a diffusion suppression operation to prevent over-smoothing of depth. Zhang et al. (2023) combined Transformer and CNN to enhance the expressivity of the encoder-decoder network for better SPN refinement. In stark contrast, we tightly couple the proposed IEBins with the non-local spatial propagation network (Park et al. 2020) to fully exploit their individual strengths.

Rather than relying solely on a single branch, multi-branch networks have been extensively utilized for multi-modal fusion in several studies (Hu et al. 2021; Liu et al. 2021b; Tang et al. 2020; Van Gansbeke et al. 2019; Zhang & Funkhouser 2018). Traditional fusion techniques typically consist of operations like feature concatenation or elementwise summation. However, more advanced fusion strategies have been developed. Tang et al. (2020) incorporated guided image filtering (He et al. 2012) with dynamically generated spatially-variant kernels while Zhang et al. (2020) introduced a neighbor attention module. In addition, Zhao et al. (2021) developed a symmetric gated fusion mechanism. More recently, Rho et al. (2022) proposed a guided attention module to capture inter-modal dependencies.

Iterative refinement has attracted widespread attention in many fields. For instance, Teed & Deng (2020) proposed to iteratively update a motion field through the GRU for optical flow estimation, which simulates first-order optimization. The idea has further been adopted in some related tasks, including stereo (Lipson et al. 2021; Wang et al. 2022), structure from motion (Gu et al. 2023), and scene flow (Teed & Deng 2021). In this paper, we propose to refine the binning structure in an iterative elastic manner and develop a GRU-based iterative optimizer to predict the probabilistic distribution from its hidden state that is updated at each stage.

3 IEBins for Monocular Depth Estimation

3.1 Initialization

To initialize the IEBins, we discretize the entire depth range $[d_{\min}, d_{\max}]$ into N bins in a uniform space,

$$e_n = d_{\min} + nB, n = 0, 1, ..., N, \tag{1}$$

with

$$B = \frac{d_{\max} - d_{\min}}{N}, \tag{2}$$

where e_n is the n-th bin edge, B is the bin width, $d_{\max}$ is configured as 80 for the KITTI dataset (Geiger et al. 2013) and 10 for the NYU-Depth-v2 (Silberman et al. 2012) dataset, $d_{\min}$ is fixed at 0.1, and unless specified otherwise, N is set to 16. Notably, this is a significantly reduced number of bins compared to the 256 bins employed in AdaBins (Bhat et al. 2021).

Since direct usage of discrete bins for depth prediction is not feasible, we take the bin center to stand for each bin, i.e., depth candidate

$$\mathcal{D}_n = \frac{e_n + e_{n+1}}{2}, n = 0, 1, ..., N-1, \tag{3}$$

where $\mathcal{D}_n$ denotes the depth candidate associated with the n-th bin. Once the per-pixel probabilistic distribution corresponding to depth candidates is predicted, we are able to derive the depth prediction by combining them linearly

$$\widehat{\mathbf{D}}(\mathbf{p}) = \sum_{n=0}^{N-1} \mathcal{D}_n \cdot \mathbf{P}_n(\mathbf{p}), \tag{4}$$

where $\widehat{\mathbf{D}}(\mathbf{p})$ is the predicted depth at pixel coordinate $\mathbf{p}$ and $\mathbf{P}_n$ is the n-th depth probability.

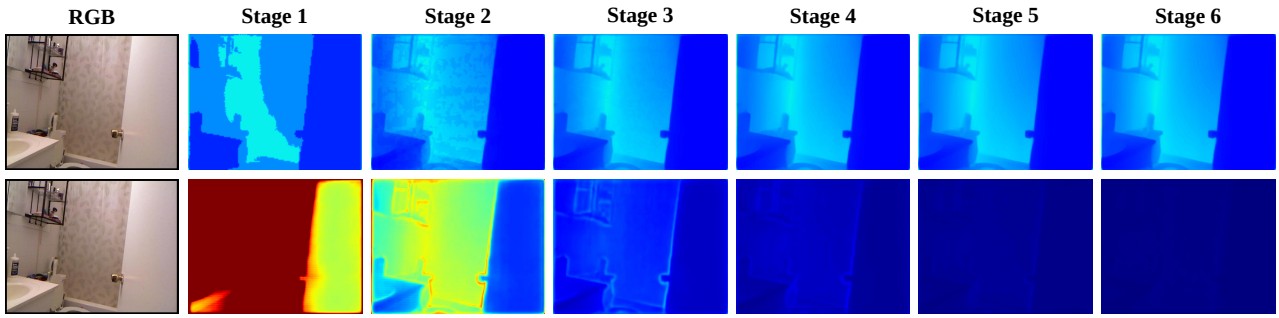

Fig. 2 Illustration of depth and uncertainty maps at each stage. The top row stands for depth maps, visualized using the bin center of each target bin to enhance the clarity of the refinement process. The bottom row depicts uncertainty maps, with yellow/red indicating high/highest uncertainty and blue representing low uncertainty.

3.2 Update

In every successive stage, the search is conducted at a finer granularity within the target bin of the preceding stage. The target bin signifies the specific bin where the predicted depth is situated, identified by comparing the predicted depth with bin edges

$$e_n \leq \widehat{\mathbf{D}}\left(\mathbf{p}\right) < e_{n+1} \tag{5}$$

and denoted as $[e_n, e_{n+1}]$. Nonetheless, as highlighted in the introduction section, the depth ground-truth may deviate from the target bin because of errors in depth prediction. In such case, the errors will accumulate progressively with each stage, leading to the instability in optimization and a decline in accuracy. To alleviate this potential failure instance and strengthen the paradigm robustness, we introduce a novel elastic target bin to update depth candidates, the width of which is adapted dynamically based on the depth uncertainty. Taking inspiration from (Hu et al. 2022), we use the variance of the probabilistic distribution to quantify uncertainty, mathematically,

$$\widehat{\mathbf{V}}\left(\mathbf{p}\right) = \sum_{n=0}^{N-1} \left(\mathcal{D}_n - \widehat{\mathbf{D}}\left(\mathbf{p}\right)\right)^2 \cdot \mathbf{P}_n\left(\mathbf{p}\right), \tag{6}$$

where $\widehat{\mathbf{V}}\left(\mathbf{p}\right)$ denotes the calculated variance. Then, the corresponding standard deviation is acquired by

$$\widehat{\sigma}\left(\mathbf{p}\right) = \sqrt{\widehat{\mathbf{V}}\left(\mathbf{p}\right)}. \tag{7}$$

Given the target bin $[e_n, e_{n+1}]$ and standard deviation $\widehat{\sigma}\left(\mathbf{p}\right)$, the elastic target bin is determined by

$$[e_n - \kappa\widehat{\sigma}\left(\mathbf{p}\right), e_{n+1} + \kappa\widehat{\sigma}\left(\mathbf{p}\right)], \tag{8}$$

where κ is a coefficient set to 0.5, governing the degree of error tolerance. In the presence of a large depth error in a pixel, the standard deviation increases, providing the elastic target bin with higher immunity against error. On top of that, we renew $d_{\min}$ and $d_{\max}$ as $e_n - 0.5\widehat{\sigma}\left(\mathbf{p}\right)$ and $e_{n+1} + 0.5\widehat{\sigma}\left(\mathbf{p}\right)$, respectively. Using Eqs.1, 2, 3 and 4, we acquire the updated depth candidates and prediction. The update stage continues until reaching the final step. It is crucial to highlight that the depth candidates $\mathcal{D}$ actually demonstrate spatial variability, attributed to the distinct elastic target bins in subsequent update steps. An example of depth maps and corresponding uncertainty maps at each stage is showcased in Fig. 2.

3.3 Estimation Framework

Feature extractor, which adopts an encoder-decoder structure with skip-connections. To build the encoder, the latest Swin-Transformer family backbones as introduced in (Liu et al. 2022a, 2021c) are employed. It takes an RGB image with dimension $H \times W$ as input and produces a four-level feature pyramid, which is then transmitted into the decoding phase via skip-connections. Inside the decoder, three neural conditional random field (CRF) modules (Yuan et al. 2022) are employed to capture the essential long-range dependencies. We interleave the neural CRF module with pixel shuffle (Shi et al. 2016) until reaching the $\frac{H}{4} \times \frac{W}{4}$ resolution. Next, the feature maps at $\frac{H}{4} \times \frac{W}{4}$ resolution from the encoder and decoder serve as the context feature and the initialization of the GRU hidden state for the iterative optimizer, respectively.

Iterative optimizer. For efficient prediction of the probabilistic distribution at each stage, we employ a GRU-based iterative optimizer, inspired by the approach in (Teed & Deng 2020). The rationale behind this lies in its ability to preserve information from previous stages, enabling comprehensive exploration of temporal context during iterations. Notably, the optimizer functions at a resolution of $\frac{H}{4} \times \frac{W}{4}$.

To begin with, the depth candidates $\mathcal{D}$ undergo a projection into the feature space through four 3×3 convolutional layers, each followed by a ReLU activation (Glorot et al. 2011). Subsequently, the resulting projected feature is concatenated with the context feature to achieve the input tensor $\mathbf{I}^k$. The structure inside GRU is as follows:

$$\mathbf{z}^{k+1} = \text{sigmoid}\left(Conv_{5\times5}\left(\left[\mathbf{h}^k, \mathbf{I}^k\right], W_z\right)\right), \tag{9}$$

$$\mathbf{r}^{k+1} = \text{sigmoid}\left(Conv_{5\times5}\left(\left[\mathbf{h}^k, \mathbf{I}^k\right], W_r\right)\right), \tag{10}$$

$$\widehat{\mathbf{h}}^{k+1} = \tanh\left(Conv_{5\times5}\left(\left[\mathbf{r}^{k+1} \odot \mathbf{h}^k, \mathbf{I}^k\right], W_h\right)\right), \tag{11}$$

$$\mathbf{h}^{k+1} = \left(1 - \mathbf{z}^{k+1}\right) \odot \mathbf{h}^k + \mathbf{z}^{k+1} \odot \widehat{\mathbf{h}}^{k+1}, \tag{12}$$

where k denotes the stage index, $\mathbf{z}$ denotes the update gate, $\mathbf{r}$ stands for the reset gate, $Conv_{5\times5}$ denotes the separable 5×5 convolution, $\odot$ denotes the element-wise multiplication, W_z, W_r and W_h stands for learnable parameters. The

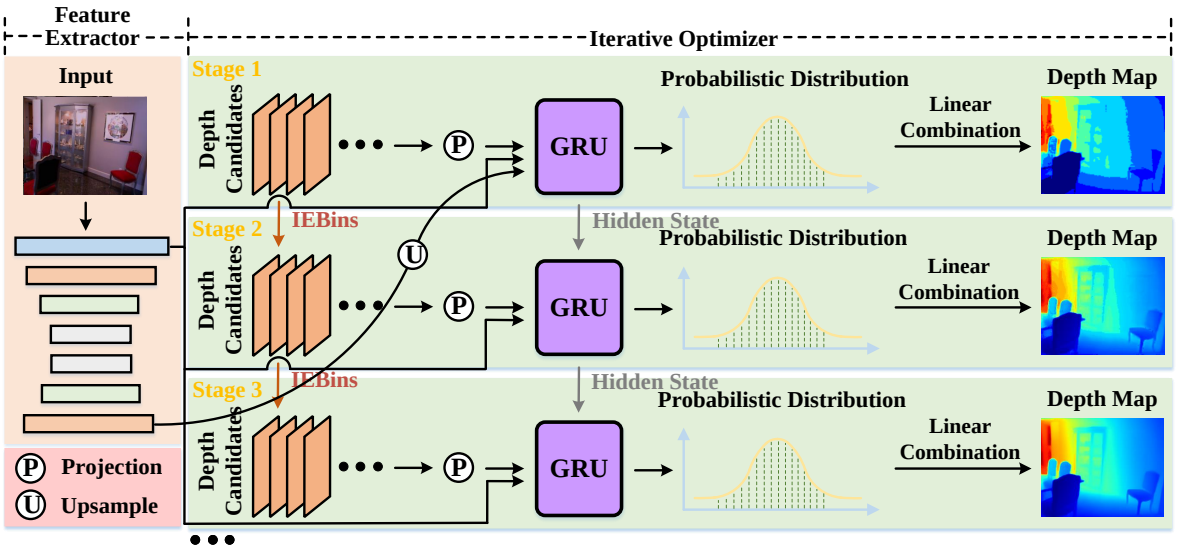

Fig. 3 An overview of the IEBins-based monocular depth estimation framework. The upsample refers to pixel shuffle (Shi et al. 2016) and the projection stands for four 3×3 convolutional layers.

hidden state $\mathbf{h}$ is initialized using the output of the decoder at $\frac{H}{4} \times \frac{W}{4}$ resolution. The probabilistic distribution prediction arises from the updated hidden state utilizing two 3×3 convolutional layers. At the same time, the feature maps undergo regularizations by ReLU and Softmax activations, respectively. The depth prediction is obtained through a linear combination of the predicted probabilistic distribution and depth candidates. We further leverage the bilinear interpolation to upsample the depth prediction to the original resolution.

Utilizing this optimizer, the refinement process starts from an initial coarse prediction and iteratively improves until the depth converges to the final result. An overview of the whole monocular depth estimation framework is demonstrated in Fig. 3.

3.4 Training Loss Function

Pixel-wise depth loss. In accordance with (Bhat et al. 2021; Yuan et al. 2022), we leverage a scaled Scale-Invariant loss for depth supervision, as introduced by (Lee et al. 2019),

$$\mathcal{L}_{pixel} = \sum_{k=1}^{K} \alpha \sqrt{\frac{1}{|\mathbf{T}|} \sum_{\mathbf{p}} \left(\mathbf{g}\left(\mathbf{p}\right)\right)^2 - \frac{\beta}{|\mathbf{T}|^2} \left(\sum_{\mathbf{p}} \mathbf{g}\left(\mathbf{p}\right)\right)^2}, \quad (13)$$

where $\mathbf{g}\left(\mathbf{p}\right) = \log \widehat{\mathbf{D}}\left(\mathbf{p}\right) - \log \mathbf{D}^{gt}\left(\mathbf{p}\right)$, K denotes the maximum number of stages, set to 6, while $\mathbf{T}$ denotes the set of pixels with valid ground-truth values. The values for α and β are chosen as 10 and 0.85, respectively, based on (Lee et al. 2019).

4 IEBins for Depth Completion

In this section, the introduced IEBins is extended to study depth completion. The **initialization** part is the same as monocular depth estimation. However, the **update** part is specifically crafted for depth completion by taking into consideration the spatial propagation network (SPN) refinement, which is widely adopted in depth completion to preserve the accurate sparse depth input and improve the overall quality of depth prediction (Cheng et al. 2020, 2018; Lin et al. 2022; Park et al. 2020).

4.1 update

In each subsequent stage, SPN refinement is first performed on the depth prediction obtained by linearly combining the probabilistic distribution and depth candidates as Eq.4. Following (Zhang et al. 2023), we leverage the non-local spatial propagation network (Park et al. 2020) (NLSPN) for further refinement. In order to enhance the resilience against inaccurate depth prediction during propagations, Park et al. (2020) proposed to predict a confidence map to minimize the propagation of unreliable depth values. Unfortunately, the confidence map in (Park et al. 2020) remains unchanged during the propagation process, inevitably limiting the representation capabilities. Our IEBins is capable of providing the uncertainty map without requiring additional parameters. More importantly, the uncertainty map is updated at each update stage. Hence, it is a natural choice to adopt the uncertainty map from IEBins. The spatial propagation of $\widehat{\mathbf{D}}\left(\mathbf{p}\right)$ utilizing its non-local neighbors $\Omega(\mathbf{p})$ is formulated as

$$\widetilde{\mathbf{D}}\left(\mathbf{p}\right) = \widehat{\mathbf{W}}_{\mathbf{p}}\left(\mathbf{p}\right) \widehat{\mathbf{D}}\left(\mathbf{p}\right) + \sum_{\mathbf{p}_{nei} \in \Omega(\mathbf{p})} \widehat{\mathbf{W}}_{\mathbf{p}_{nei}}\left(\mathbf{p}\right) \widehat{\mathbf{D}}\left(\mathbf{p}_{nei}\right), \quad (14)$$

where $\widehat{\mathbf{W}}_{\mathbf{p}_{nei}}\left(\mathbf{p}\right)$ is the confidence-modulated affinity weight between the target pixel $\mathbf{p}$ and its neighbor pixel $\mathbf{p}_{nei}$, $\widehat{\mathbf{W}}_{\mathbf{p}}\left(\mathbf{p}\right) = 1 - \sum_{\mathbf{p}_{nei} \in \Omega(\mathbf{p})} \widehat{\mathbf{W}}_{\mathbf{p}_{nei}}\left(\mathbf{p}\right)$, indicating the extent to which the original depth will be preserved. The affinity modulation is achieved by

$$\overline{\mathbf{W}}_{\mathbf{p}_{nei}}\left(\mathbf{p}\right) = \tanh(\mathbf{W}_{\mathbf{p}_{nei}}\left(\mathbf{p}\right))/\gamma, \quad (15)$$

$$\overline{\overline{\mathbf{W}}}_{\mathbf{p}_{nei}}\left(\mathbf{p}\right) = \widehat{\mathbf{C}}\left(\mathbf{p}_{nei}\right) \overline{\mathbf{W}}_{\mathbf{p}_{nei}}\left(\mathbf{p}\right), \quad (16)$$

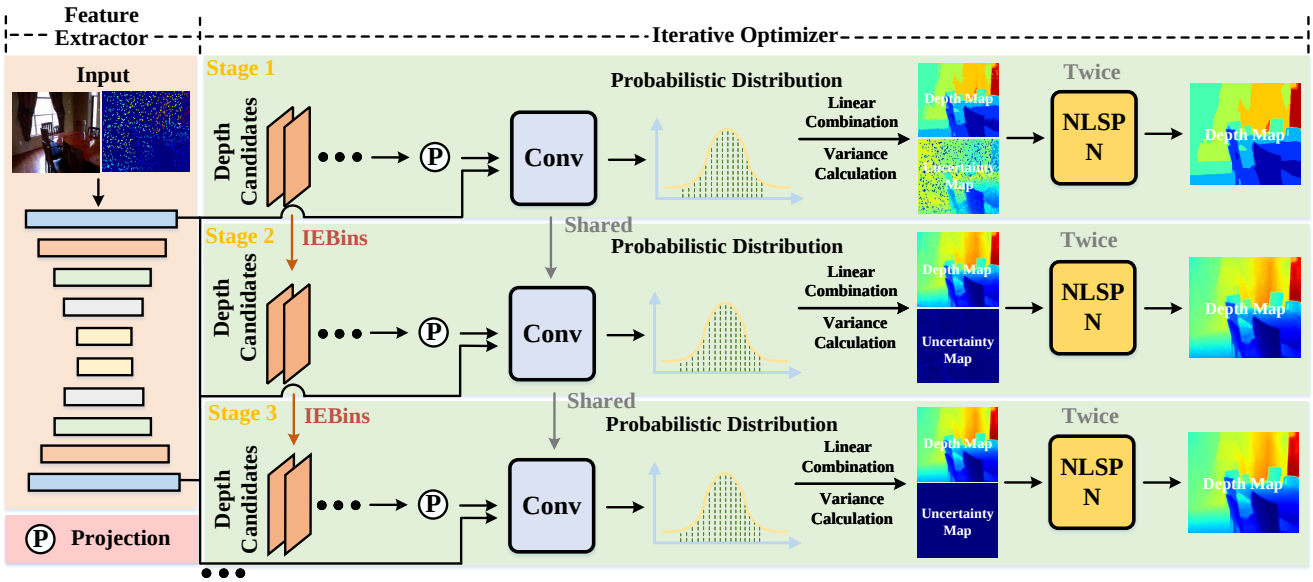

Fig. 4 An overview of the IEBins-based depth completion framework. The projection stands for a single 3×3 convolutional layer. We note that the NLSPN refinement (Park et al. 2020) is performed twice at each stage.

$$\widehat{\mathbf{W}}_{\mathbf{p}_{nei}}(\mathbf{p}) = \overline{\overline{\mathbf{W}}}_{\mathbf{p}_{nei}}(\mathbf{p}) \Big/ \sum_{\mathbf{p}_{nei} \in \Omega(\mathbf{p})} \left| \overline{\overline{\mathbf{W}}}_{\mathbf{p}_{nei}}(\mathbf{p}) \right|, \tag{17}$$

where $\mathbf{W}$ denotes the initial affinity weight predicted by the network, γ denotes the normalization parameter and is set to 4 based on (Park et al. 2020), $\widehat{\mathbf{C}}$ denotes the confidence map and is acquired by

$$\widehat{\mathbf{C}}(\mathbf{p}) = 1 - \text{sigmoid}(\log \widehat{\sigma}(\mathbf{p})). \tag{18}$$

The sigmoid activation is used to map the values of $\widehat{\sigma}(\mathbf{p})$ to the range of 0 to 1. We set the number of non-local neighbors as 8 following (Park et al. 2020) and perform the NLSPN refinement twice at each update stage. Then, we employ the refined depth prediction $\widetilde{\mathbf{D}}(\mathbf{p})$ to identify the target bin

$$e_n \leq \widetilde{\mathbf{D}}(\mathbf{p}) < e_{n+1}, \tag{19}$$

and further acquire the elastic target bin $[e_n - 0.5\widehat{\sigma}(\mathbf{p}), e_{n+1} + 0.5\widehat{\sigma}(\mathbf{p})]$ by adding $\widehat{\sigma}(\mathbf{p})$. Next, similar to monocular depth estimation, we revise $d_{\min}$ and $d_{\max}$ as $e_n - 0.5\widehat{\sigma}(\mathbf{p})$ and $e_{n+1} + 0.5\widehat{\sigma}(\mathbf{p})$, respectively, and use Eqs.1, 2, 3 and 4 to obtain the updated depth candidates and prediction. The update stage is repeated upto the final step.

Compared with direct usage of linearly combined depth prediction to acquire the depth candidates, NLSPN refined depth map enables IEBins to determine more reasonable target bins, while IEBins is able to deliver the uncertainty map for NLSPN refinement to reduce the negative impact of unreliable depth values. Hence, the IEBins and NLSPN refinement are closely integrated in a complementary way.

4.2 Completion Framework

Feature extractor. We employ the network architecture proposed by (Zhang et al. 2023), in which multiple joint convolutional attention and Transformer (JCAT) blocks are employed to fully exploit both the local details and global correlations. It is built upon an encoder-decoder structure with skip connections. The encoder receives an RGB image and a sparse depth map, which are mapped into high-dimensional features using an RGB and depth embedding module. The mapped features pass through two ResNet blocks (He et al. 2016) and four JCAT blocks to generate a five-layer feature pyramid. During the decoding phase, five convolutional attention blocks (Woo et al. 2018) and deconvolution layers are used to process features and recover resolutions, respectively. The feature maps at full resolution from both the encoder and decoder are then fed into the iterative optimizer.

Iterative optimizer. In compliance with the non-local spatial propagation network (Park et al. 2020), the iterative optimizer operates at full resolution. Since deploying GRU at the full resolution to assist probabilistic distribution prediction would incur huge memory consumption even though it may lead to higher accuracy, we only adopt three 3×3 convolutional layers here. The first layer is used to project the depth candidates into the feature space, regularized with the ReLU activation and the last two layers are used to merge the projected features with the features from the feature extractor to output a probabilistic distribution, regularized by the ReLU and Softmax activations, respectively. It should be noted that the parameters of these three layers are shared across different stages. Besides, we use two 3×3 convolutional layers to predict the initial affinity weights from features provided by the feature extractor following (Park et al. 2020). An overview for the whole depth completion framework is presented in Fig. 4.

4.3 Training Loss Function

Pixel-wise depth loss. We adopt a combined L1 and L2 loss for depth supervision (Park et al. 2020; Zhang et al. 2023),

$$\mathcal{L}_{\mathbf{D}} = \sum_{k=1}^{K} \frac{1}{|\mathbf{T}|} \sum_{\mathbf{p}} \left(\left| \widehat{\mathbf{D}}(\mathbf{p}) - \mathbf{D}^{gt}(\mathbf{p}) \right| + \left| \widehat{\mathbf{D}}(\mathbf{p}) - \mathbf{D}^{gt}(\mathbf{p}) \right|^2 \right), \tag{20}$$

where K is set 3.

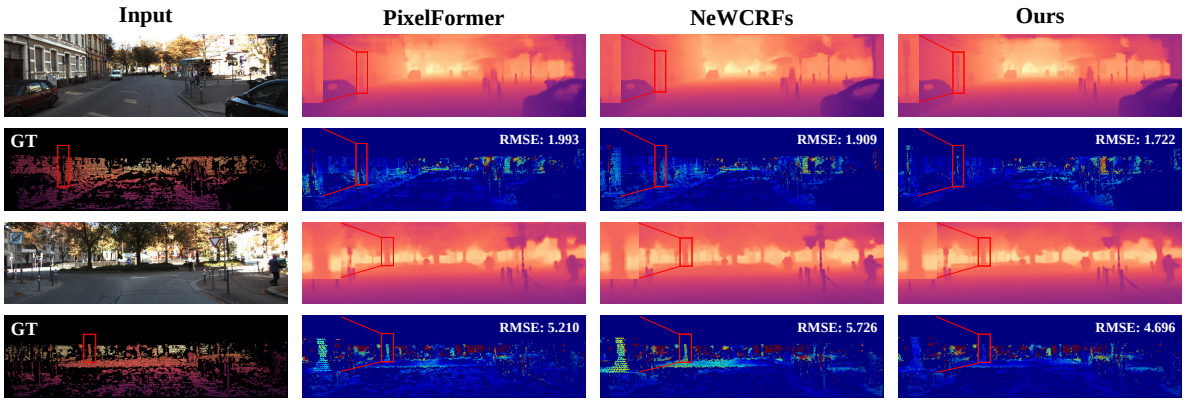

Fig. 5 Qualitative depth estimation comparison on the Eigen split of KITTI dataset. The first and third rows display depth predictions, while the second and fourth rows show the corresponding error maps. In the error maps, large errors are highlighted in orange or red, while small errors are highlighted in blue. The red boxes indicate the emphasized regions.

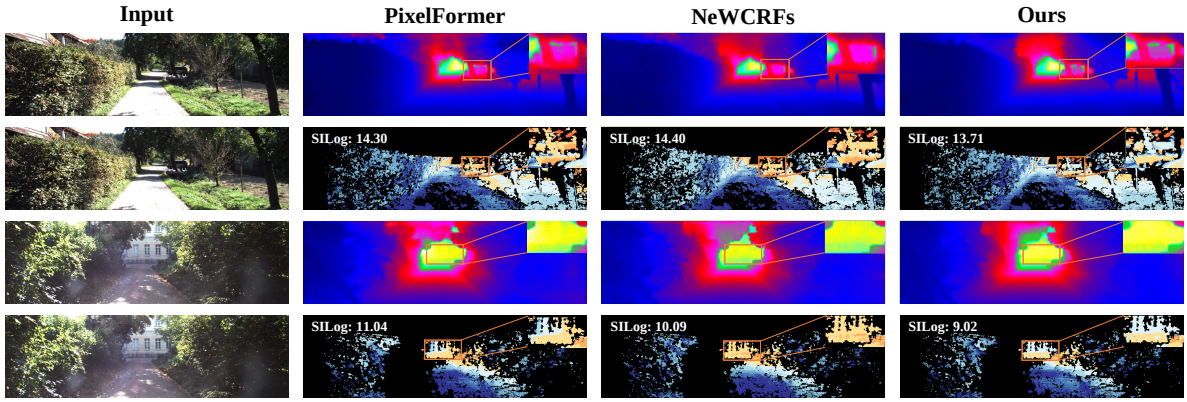

Fig. 6 Qualitative depth estimation comparison on the official split of KITTI dataset. The first and third rows represent depth predictions, while the second and fourth rows stand for corresponding error maps, with large errors highlighted in orange or red. The orange boxes indicate regions of emphasis.

5 Experiment

5.1 Datasets and Evaluation Metrics

We assess the proposed method utilizing datasets from both outdoor and indoor scenes, comprising KITTI (Geiger et al. 2013), NYU-Depth-v2 (Silberman et al. 2012), SUN RGB-D (Song et al. 2015), ScanNet (Dai et al. 2017a) and DIODE (Vasiljevic et al. 2019).

KITTI comprises data captured in outdoor environments using equipment mounted on a moving vehicle, which provides stereo images, sparse depth maps, and corresponding ground-truth depth information. The image resolutions are not always the same, around 1241×376 pixels. For monocular depth estimation, we utilize two commonly adopted data splits: the Eigen split, consisting of 23488 training images and 697 test images (Eigen et al. 2014), and the official data split, which includes 85898 training images, 1000 validation images, and 500 test images. Regarding depth completion, we adopt the official split, encompassing 85898 training images, 1000 validation images, and 1000 test images. By taking into consideration the lack of LiDAR returns at the top of the depth map, we further apply a bottom center-cropping operation during training, following (Park et al. 2020; Zhang et al. 2023). This operation results in a reduced resolution of 1216×240. Note that the official test set lacks ground-truth depth maps, and the corresponding evaluation results are provided by the online server.

NYU-Depth-v2 consists of data collected in indoor environments using Microsoft Kinect, which offers images and ground-truth depth maps. The resolution is 640×480 pixels. For monocular depth estimation, we employ 36253 images for training and 654 images for testing, adhering to (Agarwal & Arora 2023; Yuan et al. 2022). Both the training and test images have a resolution of 640×480 pixels. Regarding depth completion, we utilize 50000 images for training and 654 images for testing, as in (Park et al. 2020; Zhang et al. 2023). Besides, during both training and testing phases, the images and ground-truth depth maps undergo a process of half-downsampling and center-cropping, resulting in a resolution of 304×288 pixels. The required sparse depth maps are generated by randomly sampling from the corresponding ground-truth depth maps in line with (Park et al. 2020; Zhang et al. 2023).

SUN RGB-D and **ScanNet** are sourced from indoor environments. We use the official test set of SUN RGB-D comprising 5050 images, and a sampled test set (Ke et al. 2024) of ScanNet consisting of 800 images. **DIODE** is collected from both indoor and outdoor scenarios. We adopt the official validation set, which contains 325 indoor samples and 446 outdoor samples. These three datasets are employed for the zero-shot generalization study.

Method	Backbone	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓	$\delta_{1.25}$ ↑	$\delta_{1.25^2}$ ↑	$\delta_{1.25^3}$ ↑
DORN (Fu et al. 2018)	ResNet-101	0.072	0.307	2.727	0.120	0.932	0.984	0.994
VNL (Yin et al. 2019)	ResNeXt-101	0.072	-	3.258	0.117	0.938	0.990	0.998
BTS (Lee et al. 2019)	DenseNet-161	0.060	0.249	2.798	0.096	0.955	0.993	0.998
PWA (Lee et al. 2021b)	ResNeXt-101	0.060	0.221	2.604	0.093	0.958	0.994	**0.999**
TransDepth (Yang et al. 2021)	R-50+ViT-B/16†	0.064	0.252	2.755	0.098	0.956	0.994	**0.999**
AdaBins (Bhat et al. 2021)	E-B5+mini-ViT	0.058	0.190	2.360	0.088	0.964	0.995	**0.999**
P3Depth (Patil et al. 2022)	ResNet-101	0.071	0.270	2.842	0.103	0.953	0.993	0.998
NeWCRFs (Yuan et al. 2022)	Swin-Large†	0.052	0.155	2.129	0.079	0.974	0.997	**0.999**
BinsFormer (Li et al. 2022)	Swin-Tiny	0.058	0.183	2.286	0.088	0.968	0.995	**0.999**
BinsFormer (Li et al. 2022)	Swin-Large†	0.052	0.151	2.098	0.079	0.974	0.997	**0.999**
PixelFormer (Agarwal & Arora 2023)	Swin-Large†	0.051	0.149	2.081	0.077	0.976	0.997	**0.999**
Ours	Swin-Tiny	0.056	0.169	2.205	0.084	0.970	0.996	**0.999**
Ours	Swin-Large†	**0.050**	**0.142**	**2.011**	**0.075**	**0.978**	**0.998**	**0.999**

Table 1 Quantitative depth estimation comparison on the Eigen split of KITTI dataset. We present the results of the proposed method using Swin-Large and Swin-Tiny backbones (Liu et al. 2021c). The maximum depth is capped at 80m. R-50 and E-B5 abbreviate ResNet-50 (He et al. 2016) and EfficientNet-B5 (Tan & Le 2019), respectively. † indicates models pre-trained on ImageNet-22K. '-' denotes not applicable. The best results are highlighted in **bold**.

Method	dataset	SILog ↓	Abs Rel	Sq Rel ↓	iRMSE ↓	RMSE ↓	$\delta_{1.25}$ ↑	$\delta_{1.25^2}$ ↑	$\delta_{1.25^3}$ ↑
DORN (Fu et al. 2018)	validation	12.22	11.78	3.03	11.68	3.80	0.913	0.985	0.995
BTS (Lee et al. 2019)	validation	10.67	7.51	1.59	8.10	3.37	0.938	0.987	0.996
BA-Full (Aich et al. 2021)	validation	10.64	8.25	1.81	8.47	3.30	0.938	0.988	0.997
NeWCRFs (Yuan et al. 2022)	validation	8.31	5.54	0.89	6.34	2.55	0.968	0.995	0.998
Ours	validation	**7.58**	**5.10**	**0.75**	**5.90**	**2.37**	**0.974**	**0.996**	**0.999**
DORN (Fu et al. 2018)	online test	11.77	8.78	2.23	12.98	-	-	-	-
BTS (Lee et al. 2019)	online test	11.67	9.04	2.21	12.23	-	-	-	-
BA-Full (Aich et al. 2021)	online test	11.61	9.38	2.29	12.23	-	-	-	-
PWA (Lee et al. 2021b)	online test	11.45	9.05	2.30	12.32	-	-	-	-
Vip-Deeplab (Qiao et al. 2021)	online test	10.80	8.94	2.19	11.77	-	-	-	-
NeWCRFs (Yuan et al. 2022)	online test	10.39	8.37	1.83	11.03	-	-	-	-
PixelFormer (Agarwal & Arora 2023)	online test	10.28	8.16	1.82	10.84	-	-	-	-
BinsFormer (Li et al. 2022)	online test	10.14	8.23	1.69	10.90	-	-	-	-
Ours	online test	**9.63**	**7.82**	**1.60**	**10.68**	-	-	-	-

Table 2 Quantitative depth estimation comparison on the official split of KITTI dataset, where SILog serves as the primary ranking metric.

Evaluation metrics. Similar to previous works (Agarwal & Arora 2023; Yuan et al. 2022), we adopt metrics SILog, Abs Rel, Sq Rel, RMSE, iRMSE, RMSE log, $\log_{10}$, $\delta_{1.25}$, $\delta_{1.25^2}$ and $\delta_{1.25^3}$ for monocular depth estimation. Following (Park et al. 2020), we adopt metrics REL, RMSE, iRMSE, MAE, iMAE, $\delta_{1.25}$, $\delta_{1.25^2}$ and $\delta_{1.25^3}$ for depth completion.

5.2 Implementation Details

Our frameworks are implemented via PyTorch (Paszke et al. 2017) and trained on NVIDIA A5000 24GB GPUs. For monocular depth estimation, the total number of epochs is 20. We adopt the Adam optimizer (Kingma & Ba 2015) where $\beta_1 = 0.9$, $\beta_2 = 0.999$. The batch size is set to 8 and the learning rate follows a polynomial decay schedule, gradually reduced from 2e-5 to 2e-6. As for depth completion, we employ the AdamW optimizer (Loshchilov & Hutter 2018) where β_1 and β_2 are set to 0.9 and 0.999, respectively. The batch size is set to 12 and the initial learning rate is 0.001. The model undergoes training for 72 epochs on the NYU-Depth-v2 dataset, with the learning rate halved at epochs 36, 48, 56, and 64. In terms of the KITTI dataset, the training spans 100 epochs, and the learning rate experiences decay by a factor of 0.5 at epochs 50, 60, 70, 80, and 90.

5.3 Monocular Depth Estimation

5.3.1 Comparison to Previous State-of-the-art Competitors

KITTI. Table 1 demonstrates results on the Eigen split. As evident, the proposed method surpasses the leading competitors by a considerable margin. Notably, when compared to PixelFormer and BinsFormer, both of which are also rooted in the classification-regression paradigm, our method shows superior performance. Table 2 summarizes results on the official split. As can be seen, the proposed method consistently outperforms prior leading competitors. It is noteworthy that

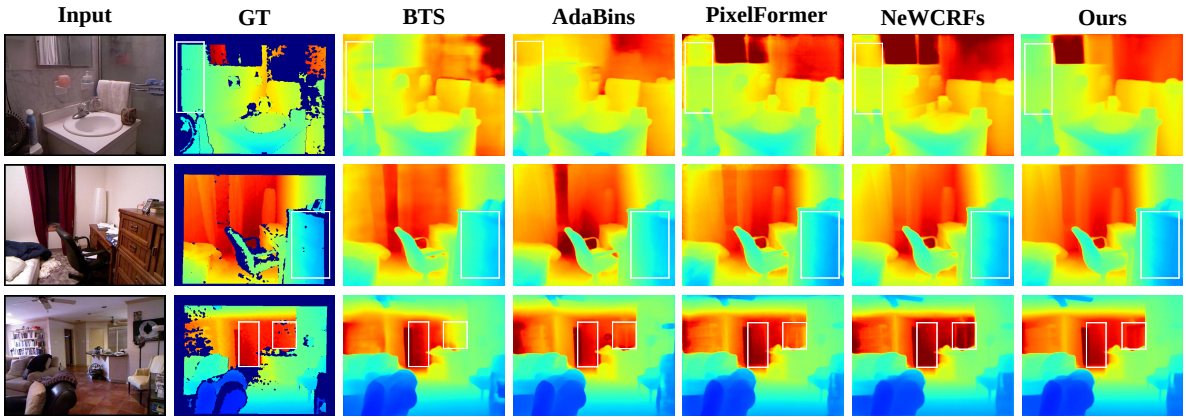

Fig. 7 Qualitative depth estimation comparison on the NYU-Depth-v2 dataset, with white boxes indicating the highlighted regions.

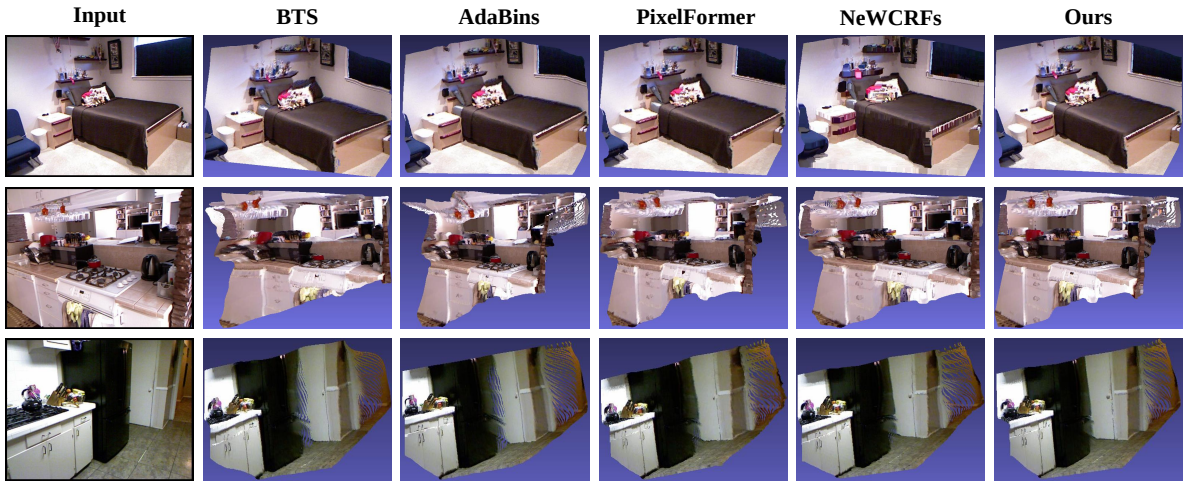

Fig. 8 Qualitative point cloud comparison for depth estimation on the NYU-Depth-v2 dataset.

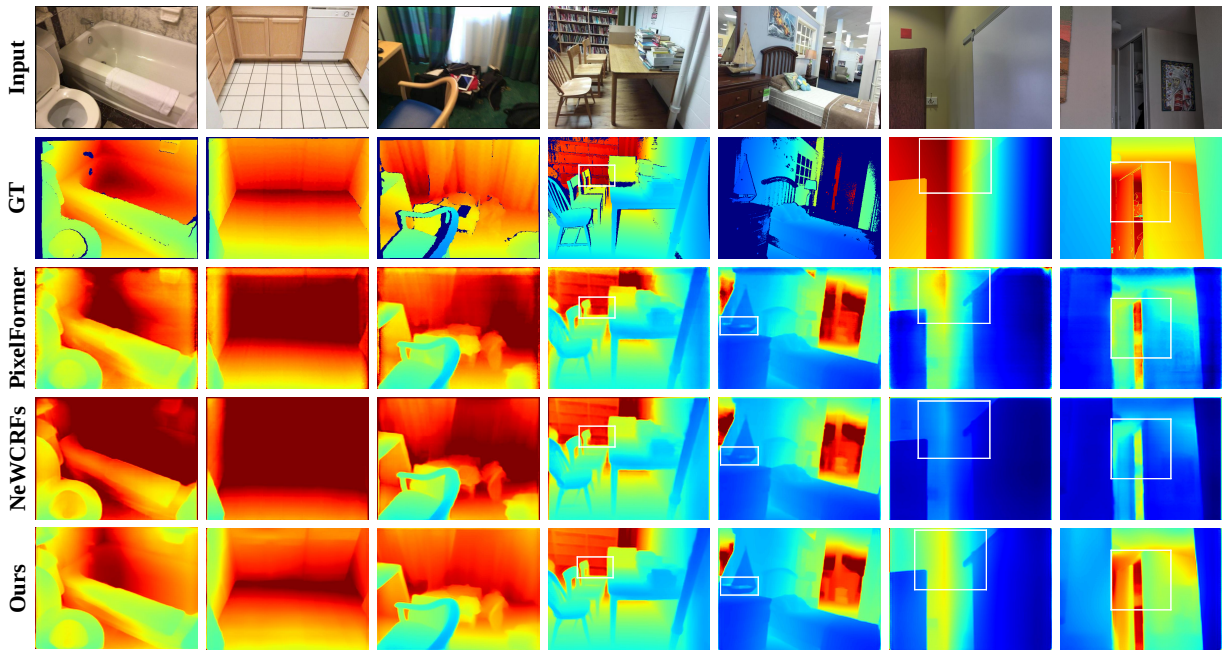

Fig. 9 Qualitative depth estimation comparison on the SUN RGB-D, ScanNet and DIODE Indoor datasets. From left to right, the first three samples are from ScanNet, the fourth and fifth samples are from SUN RGB-D, and the last two samples are from DIODE Indoor. The white boxes indicate the highlighted regions.

the primary ranking metric, SILog, experiences a significant reduction.

In Fig. 5, we demonstrate a qualitative depth comparison on the Eigen split. It can be seen that the proposed method

Method	Backbone	Abs Rel ↓	Sq Rel ↓	RMSE ↓	$\log_{10}$ ↓	$\delta_{1.25}$ ↑	$\delta_{1.25^2}$ ↑	$\delta_{1.25^3}$ ↑
DORN (Fu et al. 2018)	ResNet-101	0.115	-	0.509	0.051	0.828	0.965	0.992
VNL (Yin et al. 2019)	ResNeXt-101	0.108	-	0.416	0.048	0.875	0.976	0.994
BTS (Lee et al. 2019)	DenseNet-161	0.110	0.066	0.392	0.047	0.885	0.978	0.994
PWA (Lee et al. 2021b)	DenseNet-161	0.105	-	0.374	0.045	0.892	0.985	0.997
Long *et al.* (Long et al. 2021)	HRNet-48	0.101	-	0.377	0.044	0.890	0.982	0.996
TransDepth (Yang et al. 2021)	R-50+ViT-B/16†	0.106	-	0.365	0.045	0.900	0.983	0.996
AdaBins (Bhat et al. 2021)	E-B5+mini-ViT	0.103	-	0.364	0.044	0.903	0.984	0.997
P3Depth (Patil et al. 2022)	ResNet-101	0.104	-	0.356	0.043	0.898	0.981	0.996
LocalBins (Bhat et al. 2022)	E-B5	0.099	-	0.357	0.042	0.907	0.987	**0.998**
NeWCRFs (Yuan et al. 2022)	Swin-Large†	0.095	0.045	0.334	0.041	0.922	**0.992**	**0.998**
BinsFormer‡ (Li et al. 2022)	Swin-Tiny	0.113	-	0.379	0.047	0.890	0.983	0.996
BinsFormer‡ (Li et al. 2022)	Swin-Large†	0.094	-	0.330	0.040	0.925	0.989	0.997
PixelFormer (Agarwal & Arora 2023)	Swin-Large†	0.090	-	0.322	0.039	0.929	0.991	**0.998**
Ours	Swin-Tiny	0.108	0.061	0.375	0.046	0.893	0.984	0.996
Ours	Swin-Large†	**0.087**	**0.040**	**0.314**	**0.038**	**0.936**	**0.992**	**0.998**

Table 3 Quantitative depth estimation comparison on the NYU-Depth-v2 dataset, with the maximum depth capped at 10m. ‡ indicates that the model is trained using auxiliary scene class information.

Dataset	Method	Backbone	Add. Info	Abs Rel ↓	RMSE ↓	$\delta_{1.25}$
KITTI	NDDepth (Shao et al. 2023a)	Swin-Large†	Planar Priors	0.050	2.025	**0.978**
	GEDepth (Yang et al. 2023)	Swin-Large†	Camera Parameters	**0.048**	2.050	-
	MAMo (Yasarla et al. 2023)	Swin-Large†	Multiple Frames	0.049	**1.884**	0.977
	Ours	Swin-Large†	-	0.050	2.011	**0.978**
NYU-Depth-v2	VPD (Zhao et al. 2023)	Stable Diffusion	Text Description	**0.069**	**0.254**	**0.964**
	AiT (Ning et al. 2023)	Swinv2-Large★	-	0.079	0.284	0.949
	NVDS (Wang et al. 2023)	DPT	Multiple Frames	0.081	-	0.931
	NDDepth (Shao et al. 2023a)	Swin-Large†	Planar Priors	0.087	0.311	0.936
	Ours	Swin-Large†	-	0.087	0.314	0.936

Table 4 Quantitative depth estimation comparison against concurrent works on the KITTI and NYU-Depth-v2 datasets. ★ indicates that the backbone is self-supervised pre-trained. Add. Info: additional information; Swin-Large (Liu et al. 2021c); Swinv2-Large (Liu et al. 2022b); Stable Diffusion (Rombach et al. 2022); DPT (Ranftl et al. 2021).

can provide more accurate depth predictions for the outlines of traffic sign poles and trees. In Fig. 6, we further present a qualitative depth comparison on the official split, where the corresponding error maps generated by the online server are attached. The proposed method is capable of more accurately delineating the outline of the car in the distance.

NYU-Depth-v2. The results, presented in Table 3, showcase the consistent superiority of the proposed method over prior competing competitors, with notable improvements across most metrics. Specifically, it outperforms BinsFormer by 7.4% and 4.8% on the Abs Rel and RMSE metrics, respectively, underscoring the effectiveness of our method. In Fig. 7, we provide a qualitative depth comparison. The proposed method exhibits the preservation of fine-grained details, such as boundaries, and generates more continuous depth values in the planar regions. In order to further evaluate the depth map quality from the 3D shape, we convert the depth maps into point clouds and present a qualitative point cloud comparison in Fig. 8. The proposed method shows fewer distortions than the compared competitors and recovers the structures of the 3D world reasonably.

5.3.2 Comparison to concurrent works

In Table 4, we demonstrate a quantitative comparison against concurrent works on the KITTI and NYU-Depth-v2 datasets. As can be seen, most of these works employ additional information, such as text, camera parameters, etc, to boost performance. In contrast, the proposed method requires only a single RGB image as input and the corresponding ground-truth depth map during the training phase. Nevertheless, the proposed method achieves comparable performance and even shows strengths on some metrics.

5.3.3 Zero-shot Generalization

In line with the approach of prior studies (Bhat et al. 2021; Li et al. 2022), we undertake a cross-dataset evaluation in a challenging zero-shot setting. To this end, we leverage the SUN RGB-D, ScanNet and DIODE datasets, which have not been seen during the training phase. For evaluation on the SUN RGB-D, ScanNet, and DIODE Indoor datasets, the models are trained on the NYU-Depth-v2 dataset. As for the

Method	Backbone	Abs Rel ↓	RMSE ↓	$\log_{10}$ ↓	$\delta_{1.25}$ ↑	$\delta_{1.25^2}$ ↑	$\delta_{1.25^3}$ ↑
Chen *et al.* (Chen et al. 2019b)	SENet	0.166	0.494	0.071	0.757	0.943	0.984
VNL (Yin et al. 2019)	ResNeXt-101	0.183	0.541	0.082	0.696	0.912	0.973
BTS (Lee et al. 2019)	DenseNet-161	0.172	0.515	0.075	0.740	0.933	0.980
AdaBins (Bhat et al. 2021)	E-B5+Mini-ViT	0.159	0.476	0.068	0.771	0.944	0.983
LocalBins (Bhat et al. 2022)	E-B5	0.156	0.470	0.067	0.777	0.949	0.985
NeWCRFs (Yuan et al. 2022)	Swin-Large†	0.151	0.424	0.064	0.798	0.967	0.992
PixelFormer (Agarwal & Arora 2023)	Swin-Large†	0.144	0.441	0.062	0.802	0.962	0.990
BinsFormer‡ (Li et al. 2022)	Swin-Tiny	0.162	0.478	0.069	0.760	0.945	0.985
BinsFormer‡ (Li et al. 2022)	Swin-Large†	0.143	0.421	0.061	0.805	0.963	0.990
Ours	Swin-Tiny	0.157	0.476	0.069	0.768	0.950	0.987
Ours	Swin-Large†	**0.135**	**0.405**	**0.059**	**0.822**	**0.971**	**0.993**

Table 5 Generalized depth estimation comparison on the SUN RGB-D dataset in a zero-shot setting.

Dataset	Method	Backbone	Abs Rel ↓	RMSE ↓	$\log_{10}$ ↓	$\delta_{1.25}$ ↑	$\delta_{1.25^2}$ ↑	$\delta_{1.25^3}$ ↑
ScanNet	NeWCRFs (Yuan et al. 2022)	Swin-Large†	0.192	0.357	0.076	0.716	0.939	0.988
	PixelFormer (Agarwal & Arora 2023)	Swin-Large†	0.169	0.321	0.067	0.775	0.949	0.987
	Ours	Swin-Large†	**0.156**	**0.308**	**0.064**	**0.794**	**0.957**	**0.990**
DIODE (in)	NeWCRFs (Yuan et al. 2022)	Swin-Large†	0.392	1.864	0.240	0.188	0.499	0.749
	PixelFormer (Agarwal & Arora 2023)	Swin-Large†	0.340	1.658	0.196	0.316	0.617	0.793
	Ours	Swin-Large†	**0.327**	**1.614**	**0.187**	**0.348**	**0.639**	**0.796**
DIODE (out)	NeWCRFs (Yuan et al. 2022)	Swin-Large†	0.676	9.039	0.280	0.178	0.371	0.591
	PixelFormer (Agarwal & Arora 2023)	Swin-Large†	0.664	8.224	0.245	0.214	0.447	0.674
	Ours	Swin-Large†	**0.642**	**7.665**	**0.224**	**0.231**	**0.498**	**0.733**

Table 6 Generalized depth estimation comparison on the ScanNet and DIODE datasets in a zero-shot setting. DIODE (in): DIODE Indoor; DIODE (out): DIODE Outdoor.

Seq.	**Ours** (key)	NeW. (key)	Mono. (key)	**Ours** (all)	NeW. (all)
01	**117.06**	536.53	530.09	**125.09**	583.20
02	**12.22**	13.32	287.46	**13.59**	13.97
03	6.72	8.31	**3.56**	**7.15**	9.04
04	16.70	31.56	**1.89**	**16.61**	30.59
05	8.10	**8.05**	20.70	**7.56**	7.86
06	1.32	**0.96**	10.45	1.35	**0.95**
07	**2.48**	3.09	8.09	**2.55**	3.24
08	10.89	**9.82**	140.49	11.06	**9.90**
09	**5.44**	7.61	6.79	**5.68**	7.67
10	**7.21**	11.73	384.49	**8.24**	12.66

Table 7 Quantitative visual odometry comparison on the KITTI odometry dataset. "key" and "all" represent keyframes and all frames, respectively. Seq.: sequence; NeW.:NeWCRFs (Yuan et al. 2022); Mono.: Monocular.

evaluation on the DIODE Outdoor dataset, the models are trained on the KITTI dataset. The results are summarized in Tables 5 and 6. Regarding the SUN RGB-D dataset results, we observe that despite the proposed approach with Swin-Tiny backbone exhibiting slightly lower performance than AdaBins on the NYU-Depth-v2, it outperforms AdaBins on specific metrics, such as Abs Rel. For results on the ScanNet, DIODE Indoor and Outdoor, the proposed method also achieves stronger generalization performance than the compared competitors. In Fig. 9, we provide a qualitative depth comparison. The depth maps predicted by NeWCRFs and PixelFormer for the ScanNet samples exhibit large holes. By contrast, the proposed method shows higher robustness. In addition, the depth maps of the proposed method retain more details, particularly in boundary areas.

5.3.4 Application to SLAM

To indicate the benefits of improvements in downstream tasks such as SLAM, we integrate IEBins and NeWCRFs (Yuan et al. 2022) into ORB-SLAM2 (Mur-Artal & Tardós 2017) in the RGB-D setting and evaluate the visual odometry performance on the KITTI odometry dataset. We demonstrate results for both the keyframes (selected by the ORB-SLAM2) and all frames in sequences 01-10, using the ATE (m) metric. Moreover, we provide the odometry results in the monocular setting, which are available only for keyframes. As demonstrated in Table 7, the proposed approach either significantly outperforms NeWCRFs or achieves comparable performance with the latter. Further, the RGB-D-based SLAM shows superior performance across most sequences compared to the monocular SLAM. In Fig. 10, we visualize the trajectories on sequences 03 and 09. It can be seen that the trajectory of

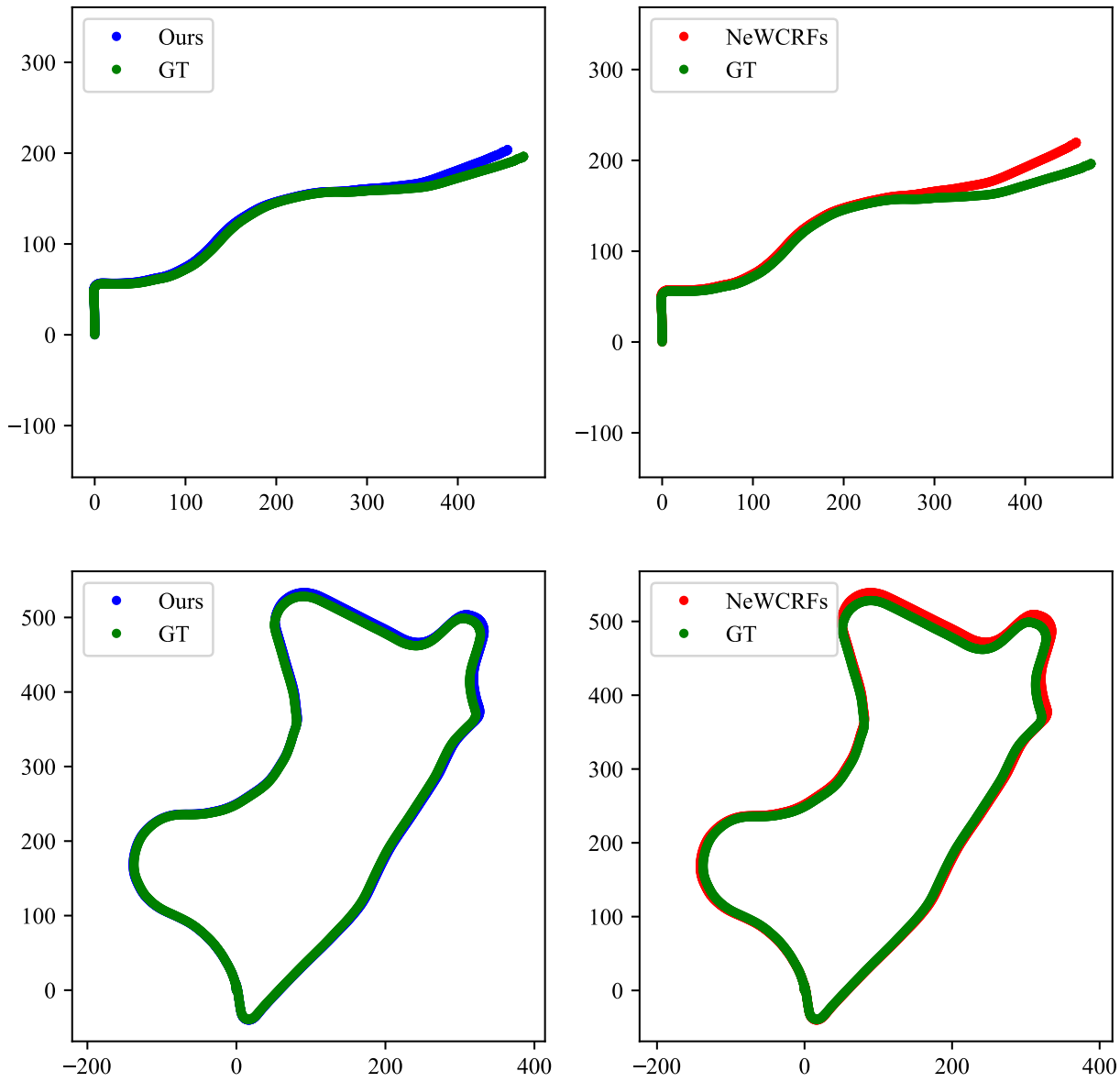

Fig. 10 Qualitative visual odometry comparison on the KITTI odometry dataset. The trajectories of the upper row are from sequence 03, and the trajectories of the lower row are from sequence 09.

Method	Abs Rel ↓	RMSE ↓	$\log_{10}$ ↓	$\delta_{1.25}$ ↑	$\delta_{1.25^2}$ ↑	$\delta_{1.25^3}$ ↑
Baseline + Regression	0.095	0.337	0.041	0.921	0.991	0.998
Baseline + UBins (Fu et al. 2018)	0.091	0.328	0.040	0.925	0.991	**0.998**
Baseline + SIBins (Fu et al. 2018)	0.090	0.326	0.039	0.928	**0.992**	**0.998**
Baseline + AdaBins (Bhat et al. 2021)	0.089	0.320	**0.038**	0.931	0.991	**0.998**
Baseline + LocalBins (Bhat et al. 2022)	0.090	0.319	**0.038**	0.932	**0.992**	**0.998**
Baseline + IBins	0.090	0.317	**0.038**	0.932	0.991	**0.998**
Baseline + IEBins	**0.087**	**0.314**	**0.038**	**0.936**	**0.992**	**0.998**

Table 8 Comparison of different bin types on the NYU-Depth-v2 dataset. The baseline is established by excluding the iterative optimizer from our whole framework. Other bin types include UBins (uniform bins), SIBins (space increasing bins), AdaBins (adaptive bins), LocalBins (local bins), and IBins (iterative bins), where the IBins involves substituting the elastic target bin with the original target bin at each stage. The number of bins for UBins, SIBins, AdaBins, and LocalBins is set to 256, following (Bhat et al. 2021, 2022; Fu et al. 2018).

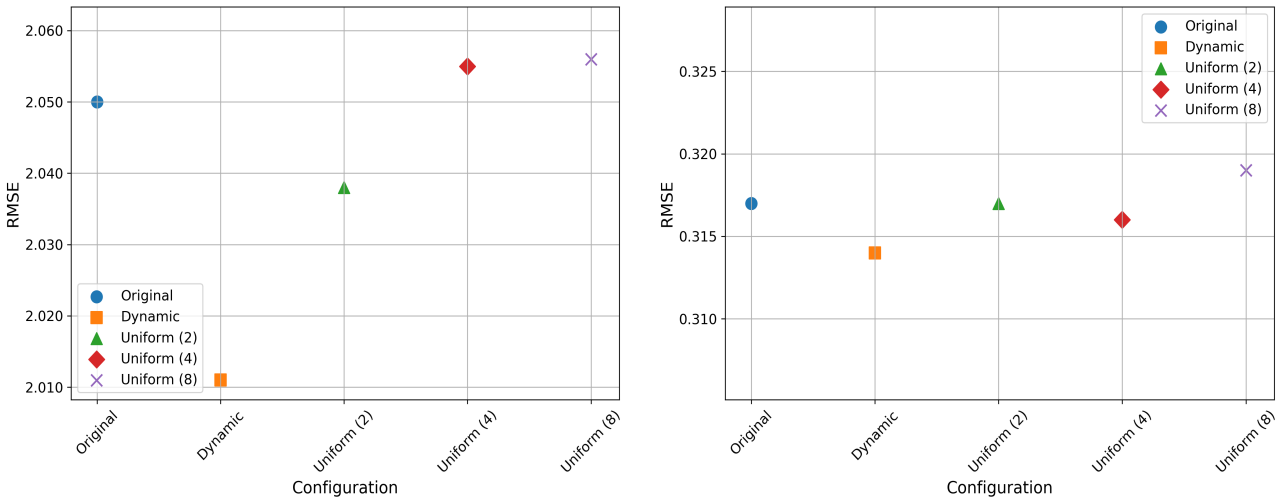

Fig. 11 Comparison of different target bin adjustment strategies. Left: KITTI; Right: NYU-Depth-v2. Original: the original target bin is used as the depth search range for the next stage. Dynamic: the width of target bin is dynamically adapted based on depth uncertainty. Uniform (2), (4) and (8): the width of target bin is uniformly expanded by 2, 4 and 8 times, respectively.

Method	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓	$\delta_{1.25}$ ↑	$\delta_{1.25^2}$ ↑	$\delta_{1.25^3}$ ↑
Baseline + Regression	0.053	0.153	2.129	0.080	0.974	0.997	0.999
Baseline + UBins (Fu et al. 2018)	0.052	0.151	2.095	0.078	0.975	0.997	0.999
Baseline + SIBins (Fu et al. 2018)	0.051	0.147	2.092	0.078	0.976	0.997	0.999
Baseline + AdaBins (Bhat et al. 2021)	0.051	0.146	2.048	0.077	0.976	**0.998**	**0.999**
Baseline + IBins	**0.050**	0.143	2.050	0.076	0.977	**0.998**	**0.999**
Baseline + IEBins	**0.050**	**0.142**	**2.011**	**0.075**	**0.978**	**0.998**	**0.999**

Table 9 Comparison of different bin types on the KITTI dataset.

Num. B.	Abs Rel ↓	RMSE ↓	$\log_{10}$ ↓	$\delta_{1.25}$ ↑	$\delta_{1.25^2}$ ↑	$\delta_{1.25^3}$ ↑
4	0.105	0.335	0.043	0.900	0.982	0.999
8	0.089	0.317	**0.038**	0.934	**0.992**	**0.998**
16	**0.087**	**0.314**	**0.038**	**0.936**	**0.992**	**0.998**
32	0.088	0.317	**0.038**	0.932	**0.992**	**0.998**

Table 10 Impact of different numbers of bins on the NYU-Depth-v2 dataset. Num. B.: number of bins.

Num. B.	Abs Rel ↓	Sq Rel ↓	RMSE ↓	$\delta_{1.25}$ ↑	$\delta_{1.25^2}$ ↑	$\delta_{1.25^3}$ ↑
4	0.288	1.199	4.146	0.574	0.910	0.979
8	0.054	0.147	**2.009**	0.973	0.996	**0.999**
16	**0.050**	**0.142**	2.011	**0.978**	**0.998**	**0.999**
32	**0.050**	**0.142**	2.026	0.977	**0.998**	**0.999**

Table 11 Impact of different bin numbers on the KITTI dataset. Num. B.: number of bins.

Method	Backbone	Abs Rel ↓	Param. (M) ↓	Inf. (s) ↓
NeWCRFs (Yuan et al. 2022)	Swin-Large†	0.095	270	0.052
BinsFormer‡ (Li et al. 2022)	Swin-Large†	0.094	255	0.216
PixelFormer (Agarwal & Arora 2023)	Swin-Large†	0.090	271	0.082
Ours	Swin-Large†	0.087	273	0.085

Table 12 Comparison of model parameters and inference time on the NYU-Depth-v2 dataset. Param.: parameter; Inf.: inference time.

NeWCRFs drifts severely at the end of sequence 03 due to error accumulation, while the trajectory of IEBins is able to highly coincide with the ground-truth.

5.3.5 Ablation Study

To more precisely elucidate the impact of each specific component, we carry out several ablation studies, which are categorized into **IEBins**, **bin types**, **effect of bin numbers** and **target bin adjustment strategies**.

IEBins. We begin by evaluating the importance of IEBins through a comparison with the standard regression and IBins. In order to establish a baseline, we remove the iterative optimizer from our entire framework. In standard regression, the baseline is utilized to directly predict the depth map. For IBins, the elastic target bin is replaced with the original target bin at each stage. As depicted in Tables 8 and 9, standard regression exhibits inferior performance when compared to IBins and IEBins. Equipped with the elastic target bin, more robust IEBins takes the IBins a step further.

Bin types. We further compare IEBins against other alternatives, which include uniform bins (Fu et al. 2018), space increasing bins (Fu et al. 2018), adaptive bins (Bhat et al. 2021), and local bins (Bhat et al. 2022). In order to ensure a fair comparison, we maintain consistency in all other con-

figurations, except for the bin types. As indicated in Tables 8 and 9, the baseline incorporating IEBins exhibits a notable improvement in performance, outperforming other variants.

Effect of bin numbers. In order to investigate the impact of different numbers of bins on performance, we train our model using diverse settings of bin numbers, and present the results in Tables 10 and 11. Initially, the errors demonstrate a sharp decline as the number of bins increases. However, this improvement levels off when the number exceeds 16 bins. Consequently, we opt for 16 bins in our final model.

Target bin adjustment strategies. Apart from the dynamic target bin width adjustment in light of depth uncertainty, we try the uniform expansion. To be specific, we uniformly expand the width of the current target bin by 2 times, 4 times, and 8 times, respectively, for the depth search range in the next stage. As observed in Fig. 11, the uniform expansion achieves lower performance compared to dynamic adjustment and can sometimes degrade performance compared to no adjustment. Although the expansion alleviates error accumulation, it also hinders finer-grained depth search. The proposed elastic mechanism, which adaptively changes the depth search range for every pixel and thereby learns a compact representation, is a more suitable choice.

5.3.6 Model Parameters and Inference Time

Table 12 lists the results on model parameters and inference time for the proposed method, NeWCRFs, BinsFormer and PixelFormer, all using the Swin-Large backbone. The inference time is calculated on the NYU-Depth-v2 test set using a batch size of 1. As we can see, the proposed method has almost the identical number of parameters as NeWCRFs and PixelFormer but slightly more than BinsFormer. Nevertheless, the proposed method achieves an inference time almost 60% faster than BinsFormer, despite both being classification-regression-based methods. At the same time, the proposed method attains superior performance than these three competitors.

5.4 Depth Completion

5.4.1 Comparison to Prior State-of-the-art Competitors

NYU-Depth-v2. Table 13 presents a quantitative depth comparison. As can be seen, the proposed method surpasses previous leading competitors. In particular, it improves CFormer by 3.3% and 8.3% on the RMSE and REL metrics, respectively. Fig. 13 presents a qualitative depth comparison. The proposed method preserves fine-grained details and is able to alleviate the depth mixing issue. For example, in the highlighted region of the last row, the ground depth predicted by CFormer is mixed with the chair depth, while the depth map predicted by the proposed method accurately delineates the chair boundary. We further convert the depth predictions into point clouds and display a qualitative point cloud comparison in Fig. 14. Our point clouds exhibit fewer distortions and preserve the prominent geometric features. Notably, in some scenarios, such as the object highlighted using white boxes in the first column, the recovered point clouds from ground-truth and CFormer both demonstrate severe distortion, while the proposed method effectively mitigates such instances of failure.

Method	RMSE	REL	$\delta_{1.25}$	$\delta_{1.25^2}$	$\delta_{1.25^3}$
S2D (Ma & Karaman 2018)	0.230	0.044	97.1	99.4	99.8
DepthCoeff (Imran et al. 2019)	0.118	0.013	99.4	**99.9**	-
CSPN (Cheng et al. 2018)	0.117	0.016	99.2	**99.9**	**100.0**
CSPN++ (Cheng et al. 2020)	0.116	-	-	-	-
DeepLiDAR (Qiu et al. 2019)	0.115	0.022	99.3	**99.9**	**100.0**
DepthNormal (Xu et al. 2019)	0.112	0.018	99.5	**99.9**	**100.0**
ACMNet (Zhao et al. 2021)	0.105	0.015	99.4	**99.9**	**100.0**
PRNet (Lee et al. 2021a)	0.104	0.014	99.4	**99.9**	**100.0**
NLSPN (Park et al. 2020)	0.092	0.012	**99.6**	**99.9**	**100.0**
TWISE (Imran et al. 2021)	0.097	0.013	**99.6**	**99.9**	**100.0**
AGG-Net (Chen et al. 2023)	0.092	0.014	99.4	**99.9**	**100.0**
RigNet (Yan et al. 2022)	0.090	0.013	**99.6**	**99.9**	**100.0**
DySPN (Lin et al. 2022)	0.090	0.012	**99.6**	**99.9**	**100.0**
CFormer (Zhang et al. 2023)	0.090	0.012	**99.6**	**99.9**	**100.0**
Ours	**0.087**	**0.011**	**99.6**	**99.9**	**100.0**

Table 13 Quantitative depth completion comparison on the NYU-Depth-v2 dataset. Notably, S2D (Ma & Karaman 2018) employs 200 sampled depth points per image as input, in contrast to the other methods, which utilize 500.

Method	RMSE↓ (mm)	MAE↓ (mm)	iRMSE↓ (1/km)	iMAE↓ (1/km)
CSPN (Cheng et al. 2018)	1019.64	279.46	2.93	1.15
S2D (Ma & Karaman 2018)	814.73	249.95	2.80	1.21
DepthNormal (Xu et al. 2019)	777.05	235.17	2.42	1.13
DeepLiDAR (Qiu et al. 2019)	758.38	226.50	2.56	1.15
FuseNet (Chen et al. 2019c)	752.88	221.19	2.34	1.14
CSPN++ (Cheng et al. 2020)	743.69	209.28	2.07	0.90
NLSPN (Park et al. 2020)	741.68	199.59	1.99	0.84
ACMNet (Zhao et al. 2021)	744.91	206.09	2.08	0.90
TWISE (Imran et al. 2021)	840.20	195.58	2.08	**0.82**
PointDC (Yu et al. 2023)	736.07	201.87	1.97	0.87
RigNet (Yan et al. 2022)	712.66	203.25	2.08	0.90
GuideFormer (Rho et al. 2022)	721.48	207.76	2.14	0.97
DySPN (Lin et al. 2022)	709.12	192.71	**1.88**	**0.82**
CFormer (Zhang et al. 2023)	708.87	203.45	2.01	0.88
Ours	**700.33**	**192.54**	1.90	**0.82**

Table 14 Quantitative depth completion comparison on the KITTI dataset, where RMSE serves as the primary ranking metric.

KITTI. Table 14 demonstrates a quantitative depth comparison. As can be seen, the proposed method outperforms the compared competitors on the main ranking metric RMSE and beats recent CFormer in all metrics. In Fig. 12, we present qualitative results obtained from the online server, indicating that the proposed method derives depth maps of higher quality compared to those generated by CFormer, for example, thinner poles.

5.4.2 Sparsity Study

To indicate the robustness of our method across diverse levels of data sparsity, we deliberately generate sparse data by manipulating different configurations for training and testing. Specifically, for the NYU-Depth-v2 dataset, we perform

ID	Setting	RMSE ↓	REL ↓	$\delta_{1.25}$ ↑	$\delta_{1.25^2}$ ↑	$\delta_{1.25^3}$ ↑
1	Baseline + Regression	0.098	0.016	99.5	**99.9**	**100.0**
2	Baseline + Regression + NLSPN	0.090	0.012	**99.6**	**99.9**	**100.0**
3	Baseline + IBins	0.088	0.012	**99.6**	**99.9**	**100.0**
4	Baseline + IEBins	**0.087**	**0.011**	**99.6**	**99.9**	**100.0**
5	Baseline + IEBins (w/o NLSPN)	0.093	0.014	**99.6**	**99.9**	**100.0**
6	Baseline + IEBins (w/ confidence (Park et al. 2020))	0.089	0.012	**99.6**	**99.9**	**100.0**

Table 15 Ablation study for depth completion. The baseline is built by removing the iterative optimizer from our entire framework. NLSPN: non-local spatial propagation network (Park et al. 2020). We note that IBins and IEBins for depth completion include the NLSPN refinement.

Method		RMSE↓				
		PackNet-SAN	GuideNet	NLSPN	CFormer	**Ours**
Sample Number	0	-	-	0.562	0.490	**0.480**
	50	-	-	0.223	0.208	**0.197**
	200	0.155	0.142	0.129	0.127	**0.124**
	500	0.120	0.101	0.092	0.090	**0.087**

Table 16 Sparsity study on the NYU-Depth-v2 dataset. The study is performed with 0, 50, 200, and 500 samples. The methods compared include PackNet-SAN (Guizilini et al. 2021), GuideNet (Tang et al. 2020), NLSPN (Park et al. 2020), and CFormer (Zhang et al. 2023).

Dataset	RMSE↓		
	NLSPN (Park et al. 2020)	CFormer (Zhang et al. 2023)	**Ours**
SUN RGB-D	0.115	0.115	**0.113**
ScanNet	0.070	0.070	**0.068**

Table 17 Generalized depth completion comparison on the SUN RGB-D and ScanNet datasets in a zero-shot setting.

Scanning Lines	Method	RMSE↓ (mm)	MAE↓ (mm)	iRMSE↓ (1/km)	iMAE↓ (1/km)
1	NLSPN	3507.7	1849.1	13.8	8.9
	DySPN	3625.5	1924.7	13.8	8.9
	CFormer	3250.2	1582.6	10.4	6.6
	Ours	**3090.6**	**1487.4**	**9.8**	**6.5**
4	NLSPN	2293.1	831.3	7.0	3.4
	DySPN	2285.8	834.3	6.3	3.2
	CFormer	2150.0	740.1	5.4	2.6
	Ours	**2133.4**	**703.3**	**5.2**	**2.4**
16	NLSPN	1288.9	377.2	3.4	1.4
	DySPN	1274.8	366.4	3.2	1.3
	CFormer	1218.6	337.4	3.0	**1.2**
	Ours	**1178.2**	**324.3**	**2.9**	**1.2**
64	NLSPN	889.4	238.8	2.6	1.0
	DySPN	878.5	228.6	2.5	1.0
	CFormer	848.7	215.9	2.5	**0.9**
	Ours	**812.4**	**207.2**	**2.3**	**0.9**

Table 18 Sparsity study on the KITTI dataset, which is performed under settings of 1, 4, 16 and 64 scanning lines. The approaches compared include NLSPN (Park et al. 2020), DySPN (Lin et al. 2022) and CFormer (Zhang et al. 2023).

random sampling and extract 0, 50, 200, and 500 points from the ground-truth depth map to emulate different sparsity levels. In the case of the KITTI dataset, we follow the approach outlined in (Imran et al. 2021), where we subsample the raw LiDAR scans in azimuth-elevation space, reducing them to 16, 4, and 1 lines. Furthermore, we employ the training set of 10000 images provided by (Zhang et al. 2023) and evaluate the models on KITTI validation set. The results of other approaches are sourced from (Zhang et al. 2023). As listed in Tables 16 and 18, the proposed method maintains good robustness when facing data with different sparsity levels and outperforms prior leading competitors in all cases.

5.4.3 Zero-shot Generalization

In Table 17, we provide a zero-shot generalized depth completion comparison on the SUN RGB-D and ScanNet datasets. The models are trained on the NYU-Depth-v2 dataset. It can be seen that the proposed method achieves better generalization performance than the compared competitors. Fig. 15 displays a qualitative depth comparison. We notice that CFormer tends to suffer from the depth mixing in regions where the depth changes dramatically, for example, boundaries, while the proposed approach avoids this phenomenon. Moreover, the depth predictions from the proposed method demonstrate better continuity.

5.4.4 Ablation Study

In Table 15, we summarize the results of ablation study designed to elucidate the specific impact of each key component. As can be seen, the proposed IEBins shows superiority over standard regression, both with and without NLSPN refinement (IDs 1,5 and IDs 2,4). The elastic target bin contributes to performance improvement by dynamically adapting the target bin width based on the depth uncertainty (IDs 3 and 4). Moreover, replacing the confidence employed in affinity modulation of NLSPN with the original confidence, or removing the NLSPN refinement from the update stages of IEBins, the performance suffers different degrees of degradation, which indicates the effectiveness of tightly combining IEBins and NLSPN (IDs 4,5 and 6).

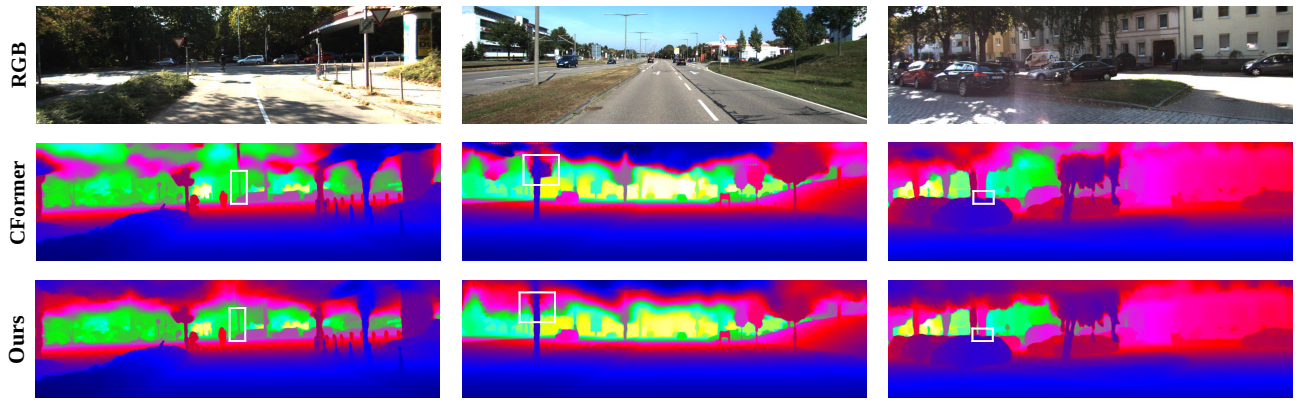

Fig. 12 Qualitative depth completion comparison on the KITTI dataset.

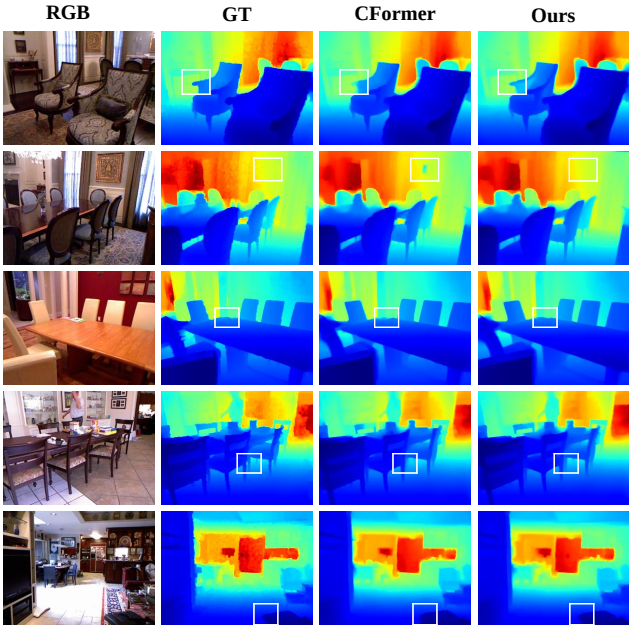

Fig. 13 Qualitative comparison of depth completion on the NYU-Depth-v2 dataset.

Fig. 14 Qualitative point cloud comparison for depth completion on the NYU-Depth-v2 dataset.

6 Failure Cases

In Fig. 16, we present failure cases of the proposed method for both monocular depth estimation and completion to better assess its robustness in real-world applications. We find that whether using depth estimation or completion, it is difficult for the proposed method to correctly predict the depth of transparent or highly reflective regions. This may be attributed to the inability of the depth sensors to capture valid depth values (column 1) or its capture of incorrect depth values (columns 2, 3 and 4) in these regions, which hinders effective supervision of the predicted depth.

7 Conclusion

In this paper, we introduce an innovative concept, iterative elastic bins, for the classification-regression-based monocular depth estimation and completion. The introduced IEBins leverages multiple small numbers of bins in an iterative elastic manner to search for high-quality depth information and can be plugged into other frameworks as a strong baseline. In addition, we developed dedicated frameworks to instantiate the IEBins, composed of a feature extractor and an iterative optimizer. Extensive experiments on the KITTI, NYU-Depth-v2, SUN RGB-D, ScanNet and DIODE datasets indicate that the proposed method outperforms prior state-of-the-art competitors. In our future work, we will try to provide theoretical analysis related to IEBins by leveraging stability analysis results on dynamic systems.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China under grant U1909215 and 51975029, in part by the A*STAR Singapore through Robotics Horizontal Technology Coordinating Office (HTCO) under Project C221518005.

Data Availability. All the datasets used in this paper are available online. KITTI (https://www.cvlibs.net/datasets/kitti/), NYU-Depth-v2 (https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html), SUN RGB-D (https://rgbd.cs.princeton.edu/), ScanNet (http://www.scan-net.org/) and DIODE (https://diode-dataset.org/).

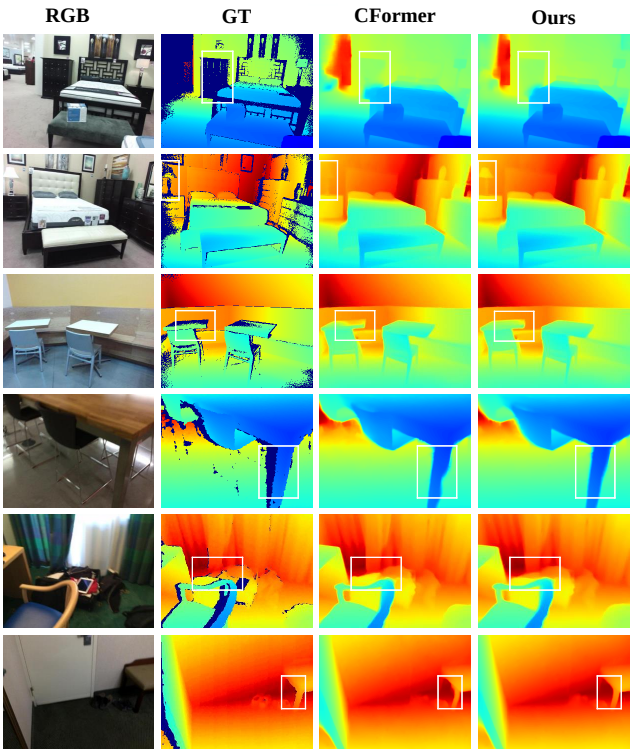

Fig. 15 Qualitative comparison of depth completion on the SUN RGB-D and ScanNet datasets. The samples in the top three rows are from the SUN RGB-D dataset, while the samples in the bottom three rows are from the ScanNet dataset.

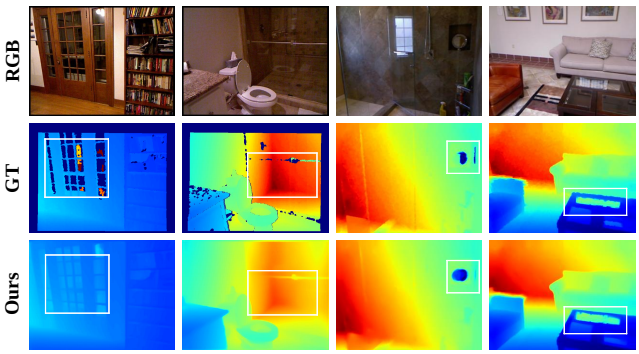

Fig. 16 Failure cases. From left to right, the depth predictions in the first two columns are from monocular depth estimation, while the depth predictions in the last two columns are from depth completion.

Statements and Declarations. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Agarwal, A., & Arora, C. (2023). Attention attention everywhere: Monocular depth prediction with skip attention. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*. (pp. 5861–5870).

Aich, S., Vianney, J. M. U., Islam, M. A., & Liu, M. K. B. (2021). Bidirectional attention network for monocular depth estimation. In *2021 IEEE International Conference on Robotics and Automation*. IEEE, (pp. 11746–11752).

Bhat, S. F., Alhashim, I., & Wonka, P. (2021). Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 4009–4018).

Bhat, S. F., Alhashim, I., & Wonka, P. (2022). Localbins: Improving depth estimation by learning local distributions. In *Proceedings of the European Conference on Computer Vision*. Springer, (pp. 480–496).

Cao, Y., Wu, Z., & Shen, C. (2017). Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Transactions on Circuits and Systems for Video Technology*, *28*(11), pp. 3174–3182.

Chen, D., Huang, T., Song, Z., Deng, S., & Jia, T. (2023). Agg-net: Attention guided gated-convolutional network for depth image completion. In *Proceedings of the IEEE International Conference on Computer Vision*. (pp. 8853–8862).

Chen, P.-Y., Liu, A. H., Liu, Y.-C., & Wang, Y.-C. F. (2019a). Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 2624–2632).

Chen, X., Chen, X., & Zha, Z.-J. (2019b). Structure-aware residual pyramid network for monocular depth estimation. In *Proceedings of the International Joint Conference on Artificial Intelligence*. (pp. 694–700).

Chen, Y., Yang, B., Liang, M., & Urtasun, R. (2019c). Learning joint 2d-3d representations for depth completion. In *Proceedings of the IEEE International Conference on Computer Vision*. (pp. 10023–10032).

Cheng, X., Wang, P., Guan, C., & Yang, R. (2020). Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34. (pp. 10615–10622).

Cheng, X., Wang, P., & Yang, R. (2018). Depth estimation via affinity learned with convolutional spatial propagation network. In *Proceedings of the European Conference on Computer Vision*. (pp. 103–119).

Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., & Niessner, M. (2017a). Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., et al. (2017b). Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*. (pp. 764–773).

Diaz, R., & Marathe, A. (2019). Soft labels for ordinal regression. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. (pp. 4738–4747).

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*.

Eigen, D., Puhrsch, C., & Fergus, R. (2014). Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems*. (pp. 2366–2374).

Fu, H., Gong, M., Wang, C., Batmanghelich, K., & Tao, D. (2018). Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 2002–2011).

Gao, W., Wan, F., Pan, X., Peng, Z., Tian, Q., Han, Z., et al. (2021). Ts-cam: Token semantic coupled attention map for weakly supervised object localization. In *Proceedings of the IEEE International Conference on Computer Vision*. (pp. 2886–2895).

Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, *32*(11), pp. 1231–1237.

Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, (pp. 315–323).

Gu, X., Yuan, W., Dai, Z., Zhu, S., Tang, C., Dong, Z., et al. (2023). Dro: Deep recurrent optimizer for video to depth. *IEEE Robotics and Automation Letters*, *8*(5), pp. 2844–2851.

Guizilini, V., Ambrus, R., Burgard, W., & Gaidon, A. (2021). Sparse auxiliary networks for unified monocular depth prediction and completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 11078–11088).

He, K., Sun, J., & Tang, X. (2012). Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(6), pp. 1397–1409.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Hu, D., Peng, L., Chu, T., Zhang, X., Mao, Y., Bondell, H., et al. (2022). Uncertainty quantification in depth estimation via constrained ordinal regression. In *Proceedings of the European Conference on Computer Vision*. Springer, (pp. 237–256).

Hu, M., Wang, S., Li, B., Ning, S., Fan, L., & Gong, X. (2021). Penet: Towards precise and efficient image guided depth completion. In *2021 IEEE International Conference on Robotics and Automation*. IEEE, (pp. 13656–13662).

Imran, S., Liu, X., & Morris, D. (2021). Depth completion with twin surface extrapolation at occlusion boundaries. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 2583–2592).

Imran, S., Long, Y., Liu, X., & Morris, D. (2019). Depth coefficients for depth completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, (pp. 12438–12447).

Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., et al. (2011). Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. (pp. 559–568).

Jia, W., Zhao, W., Song, Z., & Li, Z. (2023). Object servoing of differential-drive service robots using switched control. *Journal of Control and Decision*, *10*(3), pp. 314–325.

Jiang, Y., Chang, S., & Wang, Z. (2021). Transgan: Two transformers can make one strong gan. *Advances in Neural Information Processing Systems*.

Johnston, A., & Carneiro, G. (2020). Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume. In *Proceedings of the IEEE International Conference on Computer Vision*. (pp. 4756–4765).

Ke, B., Obukhov, A., Huang, S., Metzger, N., Daudt, R. C., & Schindler, K. (2024). Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 9492–9502).

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., & Navab, N. (2016). Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth International Conference on 3D Vision*. IEEE, (pp. 239–248).

Lee, B.-U., Lee, K., & Kweon, I. S. (2021a). Depth completion using plane-residual representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 13916–13925).

Lee, J. H., Han, M.-K., Ko, D. W., & Suh, I. H. (2019). From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*.

Lee, S., Lee, J., Kim, B., Yi, E., & Kim, J. (2021b). Patch-wise attention network for monocular depth estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35. (pp. 1873–1881).

Li, Z., Wang, X., Liu, X., & Jiang, J. (2022). Binsformer: Revisiting adaptive bins for monocular depth estimation. *arXiv preprint arXiv:2204.00987*.

Lin, Y., Cheng, T., Zhong, Q., Zhou, W., & Yang, H. (2022). Dynamic spatial propagation network for depth completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36. (pp. 1638–1646).

Lipson, L., Teed, Z., & Deng, J. (2021). Raft-stereo: Multilevel recurrent field transforms for stereo matching. In *2021 International Conference on 3D Vision (3DV)*. IEEE, (pp. 218–227).

Liu, L., Liao, Y., Wang, Y., Geiger, A., & Liu, Y. (2021a). Learning steering kernels for guided depth completion. *IEEE Transactions on Image Processing*, *30*, pp. 2850–2861.

Liu, L., Song, X., Lyu, X., Diao, J., Wang, M., Liu, Y., et al. (2021b). Fcfr-net: Feature fusion based coarse-to-fine residual learning for depth completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35. (pp. 2136–2144).

Liu, S., De Mello, S., Gu, J., Zhong, G., Yang, M.-H., & Kautz, J. (2017). Learning affinity via spatial propagation networks. *Advances in Neural Information Processing Systems*, *30*.

Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., et al. (2022a). Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 12009–12019).

Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., et al. (2022b). Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. (pp. 12009–12019).

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., et al. (2021c). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE International Conference on Computer Vision*. (pp. 10012–10022).

Long, X., Lin, C., Liu, L., Li, W., Theobalt, C., Yang, R., et al. (2021). Adaptive surface normal constraint for depth estimation. In *Proceedings of the IEEE International Conference on Computer Vision*. (pp. 12849–12858).

Loshchilov, I., & Hutter, F. (2018). Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Ma, F., Cavalheiro, G. V., & Karaman, S. (2019). Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, (pp. 3288–3295).

Ma, F., & Karaman, S. (2018). Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *IEEE International Conference on Robotics and Automation*. IEEE, (pp. 4796–4803).

Mur-Artal, R., & Tardós, J. D. (2017). Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, *33*(5), pp. 1255–1262.

Ning, J., Li, C., Zhang, Z., Wang, C., Geng, Z., Dai, Q., et al. (2023). All in tokens: Unifying output space of visual tasks via soft token. In *Proceedings of the IEEE International Conference on Computer Vision*. (pp. 19900–19910).

Park, J., Joo, K., Hu, Z., Liu, C.-K., & So Kweon, I. (2020). Non-local spatial propagation network for depth completion. In *Proceedings of the European Conference on Computer Vision*. Springer, (pp. 120–136).

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., et al. (2017). Automatic differentiation in pytorch. In *Advances in Neural Information Processing Systems Workshop Autodiff*.

Patil, V., Sakaridis, C., Liniger, A., & Van Gool, L. (2022). P3depth: Monocular depth estimation with a piecewise planarity prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 1610–1621).

Peng, R., Wang, R., Lai, Y., Tang, L., & Cai, Y. (2021). Excavating the potential capacity of self-supervised monocular depth estimation. In *Proceedings of the IEEE International Conference on Computer Vision*. (pp. 15560–15569).

Qiao, S., Zhu, Y., Adam, H., Yuille, A., & Chen, L.-C. (2021). Vip-deeplab: Learning visual perception with depth-aware video panoptic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 3997–4008).

Qiu, J., Cui, Z., Zhang, Y., Zhang, X., Liu, S., Zeng, B., et al. (2019). Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 3313–3322).

Ranftl, R., Bochkovskiy, A., & Koltun, V. (2021). Vision transformers for dense prediction. In *Proceedings of the IEEE International Conference on Computer Vision*. (pp. 12179–12188).

Rho, K., Ha, J., & Kim, Y. (2022). Guideformer: Transformers for image guided depth completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 6250–6259).

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 10684–10695).

Saxena, A., Chung, S. H., Ng, A. Y., et al. (2005). Learning depth from single monocular images. In *Advances in Neural Information Processing Systems*, volume 18. (pp. 1–8).

Shao, S., Li, R., Pei, Z., Liu, Z., Chen, W., Zhu, W., et al. (2022). Towards comprehensive monocular depth estimation: Multiple heads are better than one. *IEEE Transactions on Multimedia*.

Shao, S., Pei, Z., Chen, W., Wu, X., & Li, Z. (2023a). Nddepth: Normal-distance assisted monocular depth estimation. In *Proceedings of the IEEE International Conference on Computer Vision*. (pp. 7931–7940).

Shao, S., Pei, Z., Wu, X., Liu, Z., Chen, W., & Li, Z. (2023b). Iebins: Iterative elastic bins for monocular depth estimation. In *Advances in Neural Information Processing Systems*.

Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., et al. (2016). Real-time single image and video super-resolution

using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 1874–1883).

Silberman, N., Hoiem, D., Kohli, P., & Fergus, R. (2012). Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*. Springer, (pp. 746–760).

Song, S., Lichtenberg, S. P., & Xiao, J. (2015). Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 567–576).

Tan, M., & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*. PMLR, (pp. 6105–6114).

Tang, J., Tian, F.-P., Feng, W., Li, J., & Tan, P. (2020). Learning guided convolutional network for depth completion. *IEEE Transactions on Image Processing*, *30*, pp. 1116–1129.

Teed, Z., & Deng, J. (2020). Raft: Recurrent all-pairs field transforms for optical flow. In *Proceedings of the European Conference on Computer Vision*. Springer, (pp. 402–419).

Teed, Z., & Deng, J. (2021). Raft-3d: Scene flow using rigid-motion embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 8375–8384).

Van Gansbeke, W., Neven, D., De Brabandere, B., & Van Gool, L. (2019). Sparse and noisy lidar completion with rgb guidance and uncertainty. In *International Conference on Machine Vision Applications*. IEEE, (pp. 1–6).

Vasiljevic, I., Kolkin, N., Zhang, S., Luo, R., Wang, H., Dai, F. Z., et al. (2019). Diode: A dense indoor and outdoor depth dataset. *arXiv preprint arXiv:1908.00463*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*. (pp. 5998–6008).

Wang, F., Galliani, S., Vogel, C., & Pollefeys, M. (2022). Itermvs: iterative probability estimation for efficient multi-view stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 8606–8615).

Wang, Y., Shi, M., Li, J., Huang, Z., Cao, Z., Zhang, J., et al. (2023). Neural video depth stabilizer. In *Proceedings of the IEEE International Conference on Computer Vision*. (pp. 9466–9476).

Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. (2018). Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision*.

Xu, Y., Zhu, X., Shi, J., Zhang, G., Bao, H., & Li, H. (2019). Depth completion from sparse lidar data with depth-normal constraints. In *Proceedings of the IEEE International Conference on Computer Vision*. (pp. 2811–2820).

Yan, Z., Li, X., Wang, K., Chen, S., Li, J., & Yang, J. (2023). Distortion and uncertainty aware loss for panoramic depth completion. In *Proceedings of the International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*. PMLR, (pp. 39099–39109).

Yan, Z., Wang, K., Li, X., Zhang, Z., Li, J., & Yang, J. (2022). Rignet: Repetitive image guided network for depth completion. In *Proceedings of the European Conference on Computer Vision*. Springer, (pp. 214–230).

Yang, G., Tang, H., Ding, M., Sebe, N., & Ricci, E. (2021). Transformer-based attention networks for continuous pixel-wise prediction. In *Proceedings of the IEEE International Conference on Computer Vision*. (pp. 16269–16279).

Yang, X., Ma, Z., Ji, Z., & Ren, Z. (2023). Gedepth: Ground embedding for monocular depth estimation. In *Proceedings of the IEEE International Conference on Computer Vision*. (pp. 12719–12727).

Yasarla, R., Cai, H., Jeong, J., Shi, Y., Garrepalli, R., & Porikli, F. (2023). Mamo: Leveraging memory and attention for monocular video depth estimation. In *Proceedings of the IEEE International Conference on Computer Vision*. (pp. 8754–8764).

Yin, W., Liu, Y., Shen, C., & Yan, Y. (2019). Enforcing geometric constraints of virtual normal for depth prediction. In *Proceedings of the IEEE International Conference on Computer Vision*. (pp. 5684–5693).

Yu, Z., Sheng, Z., Zhou, Z., Luo, L., Cao, S.-Y., Gu, H., et al. (2023). Aggregating feature point cloud for depth completion. In *Proceedings of the IEEE International Conference on Computer Vision*. (pp. 8732–8743).

Yuan, L., Hou, Q., Jiang, Z., Feng, J., & Yan, S. (2021). Volo: Vision outlooker for visual recognition. *arXiv preprint arXiv:2106.13112*.

Yuan, W., Gu, X., Dai, Z., Zhu, S., & Tan, P. (2022). Neural window fully-connected crfs for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 3916–3925).

Zhang, Y., & Funkhouser, T. (2018). Deep depth completion of a single rgb-d image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 175–185).

Zhang, Y., Guo, X., Poggi, M., Zhu, Z., Huang, G., & Mattoccia, S. (2023). Completionformer: Depth completion with convolutions and vision transformers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 18527–18536).

Zhang, Y., Wei, P., Li, H., & Zheng, N. (2020). Multiscale adaptation fusion networks for depth completion. In *International Joint Conference on Neural Networks*. IEEE, (pp. 1–7).

Zhao, S., Gong, M., Fu, H., & Tao, D. (2021). Adaptive context-aware multi-modal network for depth completion. *IEEE Transactions on Image Processing*, *30*, pp. 5264–5276.

Zhao, W., Rao, Y., Liu, Z., Liu, B., Zhou, J., & Lu, J. (2023). Unleashing text-to-image diffusion models for visual perception. In *Proceedings of the IEEE International Conference on Computer Vision*. (pp. 5729–5739).

Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., et al. (2021). Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 6881–6890).