CbwLoss: Constrained Bidirectional Weighted Loss for Self-Supervised Learning of Depth and Pose

Fei Wang, *Student Member, IEEE*, Jun Cheng, *Member, IEEE*, and Penglei Liu, *Student Member, IEEE*

***Abstract*— Photometric differences are widely used as supervision signals to train neural networks for estimating depth and camera pose from unlabeled monocular videos. However, this approach is detrimental for model optimization because occlusions and moving objects in a scene violate the underlying static scenario assumption. In addition, pixels in textureless regions or less discriminative pixels hinder model training. To address these problems, in this paper, we deal with moving objects and occlusions by utilizing the differences between the flow fields, and the differences between the depth structure generated by affine transformation and view synthesis, respectively. Secondly, we mitigate the effect of textureless regions on model optimization by measuring the differences between features with more semantic and contextual information without requiring additional networks. In addition, although the bidirectionality component is used in each sub-objective function, a pair of images is reasoned about only once, which helps reduce overhead. Extensive experiments and visual analysis demonstrate the effectiveness of the proposed method, which outperforms existing state-of-the-art self-supervised methods under the same conditions and without introducing additional auxiliary information.**

***Index Terms*— Depth estimation, pose estimation, constrained bidirectional weighted loss, self-supervised learning.**

I. Introduction

THE estimation of depth and camera pose from monocular videos is a fundamental and valuable but challenging task with applications in mobile robot vision and navigation [1], driverless cars [2], and other scenarios. In these scenarios, an odometer based on a wheel encoder, which is susceptible to cumulative error from imprecision of the angular measurements, wheel slippage and conversion of rotation into distance, can be replaced by the direct estimation of the camera pose from a sequence of consecutive images [1]. Furthermore, the depth information inferred from detailed object size and location information, which can be directly obtained from monocular videos without expensive depth sensors, can be used for precise location, object detection, and obstacle avoidance for autonomous vehicles [2], [3]. Compared with traditional methods [4], [5], [6], [7], [8], [9], although the existing methods benefitting from the powerful data fitting ability of deep neural networks can achieve competitive performance in depth prediction from video sequences, these methods require expensive depth sensors and considerable labor to obtain sufficiently large amounts of data labeled with pixel-level depth information or even require stereo video sequences [7] for network training. To solve these problems, researchers have recently attempted to jointly predict depth and camera pose in a self-supervised fashion by employing geometric priors directly learned from large amounts of easily accessible unlabeled videos captured using the least expensive, least restrictive, and most ubiquitous cameras [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21].

The main principle of these self-supervised methods is that one can transform one frame into another frame based on the relative camera pose, expressed in terms of rotation angles and translations, as well as the camera intrinsic matrix and a depth map estimated using spatial transformer networks [22]; then the corresponding photometric differences can be utilized as the supervision signal for model optimization. However, the training of deep neural networks based on photometric differences requires some underlying assumptions, namely, that the scene is static (without moving objects) and that there are no occlusions between adjacent frames. The performance of geometric image reconstruction is limited if these underlying assumptions are violated. Regarding these challenges, recent works address them by leveraging prior knowledge (e.g. the velocity [23], geometry structure information [16], [24] of the objects) or multi-task learning [5], [6], [10], [15], which either fails to handle objects with different velocities or requires additional overhead. In addition, the use of the photometric difference between a pixel warped from the reference frame and the corresponding pixel captured in the target frame is often problematic because pixels in textureless regions do not provide a good foundation for a neural network to find the global minimum. This problem could be mitigated, while either off-the-shelf stereo algorithms [25] or additional networks [26], [27] are required. Furthermore, to take full advantage of the information contained in both

Manuscript received 9 May 2021; revised 14 April 2022 and 5 August 2022; accepted 20 February 2023. Date of publication 14 March 2023; date of current version 31 May 2023. This work was supported in part by the National Natural Science Foundation of China under Grant U21A20487, in part by the Shenzhen Technology Project under Grant JCYJ20180507182610734 and Grant JCYJ20220818101206014, in part by the Chinese Academy of Sciences (CAS) Key Technology Talent Program, and in part by the Shenzhen Engineering Laboratory for 3D Content Generating Technologies under Grant [2017]476. The Associate Editor for this article was S. S. Nedevschi. *(Corresponding author: Jun Cheng.)*

Fei Wang and Penglei Liu are with the Guangdong-Hong Kong-Macao Joint Laboratory of Human-Machine Intelligence-Synergy Systems, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China, also with the Shenzhen College of Advanced Technology, University of Chinese Academy of Sciences, Beijing 100049, China, and also with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: fei.wang2@siat.ac.cn; pl.liu@siat.ac.cn).

Jun Cheng is with the Guangdong-Hong Kong-Macao Joint Laboratory of Human-Machine Intelligence-Synergy Systems, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China (e-mail: jun.cheng@siat.ac.cn).

Digital Object Identifier 10.1109/TITS.2023.3250744

images, stacked frames in normal order and stacked frames in reversed order [16], [24], [28], [29] are individually fed to the network for predicting both forward and backward motion, whereas it is time-consuming and computationally expensive because stacked frames in normal and reversed order are all reasoned about by networks.

Considering these basic problems, in this paper, a bidirectional weighted photometric loss is first proposed to effectively handle moving objects and occlusions effectively while taking full advantage of the information contained in both the target and reference images to improve the robustness of the algorithm. Furthermore, benefitting from this bidirectional calculation, the estimated results can be well verified online. Specifically, the photometric loss is reweighted using both adaptive weights, which are obtained by measuring the difference between the estimated depth and the depth obtained through projection transformation (which should theoretically be consistent), and camera flow occlusion masks, which are based on our observation that corresponding pixels between adjacent frames should be similar if there is no occlusion but dissimilar in the presence of occlusion (as, in the latter case, the corresponding occluded pixels are not visible). In prior work [28], [29], the error between the motion fields, obtained by applying the networks on the frames in normal and reversed order, is minimized to deal with dynamic scenes. In this paper, (1) the bidirectional image reconstruction error (that is, differences between the target image and the synthesized target image, and the differences between the reference image and the reconstructed reference image) is employed as an optimization objective. During view synthesis, the synthesized target image is obtained by warping a reference frame I_{ref} based on the depth map D_{tgt} predicted by DepthNet, the camera intrinsic matrix K, and the relative pose $T_{tgt\to ref}$ predicted by CameraNet, the reconstructed reference image is obtained by warping a target image I_{tgt} based on the reference depth map D_{ref} estimated by DepthNet, K and the relative pose $T_{ref\to tgt}$ obtained by calculating the inverse of $T_{tgt\to ref}$ rather than the pose predicted by CameraNet; (2) we weighted the bidirectional image reconstruction error utilizing the masks calculated from both camera flow consistency check and depth structure consistency check to effectively handle moving objects and occlusions; (3) although our method is bidirectional, only one-way prediction is required. Compared with existing work [16], [24], where $T_{ref\to tgt}$ is obtained by applying CameraNet on the frames in reversed order again, it is economic because a pair of images is reasoned about by CameraNet only once.

Second, the photometric information in textureless regions (e.g. uniformly colored regions) can be ambiguous, in such cases, the features error, which depends on the extracted deep features from raw images by utilizing an encoder network, is more robust than per-pixel loss, which only relies on the low-level pixel information [30]. Furthermore, dense feature loss can be employed as an auxiliary signal for image reconstruction loss based on color intensity [31], as it can incorporate more semantic and contextual information by encoding larger-scale patterns. Accordingly, a bidirectional feature perception loss is proposed to prevent the image gradient from tending toward zero in textureless regions during network training, which can enhance the perception ability of the model for weakly textured areas. Compared with work [31], our method eliminates the need to measure stereo feature maps and use additional pre-trained models for feature extraction.

Third, a bidirectional depth structure consistency loss is proposed to not only constrain the difference between the depth obtained from the multiview geometric transformation $p_{ref} \sim K\hat{T}_{tgt\to ref}\hat{D}(p_{tgt})K^{-}p_{tgt}$ and the depth predicted using DepthNet from the corresponding reference frame but also minimize the difference between the depth obtained from the transformation $p_{tgt} \sim K\hat{T}_{ref\to tgt}\hat{D}(p_{ref})K^{-}p_{ref}$ and the depth estimated from the corresponding target frame. More importantly, the scale of depth can be kept consistent based on the above constraint and explicit constraints which force forward and backward pose to have a consistent scale by calculating the inverse.

Finally, the proposed constrained bidirectional weighted loss (CbwLoss), which is obtained by constraining the bidirectional weighted photometric loss using the bidirectional feature perception loss and the bidirectional depth structure consistency loss, is employed to guide the learning of the model's parameters.

In summary, the main contributions of this paper are summarized as follows:

- We proposed a scheme to deal with the moving objects in dynamic scenes. In this scheme, we locate the moving objects using both the camera flow consistency check and the depth structure consistency check. We then calculate the masks based on the consistency check and use these masks to weight photometric loss for reducing the contribution of the corresponding region, thereby satisfying the basic assumption of image reconstruction based on the static scene.
- We proposed a simple and economic scheme to improve the robustness of the algorithm in weak texture regions. In this scheme, neither the stereo features extracted by the pre-trained model nor the additional trained auto-encoder networks are required. We directly utilize mid-level features obtained from depth estimation networks to define feature perception loss, aiming at enhancing the perception ability of the depth estimation networks in textureless regions without increasing overhead.
- We propose a simple bidirectional scheme that not only explicitly constrains the bidirectional pose scale, but also maximizes the use of the limited data. Furthermore, although it is bidirectional, only one-way prediction is required.

This paper is organized as follows. In section II, we introduce the previous work on depth and camera pose estimation based on deep learning. The proposed algorithm is presented in section III. Experimental results are reported in section IV. Finally, we conclude the paper in section V.

II. Related Work

Recently, with the rapid development of deep learning and high-performance computing devices, artificial intelligence

technology that uses deep neural networks to analyze scene depth and camera motion to accurately perceive the surrounding environment is increasingly playing an irreplaceable role in robot navigation [1] and autonomous driving [2], [3]. Therefore, methods of taking advantage of the remarkable learning ability of deep neural networks to estimate depth and camera pose from a large number of videos captured by the least expensive, most ubiquitous cameras have attracted considerable attention [4], [5], [6], [7], [8], [9], [10], [15], [16], [17], [18], [19], [21], [25], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42]. These methods can be divided into supervised and self-supervised depth-pose estimation methods depending on whether the ground truth is required.

A. Supervised Depth-Pose Estimation

For depth estimation, Eigen et al. [32] first predicted a depth map from a single image by employing two stacked deep neural networks — a coarse-scale network and a fine-scale network. The coarse-scale network was used to estimate the depth of the scene at the global level, and then, the estimated depth was refined within local regions using the fine-scale network to allow fine-scale details to be incorporated into the global prediction. Soon afterwards, depth estimation was formulated as a structured learning problems by jointly combining convolutional neural networks and conditional random fields, but no geometric priors were considered [33]. In another work, the strong ordinal correlation of depth values was taken into account to constrain the objective function, and the depth estimation problem was cast as an ordinal regression problem [9]. Thereafter, to further improve the quality of dense depth maps, Eom et al. [4] predicted depth maps by means of a two-stream convolutional neural network with convolutional gated recurrent units, which could leverage both temporal information and spatial information in video sequences, while Park et al. [7] predicted high-precision depth map by leveraging the complementary properties of light detection and ranging (LiDAR) point clouds and stereo images. More recently, various other approaches have been adopted that can greatly benefit the depth estimation task, such as attention mechanisms [8], which can be applied to emphasize the interdependency between low-resolution features capturing long-range context and fine-grained features describing local context; the exploitation of the geometric relationships between depth and surface normals [6]; and the combination of an epipolar geometry constraint with auxiliary optical flow [5].

For pose estimation, Konda et al. [34] designed the first convolutional network for predicting camera pose from visual information extracted from consecutive frames. Wang et al. [35] formulated camera pose estimation as a sequence learning problem in which poses are directly inferred from videos based on recurrent convolutional neural networks without adopting any module in the traditional visual odometry pipeline. Based on the method of [35], the uncertainty of the camera pose has been considered and modeled using its covariance to correct the drift and bound the uncertainty [36]. Most recently, a multimodal localization fusion framework [39], in which LiDAR odometry and visual odometry are modeled simultaneously, has been proposed to obtain more robust results in a complex environment; however, the cost of the necessary calibration data and sensors is also increased under this framework.

B. Self-Supervised Depth-Pose Estimation

In contrast to supervised depth-pose estimation methods, in which depth estimation and pose estimation are treated independently as two unrelated problems, in self-supervised depth-pose estimation methods, these two problems are tightly coupled to model their correlations. Most importantly, the ground truth information, which is time consuming and laborious to acquire for real-world scenes, is not required.

As one possible approach, Garg et al. [40] first explicitly reconstructed the reference frame by explicitly generating an inverse warp of the target image using the estimated depth and camera pose and then employed the reconstruction error as the supervision signal to train networks to estimate the depth map. The standard photometric warp loss was later improved by taking contextual information into consideration, instead of relying solely on per-pixel color matching [31]. However, in the above methods, not only the relative camera pose between the reference image and the target image must be known, but also stereo video sequences are required to reconstruct the reference view from the live view. In addition, although high-quality images can be reconstructed by minimizing the reconstruction error, the estimated depth map is of poor quality. To overcome this problem, a left-right consistency check [19] has been proposed to improve the quality of synthesized depth images, but in this case, stereo images are required for network training. Different from the method in [19], Zhou et al. [20] directly estimated the depth map and camera pose from monocular videos in a fully unsupervised fashion by jointly training a depth network and a pose network for the first time. To achieve increased robustness to outliers and non-Lambertian regions, Yin et al. [18] designed a cascaded architecture consisting of two stages to model static scenes and dynamic objects independently. Similarly, Ranjan et al. [15] segmented the scene of interest into static and moving regions by employing additional segmentation networks. Specifically, the static scene in a video sequence can be analyzed based on a depth network and a pose network, while the whole scene consisting of both static and moving objects can be analyzed by employing an additional optical flow network. Then, the pixels in the scene can be assigned as belonging to either static or independently moving regions using segmentation networks. Similar to the above method [15], Luo et al. [21] decomposed the scene of interest into background and foreground using three parallel networks: one to predict the camera motion, one to predict the dense depth map, and one to predict the per-pixel optical flow. Based on the methods presented in [15] and [21], a less-than-mean mask [10] was subsequently designed to further exclude mismatched pixels disturbed by motion or illumination changes during the training of the depth and pose networks

and was also used to exclude trivial mismatched pixels in the training of the optical flow network.

In contrast to the above methods of explicitly segmenting scenes into static and dynamic object regions, Godard et al. [17] designed a per-pixel minimum reprojection loss, instead of averaging the photometric error over all source images, to handle occlusions. Bian et al. [16] adopted a geometric consistency constraint to explicitly enforce scale consistency between different samples in order to handle dynamic scenes. In addition, to improve the quality of the predicted depth maps, Watson et al. [25] enhanced the existing photometric loss using depth cues generated from off-the-shelf algorithms. Zhan et al. [41] jointly predicted depth and surface normals using an additional network. Zhao et al. [42] directly solved the fundamental matrix based on optical flow correspondence and calculated the camera pose without PoseNet; in addition, a double view triangulation module was used to recover the up-to-scale scene structure. Subsequently, feature-metric loss [27], which shares the same spirit as previous work [31], was introduced to make up for the shortcomings that the standard photometric warp loss is not robust to uniformly textured areas. However, features, which are utilized to define feature-metric loss for depth estimation, must be explicitly learned by additional networks, resulting in increased overhead. Furthermore, only unidirectional warped features were considered. In order to further improve the quality of the estimated depth maps, either semantic information [43], [44] obtained from additional semantic segmentation networks, which were jointly optimized with the depth estimation task, or the relations between depths [45], or more complex depth networks [23] are also leveraged to improved depth prediction.

In parallel, recent work [28], [29] has shown the consistency across neighboring frames could be well constrained by imposing the forward motion field, estimated by feeding a pair of frames in normal order to network, and the backward motion field, predicted by feeding the frames in reversed order to the same network again, to be the opposite of each other. Instead of directly constraining the motion field, the forward view differences between the source view and the warped view, obtained by transforming a neighboring view utilizing the forward relative pose and the backward photometric loss computed by inverting their roles again are imposed simultaneously [16], [24]. Despite these methods producing good results, a pair of frames need to be propagated twice across the network in order to get forward and backward either motion field or relative pose, resulting in increasing the training and inference overhead.

Notably, the above methods require either ground truth information for network training; additional networks (e.g., segmentation and/or optical flow networks) to filter out outliers, or off-the-shelf algorithms to enhance the photometric loss. Different from previous work [16], [24], [28], [29], in this paper, the relative pose, obtained by applying a network on frames in normal order, is reused by the inverse operation, and then is combined with camera intrinsic matrix and the corresponding predicted depth to obtain the corresponding synthetic frame, which is utilized to compute backward photometric loss. Instead of features generated from pre-trained models [31] or additional networks [27], our method utilizes mid-features, which were obtained from depth estimation networks, to define feature perception loss, aiming at enhancing the perception ability of the model in textureless regions without increasing overhead. Similarly, the pose, obtained by applying a network on frames in normal order, is also utilized to compute both projected depth and the corresponding synthetic feature in the other direction by utilizing economic inverse operations rather than costly reasoning about the frames in reversed order again. Based on these, backward feature perception error and backward depth structure consistency error could be obtained. In addition, to mitigate the adverse impact of moving objects and occlusions, the camera flow occlusion masks based on flow fields are used to weight the above photometric loss.

III. Method

In this section, the theory of view synthesis is briefly introduced, and then, the bidirectional weighted photometric loss is presented. Afterwards, the bidirectional feature perception loss that is used to deal with weakly textured or textureless regions is described. Thereafter, the bidirectional depth structure consistency constraint is shown. The general framework of our proposed method is depicted in Fig. 1.

A. View Synthesis and Bidirectional Weighted Photometric Loss

1) View Synthesis and Bidirectional Photometric Loss: Our self-supervised learning problem is treated as a novel view synthesis problem. Specifically, consider a training image sequence $I_1, I_2, \ldots, I_N$, where one of the frames is the target frame I_{tgt} and the remaining frames are reference frames I_{ref} in a monocular video, with $1 \leq ref \leq N, ref \neq tgt$. The photometric differences between the target frame I_{tgt} and synthetic target frame $\hat{I}_{tgt}^{pose}$, obtained by warping a reference frame I_{ref} based on the predicted depth map D_{tgt}, the camera intrinsic matrix K and the predicted camera pose $T_{tgt\rightarrow ref}$, which is described in terms of camera rotation angles α, β and γ and translations t_x, t_y and t_z, are utilized as the supervision signal for model optimization. For convenience of description, suppose that the image coordinates of a point of interest are expressed as $img_{xy}^{tgt} = (x, y)$ and the predicted depth is d_{tgt} at the image coordinates img_{xy}^{tgt} then, we can transform the image coordinates into the world coordinates $img_{xyz}^{tgt} = (x_w^{tgt}, y_w^{tgt}, z_w^{tgt})$ in accordance with formula (1):

$$x_w^{tgt} = \frac{d_{tgt}}{f}(x - c_x) \tag{1a}$$

$$y_w^{tgt} = \frac{d_{tgt}}{f}(y - c_y) \tag{1b}$$

$$z_w^{tgt} = d_{tgt} \tag{1c}$$

$$img_{xyz}^{tgt} = D_{tgt} K^{-} img_{xy}^{tgt} \tag{1d}$$

where (c_x, c_y, f) represents the principal point offset and focal length. Thereafter, the transformed world coordinates $\hat{img}_{xyz}^{ref} = (\hat{x}_w^{ref}, \hat{y}_w^{ref}, \hat{z}_w^{ref})$ can be calculated in accordance

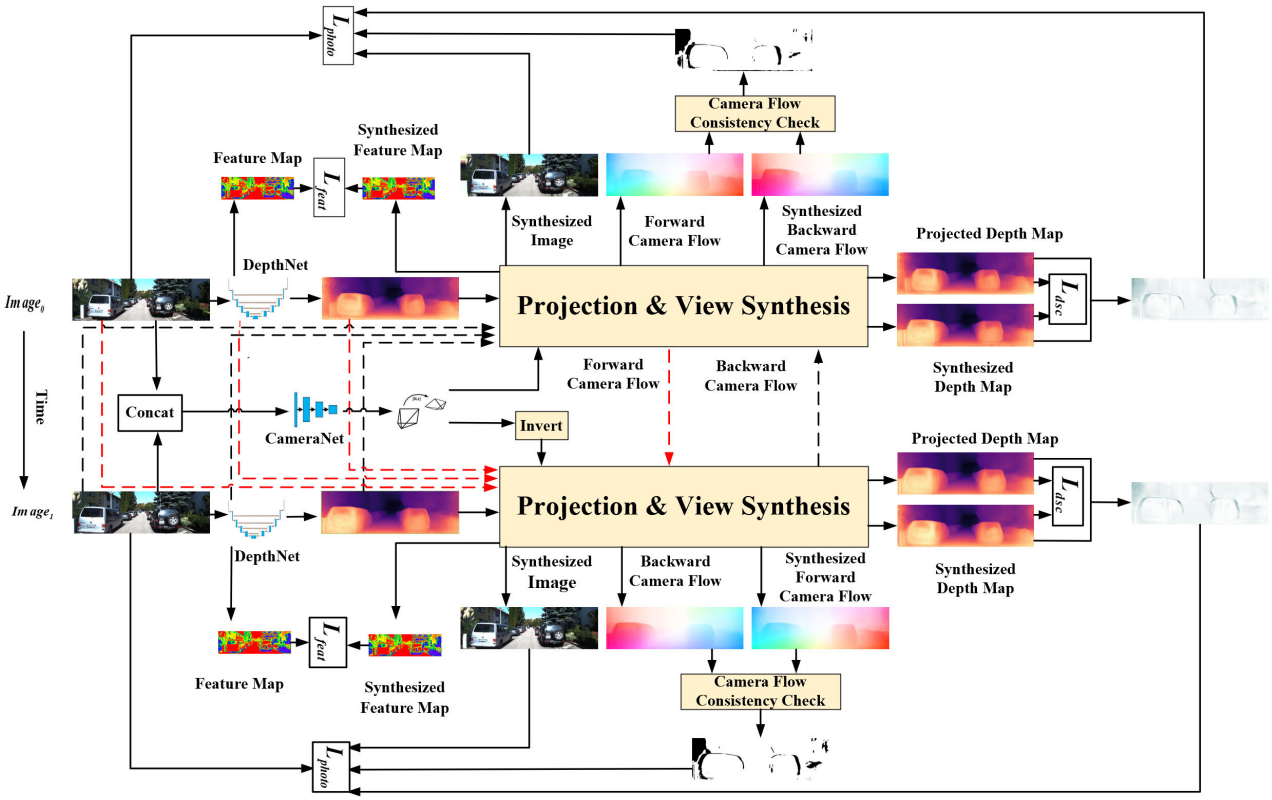

Fig. 1. Diagram of the general framework. Given two consecutive frames I_0 and I_1, the depth maps D_0 and D_1 can be estimated by DepthNet, the feature maps f_0 and f_1 of maximum resolution can be extracted from the corresponding images using the encoder network of DepthNet, and the camera pose T can be estimated by CameraNet. Then the forward/backward camera flow and the synthesized forward/backward camera flow can be obtained for a consistency check based on projection transformation and a bilinear sampling mechanism. The corresponding image, depth map, and feature map can then be synthesized for the view reconstruction loss L_{photo}, the depth structure consistency loss L_{dsc}, and the feature perception loss L_{feat}.

with formula (2):

$$\hat{img}_{xyz}^{ref} = R_x R_y R_z img_{xyz}^{tgt} + t \tag{2}$$

where $(R_x, R_y, R_z, t) \in SE3$ denotes a $3D$ rotation and translation belonging to a special Euclidean group,[1] constituting a homogeneous transformation matrix $T_{tgt\to ref}$. Thereafter, the transformed image coordinates $\hat{img}_{xy}^{ref} = (\hat{x}, \hat{y})$ and the backward camera flow u_b can be obtained according to formula (3), similarly, the forward camera flow u_f can be also acquired.

$$\hat{x} = \frac{f}{\hat{z}_w^{ref}} + c_x \tag{3a}$$

$$\hat{y} = \frac{f}{\hat{z}_w^{ref}} + c_y \tag{3b}$$

$$\hat{img}_{xy}^{ref} = K T_{tgt\to ref} img_{xyz}^{tgt} \tag{3c}$$

$$u_b = (\hat{x} - x, \hat{y} - y) \tag{3d}$$

Based on the transformed image coordinates $\hat{img}_{xy}^{ref}$, the value of the synthesized target frame $\hat{I}_{tgt}^{pose}$ can be obtained via the differentiable bilinear sampling mechanism proposed in [22]. Similarly, the transformed image coordinates $\hat{img}_{xy}^{tgt}$ can be calculated, and the synthetic reference frame $\hat{I}_{ref}^{pose}$ can be also obtained by warping the target frame I_{tgt} based on the camera intrinsic matrix K, the predicted depth map D_{ref}, and the computed camera pose $T_{ref\to tgt}$. Thus, the bidirectional photometric loss can be formulated as shown in formula (4):

$$L_{photo}^{bi} = L_{photo}^{ref\to tgt}(I_{tgt}, \hat{I}_{tgt}^{pose}) + L_{photo}^{tgt\to ref}(\hat{I}_{ref}^{pose}, I_{ref}) \tag{4}$$

where $L_{photo}^{ref\to tgt}(\cdot)$ and $L_{photo}^{tgt\to ref}(\cdot)$ are the corresponding photometric error functions that measure the differences between the target frame I_{tgt} and the corresponding synthesized frame $\hat{I}_{tgt}^{pose}$ and between the reference frame I_{ref} and the corresponding synthesized frame $\hat{I}_{ref}^{pose}$, respectively.

As is common practice [15], [19], [26], [46], we also adopt the robust image similarity measure (the structural similarity index measure, SSIM) [47] shown in formula (5) for the photometric losses shown in the formulas (6) and (7), while the robust error function shown in formula (8) is adopted instead of the $L1$ norm.

$$SSIM(a, b) = \frac{(2\mu_a\mu_b + c_1)(2\delta_{ab} + c_2)}{(\mu_a^2 + \mu_b^2 + c_1)(\delta_a^2 + \delta_b^2 + c_2)} \tag{5}$$

$$L_{photo}^{ref\to tgt}(I_{tgt}, \hat{I}_{tgt}^{pose}) = \alpha \frac{1 - SSIM(I_{tgt}, \hat{I}_{tgt}^{pose})}{2} + (1-\alpha) ERF(I_{tgt}, \hat{I}_{tgt}^{pose}) \tag{6}$$

[1] A special Euclidean group is a set of Euclidean transformations denoted by an algebraic structure composed of sets and operations.

$$L_{photo}^{tgt\to ref}(\hat{I}_{ref}^{pose}, I_{ref} = \alpha\frac{1-SSIM(I_{ref}, \hat{I}_{ref}^{pose})}{2} + (1-\alpha)ERF(I_{ref}, \hat{I}_{ref}^{pose}) \quad (7)$$

$$ERF(m,n) = \sqrt{(m-n)^2+\epsilon^2} \quad (8)$$

Here μ and δ are the local mean and variance, respectively, over the pixel neighborhood with $c_1 = 0.01^2$ and $c_2 = 0.03^2$; α is taken to be 0.85; and $ERF(\cdot)$ is our robust error measure function, where $\epsilon = 0.01$ in this paper.

2) Bidirectional Camera Flow Occlusion Masks: The prerequisites that there are no occlusions or moving objects in the scene of interest need to be satisfied when the photometric error is employed as the supervision signal to optimize a model for estimating depth and camera pose from large amounts of video data. If any of these assumptions are violated during neural network training, the gradients could be disrupted, impeding the training process. To mitigate the adverse impact from moving objects and occlusions, we propose bidirectional camera flow occlusion masks based on the observation that in general, the pixels in one frame should be similar to the pixels in another consecutive frame; however, in the case of occlusion, the pixels should not be similar because the corresponding pixels in the occluded frame are not visible. Similar to the optical flow estimation tasks [48], pixels will be marked as occlusions whenever the mismatch between different flow fields occurs. Nevertheless, different from the method [48] where flow fields between adjacent frames are directly estimated by FlowNetC, our flow fields are generated from the corresponding transformed image coordinates, which are obtained during affine transformation utilizing the estimated relative pose. Specifically, the corresponding backward camera flow u_b can be computed according to formula (3d); then, the synthetic forward camera flow $\hat{u}_f$ can be obtained by both the transformed image coordinates $\hat{img}_{xy}^{ref}$ and the forward camera flow u_f via the differentiable bilinear sampling mechanism [22]. Thereafter, the backward camera flow occlusion mask can be defined as shown in formula (9). Using a similar scheme, the synthetic backward camera flow $\hat{u}_b$ can be acquired based on $\hat{img}_{xy}^{tgt}$ and the backward camera flow u_b, and then the forward camera flow occlusion mask can be defined as shown in formula (10).

$$M_{occ}^{ref\to tgt} = \Gamma(\|u_b + \hat{u}_f\|^2, \alpha_1(\|u_b\|^2 + \|\hat{u}_f\|^2) + \alpha_2) \quad (9)$$

$$M_{occ}^{tgt\to ref} = \Gamma(\|u_f + \hat{u}_b\|^2, \alpha_1(\|u_f\|^2 + \|\hat{u}_b\|^2) + \alpha_2) \quad (10)$$

where $\Gamma(a,b)$ represents the indicator function defined in formula (11). We set $\alpha_1 = 0.01$ and $\alpha_2 = 0.5$ in all our experiments.

$$\Gamma(a,b) = \begin{cases} 1, & a < b \\ 0, & otherwise \end{cases} \quad (11)$$

Based on the bidirectional camera occlusion masks shown in formulas (9) and (10), adaptive weights (described in detail in formulas (20) and (21) in subsection III-C), and the bidirectional photometric loss shown in formula (4), the bidirectional weighted photometric loss can be defined as shown in formula (12).

$$L_{photo}^{biw} = \lambda_p^{tgt} * M_{valid}^{ref\to tgt} * W_{aw}^{ref\to tgt} * L_{photo}^{ref\to tgt} + \lambda_p^{ref} * M_{valid}^{tgt\to ref} * W_{aw}^{tgt\to ref} * L_{photo}^{tgt\to ref} \quad (12)$$

$$M_{valid}^{ref\to tgt} = 1 - \lambda_{occ}^{tgt} * M_{occ}^{ref\to tgt} \quad (13a)$$

$$M_{valid}^{tgt\to ref} = 1 - \lambda_{occ}^{ref} * M_{occ}^{tgt\to ref} \quad (13b)$$

B. Bidirectional Feature Perception Loss

The photometric error between the target frame and the corresponding frame synthesized from the reference frame is employed as the supervision signal to update the gradients and weights of the model. Therefore, the gradients play a crucial role during neural network training. Here, the loss function is reanalyzed from the gradient update perspective. To simplify the description, we analyze only the photometric loss $L_{photo}^{ref\to tgt} = L_{photo}^{ref\to tgt}(I_{tgt}(p), \hat{I}_{tgt}^{pose}(\hat{p}|T,D))$. Based on the chain rule, we can express the gradients of $L_{photo}^{ref\to tgt}$ with respect to the depth D and the camera pose T as shown in formula (14).

$$\frac{\partial L_{photo}^{ref\to tgt}}{\partial D} = \frac{\partial L_{photo}^{ref\to tgt}}{\partial \hat{I}_{tgt}^{pose}(\hat{p}|T,D)} * \frac{\partial \hat{I}_{tgt}^{pose}(\hat{p}|T,D)}{\partial \hat{p}} * \frac{\partial \hat{p}}{\partial D} \quad (14a)$$

$$\frac{\partial L_{photo}^{ref\to tgt}}{\partial T} = \frac{\partial L_{photo}^{ref\to tgt}}{\partial \hat{I}_{tgt}^{pose}(\hat{p}|T,D))} * \frac{\partial \hat{I}_{tgt}^{pose}(\hat{p}|T,D))}{\partial \hat{p}} * \frac{\partial \hat{p}}{\partial T} \quad (14b)$$

As seen from formula (14), the gradient $\frac{\partial L_{photo}^{ref\to tgt}}{\partial D}$ and the gradient $\frac{\partial L_{photo}^{ref\to tgt}}{\partial T}$ both depend on the image gradient $\frac{\partial \hat{I}_{tgt}^{pose}(\hat{p}|T,D)}{\partial \hat{p}}$. For textureless regions, the image gradients are close to zero and thus make no contribution to the gradients $\frac{\partial L_{photo}^{ref\to tgt}}{\partial D}$ and $\frac{\partial L_{photo}^{ref\to tgt}}{\partial T}$, thereby hindering network training. However, the deep features extracted from images by the encoder network can encode larger-scale patterns in the images, with redundancies and noise removed. Therefore, these features are more discriminative than the raw RGB image features for textureless regions. Features error is more robust than the per-pixel loss [30] in textureless regions. Furthermore, dense feature loss can be used as an auxiliary signal for image reconstruction loss based on color intensity [27], [31]. Instead of features generated from pre-trained models [30], [31] or by training additional auto-encoder networks [27], we directly utilize mid-level features obtained from depth estimation networks to define feature perception loss aiming at enhancing the perception ability of the depth estimation networks in textureless regions without increasing overhead. Inspired by the concept of view synthesis, we can force networks to pay more attention to textureless regions by simultaneously minimizing the difference between these features. More precisely, given a target image I_{tgt} and a reference image I_{ref}, the corresponding target features f_{tgt} and reference features f_{ref} can be extracted by the encoder network, and then, the

target features $\hat{f}_{tgt}$ can be synthesized from the reference features f_{ref} based on the transformed image coordinates $\hat{img}_{xy}^{ref}$ using the differentiable bilinear sampling mechanism. Similarly, the reference features $\hat{f}_{ref}$ can also be synthesized based on the target features f_{tgt} and the image coordinates $\hat{img}_{xy}^{tgt}$. Accordingly, the photometric loss function can be constrained by measuring the differences between the target features f_{tgt} obtained by encoding the target frame I_{tgt} and the synthesized target features $\hat{f}_{tgt}$ and between the reference features f_{ref} obtained by encoding the reference frame I_{ref} and the synthesized reference features $\hat{f}_{ref}$. Here, we define this constraint as the bidirectional feature perception loss, formulated as shown in formula (15).

$$L_{feat}^{bi} = \lambda_{feat}^{tgt} * \|f_{tgt} - \hat{f}_{tgt}\| + \lambda_{frat}^{ref} * \|f_{ref} - \hat{f}_{ref}\| \quad (15)$$

C. Bidirectional Depth Structure Consistency Loss

Given a target image I_{tgt}, the corresponding transformed world coordinates $\hat{img}_{xyz}^{ref} = (\hat{x}_w^{ref}, \hat{y}_w^{ref}, \hat{z}_w^{ref})$ can be obtained according to formula (1d). The depths in the target and reference images can be estimated by DepthNet and are denoted by d_{net}^{tgt} and d_{net}^{ref}, respectively. Because CameraNet is naturally coupled with DepthNet during training, the computed depth $\hat{z}_w^{ref}$ and the estimated depth d_{net}^{ref} should conform to the same 3D scene structure and should be consistent. However, as shown in Fig. 8, the depths $\hat{z}_w^{ref}$ and d_{net}^{ref} are not always equal, especially in regions with moving objects and occlusions. Intuitively, we can enforce consistency by minimizing the difference between $\hat{z}_w^{ref}$ and d_{net}^{ref}. In addition, moving objects and occlusions can be located using this difference. Formulas (1) and (3) show that the computed depth is only affected by camera pose, and the depth of another frame because the camera intrinsic is a fixed constant. Nevertheless, the predicted camera pose scale, inverse pose scale, and predicted depth scale are all unknown. Recent work [16] enforces predicted depth scale consistency between each consecutive image by geometry consistency constraint, whereas this constraint does not guarantee that the depth scale of adjacent frames is completely consistent because the predicted camera pose and predicted inverse pose have inconsistent scales and do not necessarily satisfy the condition of invertibility. For example, given two consecutive frames I_1, I_2, the predicted depth is D_1, D_2. Assume that the scale factors, which are used to align the predicted depth to the absolute scale depth, are s_{d1} and s_{d2} respectively. The predicted relative pose and the predicted relative inverse pose are T_1, T_2, and the corresponding scale factors are s_{t1}, s_{t2}, respectively. Based on formulas (1) and (3), we can compute the depth $\hat{D}_1$ and $\hat{D}_2$ of I_1 and I_2 respectively. And assume the scale factors are $\hat{s}_{d1}$ and $\hat{s}_{d2}$. Then they should satisfy the following relationships.

$$\hat{s}_{d1} = s_{t1} * s_{d2} \quad (16a)$$

$$\hat{s}_{d2} = s_{t2} * s_{d1} \quad (16b)$$

The constraint of $\|\hat{D}_1 - D_1\|$ and $\|\hat{D}_2 - D_2\|$ will only drive $\hat{s}_{d1}$ closer to s_{d1} and $\hat{s}_{d2}$ closer to s_{d2}. However, forcing $s_{t1} * s_{d2} = s_{d1}$ or $s_{t2} * s_{d1} = s_{d2}$ does not guarantee consistency of depth between adjacent frames because s_{t1} and s_{t2} are unknown and the predicted T_1 and the predicted T_2 are not guaranteed to be invertible. In order to ensure that this condition is satisfied, we explicitly constrain these two poses to be invertible by means of jointly estimating and computing, aiming at ensuring the depth scale of adjacent frames to be completely consistent. Note, however, that we cannot directly compute the difference between $\hat{z}_w^{ref}$ and d_{net}^{ref} because the estimated depth does not depend on the pixel grid. Therefore, we instead minimize the difference between $\hat{z}_w^{ref}$ and $d_{net\_interp}^{ref}$, which is obtained based on the predicted depth map d_{net}^{ref} and the grid coordinates $\hat{img}_{xy}^{ref} = (\hat{x}, \hat{y})$, obtained by formula (3), through bilinear interpolation. For pixel p_{ref}, the depth structure difference is defined as follows:

$$depth_{diff}^{ref \to tgt}(p_{ref}) = \frac{ERF(\hat{z}_w^{ref}, d_{net\_interp}^{ref}(p_{ref}))}{\hat{z}_w^{ref} + d_{net\_interp}^{ref}(p_{ref})} \quad (17)$$

where $ERF(\cdot)$ is the robust error function shown in formula (8).

Similarly, for pixel p_{tgt}, the corresponding transformed world coordinates $\hat{img}_{xyz}^{tgt} = (\hat{x}_w^{tgt}, \hat{y}_w^{tgt}, \hat{z}_w^{tgt})$ and image coordinates $\hat{img}_{xy}^{tgt}$ can be acquired, and then the depth structure difference between $\hat{z}_w^{tgt}$ and d_{net}^{tgt} can be obtained as shown in formula (18):

$$depth_{diff}^{tgt \to ref}(p_{tgt}) = \frac{ERF(\hat{z}_w^{tgt}, d_{net\_interp}^{tgt}(p_{tgt}))}{\hat{z}_w^{tgt} + d_{net\_interp}^{tgt}(p_{tgt})} \quad (18)$$

Then, the bidirectional depth structure consistency loss is defined as shown in formula (19), and the adaptive weights, which are obtained based on the depth differences, are defined as shown in formulas (20) and (21):

$$\begin{aligned} L_{dsc}^{bi} &= \lambda_{dsc}^{tgt} * L_{dsc}^{ref \to tgt} + \lambda_{dsc}^{ref} * L_{dsc}^{tgt \to ref} \\ &= \lambda_{dsc}^{tgt} * \frac{\sum depth_{diff}(p_{tgt})}{N_{tgt}} \\ &\quad + \lambda_{dsc}^{ref} * \frac{\sum depth_{diff}(p_{ref})}{N_{ref}} \end{aligned} \quad (19)$$

$$W_{aw}^{ref \to tgt} = 1 - \lambda_{aw}^{tgt} * depth_{diff}^{ref \to tgt}(p_{ref}) \quad (20)$$

$$W_{aw}^{tgt \to ref} = 1 - \lambda_{aw}^{ref} * depth_{diff}^{tgt \to ref}(p_{tgt}) \quad (21)$$

where N_{ref} and N_{tgt} denote the numbers of valid grid coordinates $\hat{img}_{xy}^{ref}$ and $\hat{img}_{xy}^{tgt}$ respectively.

D. Smoothness Loss

As is common practice [15], [16], [17], a smoothness loss is also employed as a regularizer for the estimated depth maps and feature maps. Here, an edge-aware term is used to weight the cost based on the depth map gradients and feature map

gradients. The smoothness loss is formulated as follows:

$$\begin{aligned} L_s^{bi} &= \lambda_f^{tgt} * \sum |\partial f_{tgt}| * e^{-|\partial I_{tgt}|} \\ &+\lambda_f^{ref} * \sum |\partial f_{ref}| * e^{-|\partial I_{ref}|} \\ &+\lambda_d^{ref} * \sum |\partial depth_{ref}| * e^{-|\partial I_{ref}|} \\ &+\lambda_d^{tgt} * \sum |\partial depth_{tgt}| * e^{-|\partial I_{tgt}|} \end{aligned} \tag{22}$$

E. Summary

The proposed CbwLoss is the bidirectional weighted photometric loss constrained by both the bidirectional feature perception loss and the bidirectional depth structure consistency loss, as shown in formula (23).

$$L_{Cbw} = L_{photo}^{biw} + L_{feat}^{bi} + L_{dsc}^{bi} \tag{23}$$

The total loss is shown in formula (24).

$$L_{total} = L_{Cbw} + L_s^{bi} \tag{24}$$

IV. Experiments

A. Dataset

We conducted experiments on the KITTI RAW dataset [49] and the KITTI Odometry dataset [50] DDAD dataset [23]. Similar to previous related work [15], [18], [19], [20], the KITTI RAW dataset was split as in [32], with approximately 40k frames used for training and 5k frames used for validation. The images were resized to 256×832 for depth estimation and camera pose estimation experiments. We evaluated DepthNet on test data consisting of 697 test frames in accordance with Eigen's testing split and tested CameraNet on sequences $09 - 10$ of the KITTI Odometry dataset. We also evaluate the proposed method on the improved ground-truth depths dataset [51]. The DDAD dataset was split as in [23], with 17,050 frames used for training and 4,150 frames used for evaluation.

B. Network Architectures

a) DepthNet: The U-Net architecture [52] with an encoder-decoder structure is adopted in our depth estimation network, which can extract both deep abstract feature information and local information. We use ResNet-50 [53] without a fully-connected layer as our encoder and finally output the deepest feature maps with a $1/32$ resolution relative to the input image after five rounds of subsampling. The decoder contains five convolution blocks, each consisting of a 3×3 convolutional layer with reflection padding and an exponential linear unit (ELU) nonlinear layer, followed by an upsampling layer. The decoder outputs feature maps with the same resolution as the input image after five rounds of upsampling. The feature maps of the last three scales are fed to a 3×3 convolutional layer with a sigmoid function for synthesizing multiscale images and are employed for estimating the corresponding depth maps via a 3×3 convolutional layer followed by a sigmoid function. The feature map of the maximum resolution extracted from the encoder is used for the feature perception loss. Finally, the predicted depths are constrained using $1/(\alpha * x + \beta)$ with $\alpha = 10$ and $\beta = 0.01$, following previous work [20].

b) CameraNet: CameraNet takes the image sequences concatenated from the target and reference frames along the channel dimension as input and outputs the relative camera poses between adjacent frames. For fairness, we use a similar network architecture as in [20] for our CameraNet, which consists of seven convolutional layers with stride 2, whose output is then fed to a 1×1 convolutional layer with $6 * N_{ref}$ output channels. Finally, we use global average pooling to aggregate the predictions at all spatial locations.

C. Training Details

The proposed learning framework was implemented using the PyTorch Library [54]. DepthNet and CameraNet were coupled by the loss function and trained jointly with a batch size of 2 and the learning rate of 10^{-4} using the AdamW [55] optimizer; during the testing phase, however, each model could be used separately. The corresponding lambda parameters in our experiments were shown in Tab. I.

We preprocessed the training set using random scaling, cropping and horizontal flipping. Then, the data to be used as input to the model were processed into the form of a tensor with a height of 256 and a width of 832. During training, following Ranjan et al. [15], five consecutive video frames were used as a training sample for model optimization, where the third image was regarded as the target image to calculate the losses with respect to the other four images, and the roles were then inverted to make the most of the limited available data. The model was trained for 150 epochs and validated in each epoch.

D. Performance Metrics

a) Monocular Depth Estimation: For depth evaluation, standard metrics from previous related work [20], [32], [40] were used, as shown in formula (25):

$$Abs\,Rel : \frac{1}{|D|} \sum_{d \in D} \frac{\|d^* - d\|}{d^*} \tag{25a}$$

$$RMSE : \sqrt{\frac{1}{|D|} \sum_{d \in D} \|d^* - d\|^2} \tag{25b}$$

$$Sq\,Rel : \frac{1}{|D|} \sum_{d \in D} \frac{\|d^* - d\|^2}{d^*} \tag{25c}$$

$$RMSElog : \sqrt{\frac{1}{|D|} \sum_{d \in D} \|logd^* - logd\|^2} \tag{25d}$$

$$\delta_t : \frac{1}{|D|} |\{d \in D | max(\frac{d^*}{d}, \frac{d}{d^*}) < 1.25^t\}| * 100\% \tag{25e}$$

where d^* and d denote the ground truth depth and the predicted depth, respectively. During the evaluation, the depth was capped at 50 m and 80 m in our experiments. To match the median with the ground truth, we needed to multiply the estimated depth maps by the scale factor computed from formula (26) following the method in [20] because the depth estimated from monocular videos using our method is defined

TABLE I
PARAMETER SETTINGS

Method	λ_p^{tgt}	λ_p^{ref}	λ_{occ}^{tgt}	λ_{occ}^{ref}	λ_{aw}^{tgt}	λ_{aw}^{ref}	λ_{dsc}^{tgt}	λ_{dsc}^{ref}	λ_{feat}^{tgt}	λ_{feat}^{ref}	λ_d^{tgt}	λ_d^{ref}	λ_f^{tgt}	λ_f^{ref}
Baseline	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.01	0.0	0.0	0.0
L_p^{bi}	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.01	0.01	0.0	0.0
$L_p + M_{occ}$	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.01	0.0	0.0	0.0
$L_p^{bi} + M_{occ}^{bi}$	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.01	0.01	0.0	0.0
$L_p + M_{occ} + L_{dsc}$	1.0	0.0	1.0	0.0	0.0	0.0	0.5	0.0	0.0	0.0	0.01	0.0	0.0	0.0
$L_p^{bi} + M_{occ}^{bi} + L_{dsc}^{bi}$	1.0	1.0	1.0	1.0	0.0	0.0	0.5	0.5	0.0	0.0	0.01	0.01	0.0	0.0
$L_p + M_{occ} + L_{dsc} + W_{aw}$	1.0	0.0	1.0	0.0	1.0	0.0	0.5	0.0	0.0	0.0	0.01	0.01	0.0	0.0
$L_p^{bi} + M_{occ}^{bi} + L_{dsc}^{bi} + W_{aw}^{bi}$	1.0	1.0	1.0	1.0	1.0	1.0	0.5	0.5	0.0	0.0	0.01	0.01	0.0	0.0
$L_p + M_{occ} + L_{dsc} + W_{aw} + L_{feat}$	1.0	0.0	1.0	0.0	1.0	0.0	0.5	0.0	0.05	0.0	0.01	0.0	0.001	0.0
$L_p^{bi} + M_{occ}^{bi} + L_{dsc}^{bi} + W_{aw}^{bi} + L_{feat}^{bi}$	1.0	1.0	1.0	1.0	1.0	1.0	0.5	0.5	0.05	0.05	0.01	0.01	0.001	0.001

only up to a scale factor.

$$scale = \frac{D_{median}^{gt}}{D_{median}^{pred}} \tag{26}$$

b) Camera Pose Estimation: For camera pose estimation, we used the absolute trajectory error (ATE) [56] as the performance metric. In our experiments, we computed the ATE by employing five frame snippets and optimized the scale factor according to formula (26) such that the predictions were best aligned with the ground truth.

E. Comparison With State-of-the-Art Methods

a) Monocular Depth Estimation: In Tab. II, we compare the depth estimation results with those of current state-of-the-art self-supervised methods trained on the KITTI dataset and with the results of methods with parameters pretrained on the Cityscapes dataset and then fine-tuned on KITTI, which are taken from the corresponding published papers. Depths capped at 50 m and 80 m were used to evaluate the model performance. The results show that the quality of the recovered depth map could be significantly improved by jointing learning different tasks (e.g., optical flow task [10], [15], [18], [21], [42], [65], segmentation task [10], [15], feature representation task [27], semantic learning task [43], [44]) utilizing different networks. Besides, it can be also seen from previous work [16], [17], [23], [24] that both the resolution of the input image and the adopted network architecture play an important role in dense depth estimation. We are more interested here in monocular methods where neither additional tasks are required nor the complexity of the model is increased. In the absence of additional auxiliary tasks and pretraining, our proposed method outperforms previous methods [16], [17], [17], [20], [28], [29], [45], except [23] where more complex network architecture is adopted. To be relatively fair, we also trained the network using the image with the same resolution as [23] and using additional training data. It shows that competitive results can be obtained under the same resolution conditions compared with the methods [23], while the depth estimator in reference [23] is four times as large as ours (see [23]). We believe that the better network architecture for DepthNet (e.g [23]) and CameraNet (e.g. [17]) is utilized, the greater improvements will be achieved in depth estimation. Moreover, although our approach introduces no additional information, it outperforms most previous methods [10], [15], [18], [21], [65]. We attribute this to the fact that our objective function can provide a better optimization direction for the network.

In addition, to further verify depth estimation results, in Tab. III, we also evaluate the model performance on the improved KITTI dataset [51]. It also shows that our proposed method outperforms previous methods [15], [16], [17], [20] and can also achieve competitive performance compared to the current state-of-the-art method [23] without additional auxiliary tasks.

In order to further quantitatively observe the robustness of the algorithm to moving objects and textureless regions, we select images with moving objects and those in which most areas are textureless from 697 test frames in accordance with Eigen's testing split, resulting in 282 test frames. Tab. IV reports the quantitative results of different methods evaluated on these challenging scenarios. It shows that our proposed method is more robust to these scenarios than the previous methods [15], [16], [17], [20], [23]. We suspect that this may be caused by the following reasons: 1) Auto-Mask scheme adopted in [17] and [23] only allows the network to ignore the contribution of objects, which move at the same velocity as the camera, to photometric loss. Nevertheless, this scheme is invalid when the moving object has a different translation speed from the camera. Our scheme based on the flow fields and depth structure could mitigate this impact. 2) The above methods are inefficient for large untextured areas due to not making full use of semantic and contextual information (e.g. the large white area in the seventh column in Fig. 2), while our proposed bidirectional feature perception loss could force the network to focus on these information. 3) Our quantitative result is slightly lower than that in [23] in 697 test frames, which should be due to the fact that more fine-grained information could be preserved by their proposed packing-unpacking blocks.

Tab. V reports the quantitative results evaluated on the DDAD dataset [23] which is a more realistic and challenging benchmark for depth estimation and contains more moving objects. It demonstrates that our proposed method outperforms prior work [23] by a big margin, which also proves the above conjecture from the side.

TABLE II

COMPARISON OF PERFORMANCE FOR MONOCULAR DEPTH ESTIMATION ON THE KITTI DATASET. '-' INDICATES THAT THE CAP PARAMETER IS NOT SPECIFIED IN THE CORRESPONDING PAPER.K DENOTES THAT OUR MODELS WERE TRAINED ONLY ON KITTI, AND CS/IN+K MEANS THAT THE MODELS WERE FINE-TUNED ON KITTI AFTER PRETRAINING ON THE CITYSCAPES/IMAGENET DATASET. THE BEST PERFORMANCE IN EACH COLUMN IS HIGHLIGHTED IN BOLD AND THE SECOND BEST IS HIGHLIGHTED IN ITALICS. 'DE/PE' REFERS TO THE BACKBONE OF THE USED ENCODER IN DEPTH/POSE ESTIMATION NETWORK, 'N' IS THE NUMBER OF TIMES THAT A PAIR OF IMAGES NEED TO BE REASONED ABOUT, 'M' REFERS TO METHODS THAT ONLY MONOCULAR(M) IMAGES ARE USED TO TRAIN NETWORK, 'RN50/RN18/VGG/HRNET' REFERS TO THE CORRESPONDING ENCODER BASED ON RESNET50 [53]/ RESNET18 [53]/ VGGNET [60]/ HRNET [61], 'DRN' REFERS TO THE DESIGNED DEPTH ESTIMATION NETWORK IN [15], 'PN7' REFERS TO A SIMPLE POSE ESTIMATION NETWORK CONSISTING OF SEVEN LAYERS CONVOLUTION BEING DESIGNED IN [20]. 'PACKNET' IS A MORE ADVANCED AND MORE COMPLEX NETWORK ARCHITECTURE [23] COMPARED WITH RESNET50 [53]. 'PWCNET' [62]/ 'FEATURENET' (BASED ON RESNET50 [53]) ARE THE CORRESPONDING ADDITIONAL NETWORKS REQUIRED FOR JOINTLY OPTIMIZING THE DEPTH ESTIMATION MODEL. 'ORBSLAM2' REFERS TO A METHOD THAT IS INTEGRATED WITH ORBSLAM2 [63] FOR OPTIMIZING THE PREDICTED DEPTHS AND POSES. 'SEMANTIC' REFERS TO METHODS THAT REQUIRE AN ADDITIONAL NETWORK (E.G. FEATURE PYRAMID NETWORK [64] WITH RESNET [53]) AND SEMANTIC LABEL TO GUIDE DEPTH ESTIMATION. ‡ INDICATES THE RESULTS, WHICH ARE DERIVED FROM GITHUB'S LATEST WEIGHT. '*' INDICATES THAT GROUP NORMALIZATION IS USED AFTER EACH CONVOLUTION LAYER IN PN7

Method	Data	Cap	Resolutions	DE	PE	N	Sup	Error↓				Accuracy↑		
								AbsRel	SqRel	RMSE	RMSElog	δ_1	δ_2	δ_3
Guizilini et al. [23]	K	80	192×640	PackNet	PN7	1	M+V	0.111	0.829	4.788	0.199	0.864	0.954	0.980
Guizilini et al. [23]	K+CS	80	192×640	PackNet	PN7	1	M+V	0.108	0.803	4.642	0.195	0.875	0.958	0.980
Luo et al. [21]	K	-	256×832	VGG	PN7	1	M+PWCNet	0.141	1.029	5.350	0.216	0.816	0.941	0.976
Wang et al. [10]	K	80	256×832	RN50	PN7	1	M+PWCNet+MaskNet	0.140	1.068	5.255	0.217	0.827	0.943	0.977
Ranjan et al. [15]	K+CS	80	256×832	DRN	PN7	1	M+PWCNet+MaskNet	0.139	1.032	5.199	0.213	0.827	0.943	0.977
Wang et al. [10]	K+CS	80	256×832	RN50	PN7	1	M+PWCNet+MaskNet	0.132	0.986	5.173	0.212	0.835	0.945	0.977
Zhao et al. [42]	K	-	256×832	RN18	-	-	M+PWCNet	0.130	0.893	5.062	0.205	0.832	0.949	0.981
Bian et al. [24]	K	80	256×832	RN50	RN18	2	M+ORBSLAM2	0.114	0.813	4.706	0.191	0.873	0.960	0.982
Shu et al. [27]	K	80	320×1024	RN50	RN18	1	M+FeatureNet	0.104	0.729	4.481	0.179	*0.893*	0.965	0.984
Ma et al. [57]	K	80	320×1024	RN50	RN18	1	M+Semantic	*0.099*	**0.624**	**4.165**	*0.171*	**0.902**	**0.969**	**0.986**
Petrovai et al. [58]	K	80	320×1024	RN50	RN18	1	M+Pseudo	**0.098**	*0.674*	*4.187*	**0.170**	**0.902**	*0.968*	*0.985*
Klingner et al. [43]	K+CS	80	384×1280	RN18	RN18	1	M+Semantic	0.107	0.768	4.468	0.186	0.891	0.963	0.982
Guizilini et al. [44]	K	80	384×1280	PackNet	PN7	1	M+Semantic	0.100	0.761	4.270	0.175	**0.902**	0.965	0.982
Godard et al. [17]	K	80	192×640	RN18	RN18	1	M	0.132	1.044	5.142	0.210	0.845	0.948	0.977
Godard et al.‡ [17]	K	80	192×640	RN50	RN50	1	M	0.131	1.020	5.060	0.206	0.849	0.951	0.979
Guizilini et al. [23]	K+IN	80	192×640	RN50	PN7*	1	M	0.117	0.900	4.826	0.196	0.873	-	-
Godard et al. [17]	K+IN	80	192×640	RN18	RN18	1	M	0.115	0.903	4.863	0.193	0.877	0.959	0.981
Guizilini et al. [23]	K	80	192×640	PackNet	PN7*	1	M	0.111	0.785	4.601	0.189	0.878	0.960	0.982
Godard et al.‡ [17]	K+IN	80	192×640	RN50	RN50	1	M	0.110	0.835	4.644	0.187	0.883	0.962	0.982
Guizilini et al. [23]	K+CS	80	192×640	PackNet	PN7*	1	M	0.108	*0.727*	4.426	0.184	0.885	0.963	*0.983*
He et al. [59]	K+IN	80	192×640	HRNet	RN18	1	M	**0.096**	**0.632**	**4.216**	**0.171**	**0.903**	**0.968**	**0.985**
Jia et al. [45]	K	80	256×832	RN18	PN7	1	M	0.136	0.895	4.834	0.199	0.832	0.950	0.982
Bian et al. [16]	K+CS	80	256×832	DRN	PN7	2	M	0.128	1.047	5.234	0.208	0.846	0.947	0.976
Gordon et al. [28]	K	80	-	RN18	RN18	2	M	0.128	0.959	5.23	0.212	0.845	0.947	0.976
Gordon et al. [28]	K+CS	80	-	RN18	RN18	2	M	0.124	0.930	5.12	0.206	0.851	0.950	0.978
Godard et al. [17]	K+IN	80	320×1024	RN18	RN18	1	M	0.115	0.882	4.701	0.190	0.879	0.961	0.982
Guizilini et al. [23]	K	80	384×1280	PackNet	PN7*	1	M	0.107	0.802	4.538	0.186	0.889	0.962	0.981
Guizilini et al. [23]	K+CS	80	384×1280	PackNet	PN7*	1	M	*0.104*	0.758	*4.386*	*0.182*	*0.895*	*0.964*	0.982
Ours	K	80	256×832	RN50	PN7	1	M	0.120	0.947	4.941	0.197	0.863	0.957	0.981
Ours	K+CS	80	256×832	RN50	PN7	1	M	0.110	0.847	4.654	0.189	0.882	0.960	0.981
Ours	K	80	384×1280	RN50	PN7	1	M	0.110	0.829	4.614	0.185	0.880	0.962	*0.983*
Ours	K+CS	80	384×1280	RN50	PN7	1	M	*0.104*	0.798	4.501	0.184	0.889	0.961	0.982
Luo et al. [21]	K	-	256×832	VGG	PN7	1	M+PWCNet	0.141	1.029	5.350	0.216	0.816	0.941	0.976
Zhao et al. [42]	K	-	256×832	RN18	-	-	M+PWCNet	0.130	0.893	5.062	0.205	0.832	0.949	0.981
Ours	K	50	256×832	RN50	PN7	1	M	0.116	0.817	4.025	0.188	0.876	0.962	*0.983*
Ours	K+CS	50	256×832	RN50	PN7	1	M	*0.105*	*0.649*	*3.571*	0.179	*0.895*	0.964	*0.983*
Ours	K	50	384×1280	RN50	PN7	1	M	0.106	0.716	3.745	*0.177*	0.892	**0.966**	**0.984**
Ours	K+CS	50	384×1280	RN50	PN7	1	M	**0.099**	**0.599**	**3.418**	**0.173**	**0.900**	*0.965*	**0.984**

In Tab. VI and VII, we compare the resource consumption of different models. As you can see from Tab. VI, the current state-of-the-art self-supervised monocular method [23] requires twice as much memory and takes four times as much reasoning time as ours under the same conditions during inferring. Furthermore, the previous work [23] takes up more memory and takes longer training times during training as shown in Tab. VII.

The qualitative results shown in Fig. 2 also prove that the proposed method outperforms the existing state-of-the-art self-supervised methods [15], [16], [17], [23] in the scene consisting of moving objects, occlusions, and textureless regions. More concretely, compared with the existing methods [15], [16], [17], [23], our method can estimate sharper and smoother scene depths, especially in areas where there are moving objects, occlusions, or textureless regions. For example, in the

TABLE III
COMPARISON OF PERFORMANCE FOR MONOCULAR DEPTH ESTIMATION ON THE IMPROVED KITTI DATASET [51]

Method	Data	Resolutions	Error↓				Accuracy↑		
			AbsRel	SqRel	RMSE	RMSElog	δ_1	δ_2	δ_3
Zhou et al. [20]	K+CS	128×416	0.176	1.532	6.129	0.244	0.758	0.921	0.971
Ranjan et al. [15]	K+CS	256×832	0.1049	0.6569	4.3128	0.1572	0.8869	0.9721	0.9914
Bian et al. [16]	K+CS	256×832	0.0984	0.6495	4.3975	0.1526	0.8917	0.9717	0.9906
Godard et al. [17]	K+IN	192×640	0.090	0.545	3.942	0.137	0.914	0.983	0.995
Guizilini et al. [23]	K+CS	192×640	*0.078*	**0.420**	**3.485**	*0.121*	*0.931*	**0.986**	**0.996**
Ours	K+CS	256×832	**0.0766**	*0.4249*	*3.5371*	**0.1207**	**0.9336**	*0.9849*	*0.9959*
Godard et al. [17]	K+IN	320×1024	0.0858	0.4619	3.5768	0.1270	0.9242	0.9861	0.9962
Guizilini et al. [23]	K+CS	384×1280	**0.071**	**0.359**	**3.153**	**0.109**	**0.944**	**0.990**	**0.997**
Ours	K+CS	384×1280	*0.0723*	*0.3767*	*3.3584*	*0.1147*	*0.9390*	*0.9872*	*0.9964*

TABLE IV
COMPARISON OF PERFORMANCE FOR MONOCULAR DEPTH ESTIMATION ON THE 282 IMAGES SELECTED FROM 697 IMAGES IN ACCORDANCE WITH EIGEN'S TESTING SPLIT. (272 IMAGES WITH MOVING OBJECTS AND 10 IMAGES IN WHICH MOST AREAS ARE TEXTURELESS)

Method	Data	Resolutions	Error↓				Accuracy↑		
			AbsRel	SqRel	RMSE	RMSElog	δ_1	δ_2	δ_3
Zhou et al. [20]	K+CS	128×416	0.1926	1.3625	6.1366	0.2767	0.7094	0.8999	0.9587
Ranjan et al. [15]	K+CS	256×832	0.1475	1.0216	4.9665	0.2265	0.8197	0.9392	0.9724
Bian et al. [16]	K+CS	256×832	0.1416	1.0717	5.1491	0.2299	0.8281	0.9367	0.9683
Godard et al. [17]	K+IN	192×640	0.1251	1.0205	4.8573	0.2137	0.8685	0.9522	0.9757
Guizilini et al. [23]	K+CS	192×640	*0.1209*	*0.9012*	*4.6110*	*0.2079*	*0.8716*	*0.9542*	*0.9768*
Ours	K+CS	256×832	**0.1145**	**0.8215**	**4.4561**	**0.2027**	**0.8793**	**0.9569**	**0.9780**
Godard et al. [17]	K+IN	320×1024	0.1249	0.9541	*4.5813*	0.2099	0.8686	*0.9553*	*0.9766*
Guizilini et al. [23]	K+CS	384×1280	*0.1159*	*0.8936*	4.5912	*0.2079*	**0.8824**	0.9539	0.9757
Ours	K+CS	384×1280	**0.1128**	**0.7848**	**4.3275**	**0.1994**	*0.8727*	**0.9565**	**0.9793**

TABLE V
COMPARISON OF PERFORMANCE FOR MONOCULAR DEPTH ESTIMATION ON THE DDAD DATASET [23]

Method	Resolutions	Error↓				Accuracy↑		
		AbsRel	SqRel	RMSE	RMSElog	δ_1	δ_2	δ_3
Guizilini et al. [23]	384×640	0.162	3.917	13.452	0.269	0.823	-	-
Ours	384×640	**0.1210**	**1.2288**	**6.7214**	**0.1872**	**0.8533**	**0.9575**	**0.9845**

example in the first column of Fig. 2, the brightness of the car in the depth map estimated by our method is closer to the brightness at the corresponding position in the ground truth depth map, while the brightnesses of this car as estimated by the previous methods [15], [16] are darker than that in the ground truth depth map. The windshield of the train in the second column and the large white areas in seventh column, which are a textureless region, are or close to black or not smooth enough in the depth maps predicted by the previous methods [15], [16], [17], [23]. However, the brightness at the same position in the depth map estimated by our method is very similar to that in the ground truth depth map. In other words, our method can work well even in textureless regions and accurately predict the depth of these regions, while the previous methods [15], [16], [17], [23] fail to correctly estimate the depth of such regions. In the third column, although the method [16] accurately estimates the depth of the black car in the image than the methods of [15] do, it has difficulty accurately predicting the depth in the intersection regions between the black car and the image background because of occlusion effects from the black car. The depth of the moving objects in the fifth, sixth and eighth columns is incorrectly estimated by the previous methods [15], [16], [17], [23].

In contrast, our method accurately estimates not only the depth of the black car in the image but also the depth of the intersection regions. In addition, the depth of more distant objects can also be accurately estimated, whereas the depths in the same positions as estimated by the previous methods are 'black holes' [16] or greater than the corresponding depth in the ground depth map [15]. More importantly, our method is robust not only to a small range of weakly textured regions (e.g., the front windshield of the train in the second

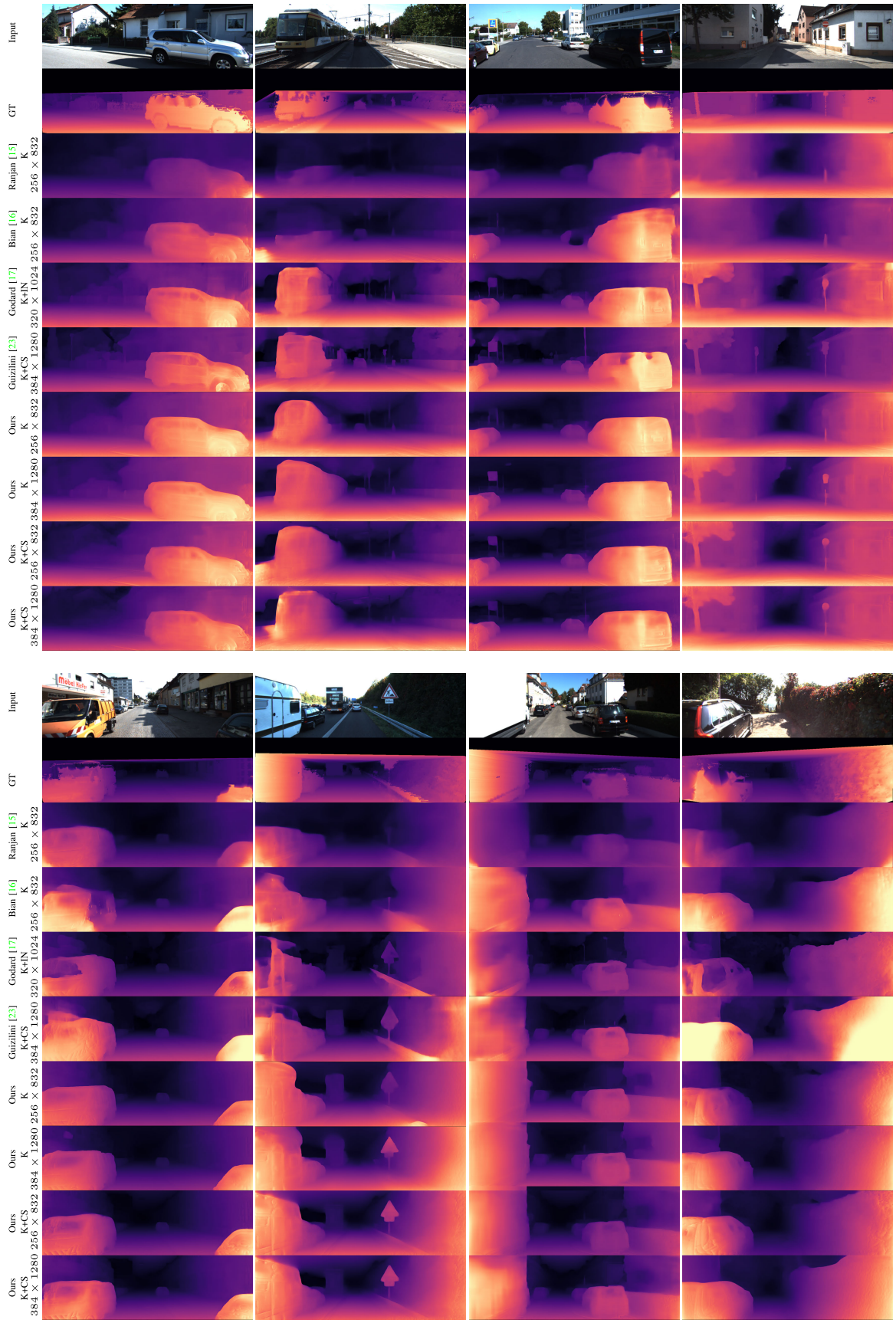

Fig. 2. Qualitative comparison of example results of our proposed self-supervised monocular depth estimation method with those of previous state-of-the-art methods as estimated on the KITTI dataset. The ground truth maps were obtained from sparse laser data for visualization only. The brighter an area in a depth map is, the closer it is to the camera.

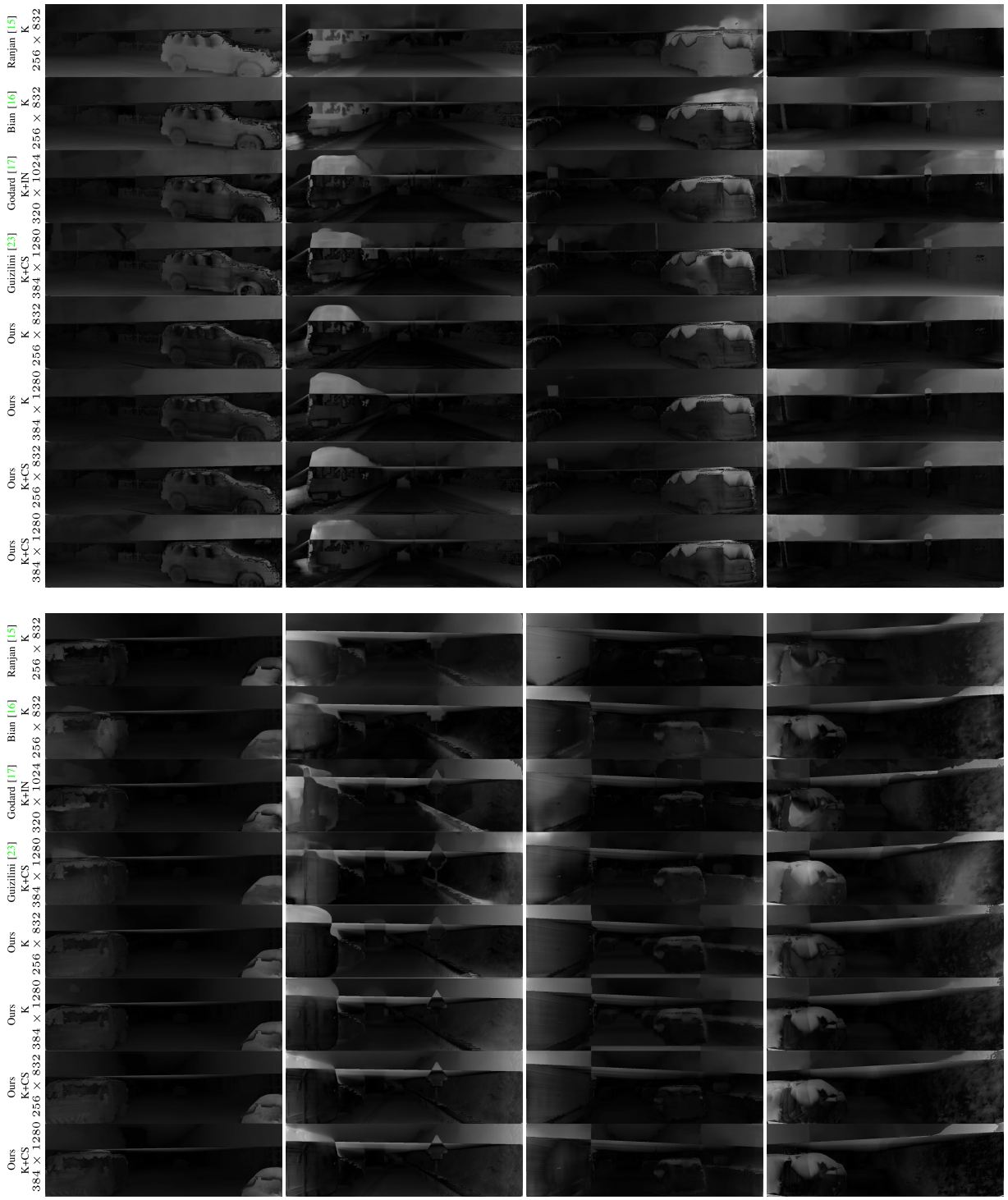

Fig. 3. Comparison of error maps between different methods on the KITTI dataset. The order of the figure here corresponds to that of Fig. 2.

column) but also to a large range of textureless regions (e.g., the white regions in the fourth and seventh column, where the previous methods of [15], [16], [17], [23], tend to predict a either more ambiguous or rougher depth map). In order to be easier to observe the difference between different methods, we also visualized the corresponding error map (The error map here is the absolute value of the difference between the estimated depth map and the ground-truth) shown in Fig. 3.

In Fig. 4, we compare the qualitative results with the previous method [23] on the DDAD dataset [23]. It shows that our proposed method could achieve more accurate depth than the previous method [23] in moving object regions. Similarly, we also provide corresponding error maps shown in Fig. 5 for observing their difference.

b) Camera Pose Estimation: In Tab. VIII, we compare the results of recent methods based on deep learning with the results of simultaneous localization and mapping

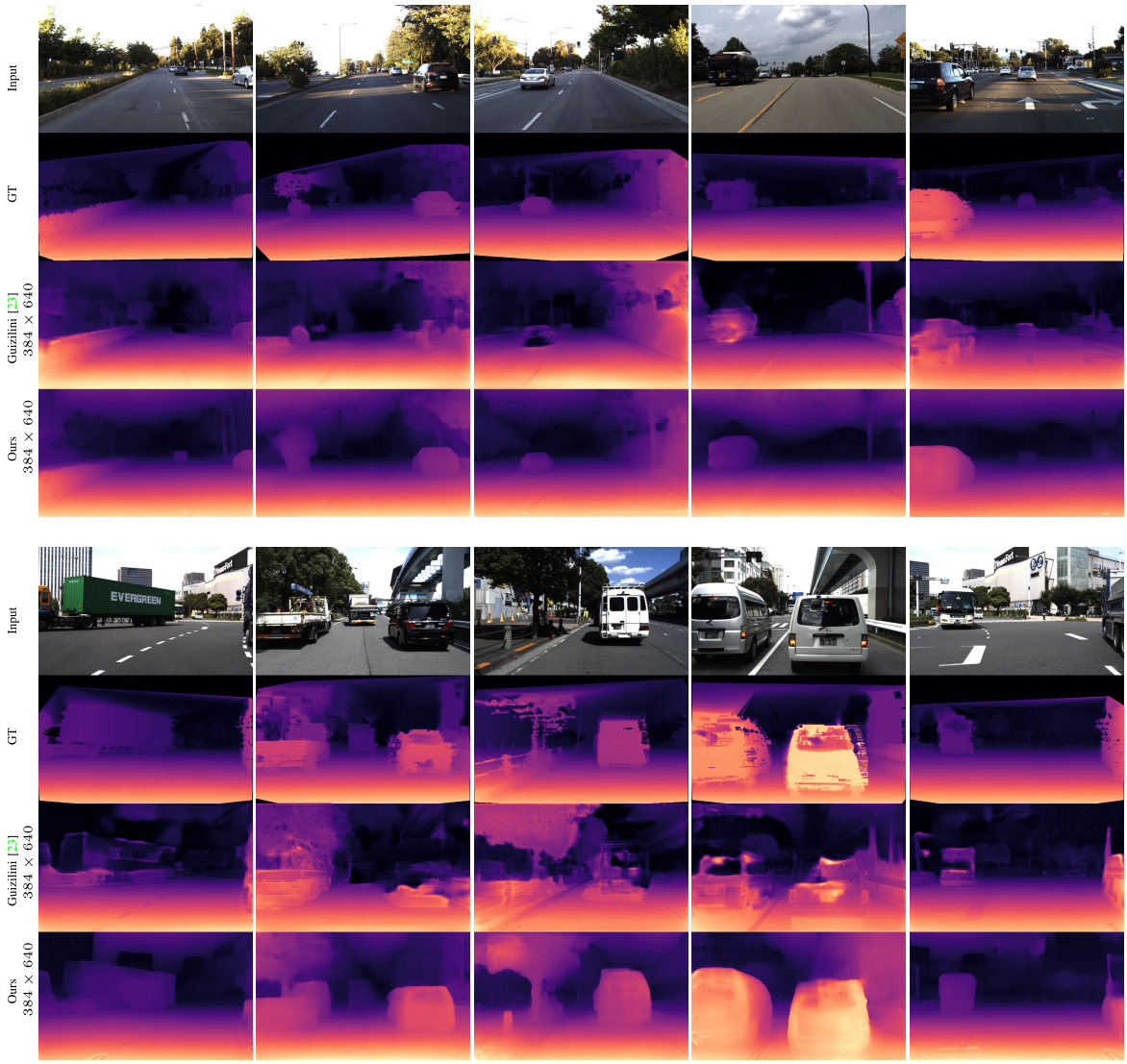

Fig. 4. Qualitative comparison of example results of our proposed self-supervised monocular depth estimation method with those of previous state-of-the-art methods as estimated on the DDAD dataset.

TABLE VI

COMPARISON OF OFFLINE INFER TIME ON THE DIFFERENT BACKBONES WITH BATCHSIZE=1. '*' INDICATES THAT GROUP NORMALIZATION IS USED AFTER EACH CONVOLUTION LAYER IN PN7. ALL RESULTS ARE EVALUATED ON RTX 3090TI WITH THE SAME SETTING. THE RESOLUTION IS SET TO 256 × 832. THE TIME OF DEPTHNET IS AVERAGED OVER 697 TEST FRAMES IN ACCORDANCE WITH EIGEN'S TESTING SPLIT, AND THE TIME OF CAMERANET IS AVERAGED OVER SEQUENCE 09 IN THE KITTI ODOMETRY DATASET. 'INFERT' INDICATES INFER TIME

	BackBone	Param(M)	InferT(ms)	GPU(M)
DepthNet	DRN	80.88	10.6	2389
	PackNet	128.29	49.3	3981
	RN50(Ours)	32.52	12.1	2143
CameraNet	PN7*	1.59	1.9	1899
	PN7(Ours)	1.59	1.5	1897
	RN18	13.01	4.5	1999

TABLE VII

COMPARISON OF TRAIN TIME ON THE DIFFERENT BACKBONES WITH BATCHSIZE=2. '*' INDICATES THAT GROUP NORMALIZATION IS USED AFTER EACH CONVOLUTION LAYER IN PN7. ALL RESULTS ARE EVALUATED ON RTX 3090TI WITH THE SAME SETTING. THE RESOLUTION IS SET TO 256 × 832. THE TIME IS AVERAGED 1000 ITERATIONS ON THE KITTI RAW DATASET

BackBone	TrainT(ms)	GPU(M)
PackNet+PN7*	1068.3	19817
RN50+PN7(Ours)	307.9	10013
RN50+RN18	321.7	10645

based on Oriented FAST and Rotated BRIEF features (ORB-SLAM) [56] as a reference. Our model can still achieve better results than ORB-SLAM (full) despite utilizing a rather short sequence. This performance improvement is attributed to the fact that high-level semantic features can be extracted in addition to low-level features. More importantly, although our CameraNet and the methods of [15], [20], [21], [65], and [66] all have the same network architecture, our method achieves more significant improvements in the ATE. This may be because our method benefits from the proposed objective function, which provides better constraints for network optimization.

Fig. 5. Comparison of error maps between different methods on the DDAD dataset. The order of the figure here corresponds to that of Fig. 4.

TABLE VIII

COMPARISON OF PERFORMANCE FOR CAMERA POSE ESTIMATION. THE RESULTS WERE TESTED ON SEQUENCES 09 AND 10 IN THE KITTI ODOMETRY DATASET. FOR ORB-SLAM (SHORT), ONLY FIVE FRAMES WERE TAKEN AS INPUT, WHEREAS FOR ORB-SLAM (FULL), THE ENTIRE SEQUENCE WAS TAKEN AS INPUT [56]. † INDICATES THAT THREE CONSECUTIVE FRAMES WITH A HEIGHT OF 384 AND A WIDTH OF 1280 WERE USED AS A TRAINING SAMPLE FOR DEPTH AND CAMERA POSE ESTIMATION EXPERIMENTS. ‡ INDICATES THE RESULTS, WHICH ARE DERIVED FROM GITHUB'S LATEST WEIGHT

Method	Seq. 09	Seq. 10
ORB-SLAM (short)	0.064±0.141	0.064±0.130
ORB-SLAM (full)	0.014±0.008	0.012±0.011
Mean Odometry	0.032±0.026	0.028±0.023
Zhou et al. [20]	0.021±0.017	0.020±0.015
Zou et al. [65]	0.017±0.007	0.015±0.009
Bian et al.‡ [16]	0.016±0.007	0.016±0.015
Godard et al.‡ [17]	0.021±0.009	0.014±0.010
Luo et al. [21]	0.013±0.007	0.012±0.008
Mahjourian et al. [66]	0.013±0.010	0.012±0.011
Ranjan et al. [15]	0.012±0.007	0.012±0.008
Ours	*0.0120±0.0068*	*0.0118±0.0081*
Ours†	**0.0084±0.0047**	**0.0084±0.0064**

Fig. 6. Qualitative examples of complementary occlusion masks. From top to bottom are the original images, the adaptive weights, and the bidirectional camera flow occlusion masks, respectively. In the first column, the adaptive weights fail to locate the black car in the lower left corner of the image, whereas the bidirectional camera flow occlusion masks succeed in locating the car. The second column shows the opposite situation.

F. Ablation Studies

To better understand the contribution of each element of the objective function proposed in section III — the bidirectional weighted photometric function, which is composed of the bidirectional photometric function (L_p^{bi}) with bidirectional camera flow occlusion masks (M_{occ}^{bi}) and adaptive weights (W_{aw}^{bi}), the bidirectional feature perception loss (L_{feat}^{bi}), and the bidirectional depth structure consistency loss (L_{dsc}^{bi}) — to the whole performance, we performed ablation studies, as shown in Tab. IX and Tab. X.

For our ablation studies, we jointly trained DepthNet and CameraNet with the same network architecture utilizing the proposed objective function combined in different ways in accordance with the idea of the control variable method. Note that the smoothness loss and the SSIM were used by default in all experiments. Tab. IX shows the depth estimation results within the range of 80 m and 50 m obtained with different objective function combinations. The corresponding camera pose estimation results are reported in Tab. X.

To analyze the performance changes caused by the proposed bidirectional photometric function, as the baseline method, we trained DepthNet and CameraNet with the same network architecture utilizing only the reconstruction error between the target view and the warped reference view. The results in Tab. IX indicate that the performance metric $\delta < 1.25$ could be significantly improved using the proposed bidirectional photometric function; simultaneously, the corresponding ATE of CameraNet was greatly reduced, as seen from the data in Tab. X. Moreover, to further verify the effect of the bidirectionality component on the objective function obtained by improving the normal photometric function utilizing the proposed other component, we also conducted additional experiments on the objective function composed of different unidirectionality components, such as the unidirectional photometric function (L_p), the unidirectional camera flow occlusion masks (M_{occ}), the unidirectional adaptive weights (W_{aw}), the unidirectional feature perception loss (L_{feat}), and the unidirectional depth structure consistency loss (L_{dsc}). The results show that the error indexes all have a different degree of decline compared

TABLE IX

ABLATION STUDIES ON MONOCULAR DEPTH ESTIMATION. THE RESULTS WERE EVALUATED ON THE KITTI EIGEN SPLIT WITH THE DEPTH CAPPED AT 80 M AND 50 M. δ REPRESENTS THE RATIO BETWEEN THE ESTIMATED DEPTH AND GROUND TRUTH DEPTHS. † INDICATES THAT THREE CONSECUTIVE FRAMES WITH A HEIGHT OF 384 AND A WIDTH OF 1280 WERE USED AS A TRAINING SAMPLE FOR DEPTH AND CAMERA POSE ESTIMATION EXPERIMENTS

Method	Cap (m)	Error↓				Accuracy↑		
		AbsRel	SqRel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Baseline	80	0.1418	0.9628	5.2890	0.2222	0.8081	0.9406	0.9768
L_p^{bi}	80	0.1390	1.0420	5.2572	0.2198	0.8272	0.9417	0.9749
$L_p + M_{occ}$	80	0.1385	0.9717	5.0650	0.2085	0.8349	0.9505	0.9810
$L_p^{bi} + M_{occ}^{bi}$	80	0.1262	0.9592	4.8118	0.2026	0.8566	0.9535	0.9795
$L_p + M_{occ} + L_{dsc}$	80	0.1271	1.0097	4.9408	0.2037	0.8545	0.9527	0.9794
$L_p^{bi} + M_{occ}^{bi} + L_{dsc}^{bi}$	80	0.1234	0.9984	4.9396	0.1988	0.8585	0.9548	0.9806
$L_p + M_{occ} + L_{dsc} + W_{aw}$	80	0.1222	1.0042	4.9935	0.1990	0.8651	0.9558	0.9799
$L_p^{bi} + M_{occ}^{bi} + L_{dsc}^{bi} + W_{aw}^{bi}$	80	0.1219	0.9833	4.9281	0.1980	0.8645	0.9558	0.9802
$L_p + M_{occ} + L_{dsc} + W_{aw} + L_{feat}$	80	0.1217	1.0233	5.0100	0.1991	0.8690	0.9557	0.9794
$L_p^{bi} + M_{occ}^{bi} + L_{dsc}^{bi} + W_{aw}^{bi} + L_{feat}^{bi}$	80	0.1199	0.9474	4.9405	0.1965	0.8630	0.9570	0.9814
$L_p^{bi} + M_{occ}^{bi} + L_{dsc}^{bi} + W_{aw}^{bi} + L_{feat}^{bi}$†	80	0.1099	0.8286	4.6139	0.1851	0.8801	0.9624	0.9828
Baseline	50	0.1370	0.7844	4.0926	0.2109	0.8235	0.9486	0.9796
L_p^{bi}	50	0.1345	0.8886	4.2324	0.2098	0.8399	0.9474	0.9772
$L_p + M_{occ}$	50	0.1333	0.8174	4.0294	0.1989	0.8492	0.9568	0.9828
$L_p^{bi} + M_{occ}^{bi}$	50	0.1228	0.8463	3.9583	0.1951	0.8675	0.9573	0.9808
$L_p + M_{occ} + L_{dsc}$	50	0.1230	0.8831	4.0035	0.1954	0.8692	0.9581	0.9805
$L_p^{bi} + M_{occ}^{bi} + L_{dsc}^{bi}$	50	0.1194	0.8794	4.0676	0.1906	0.8714	0.9596	0.9819
$L_p + M_{occ} + L_{dsc} + W_{aw}$	50	0.1185	0.8585	4.0925	0.1906	0.8772	0.9603	0.9815
$L_p^{bi} + M_{occ}^{bi} + L_{dsc}^{bi} + W_{aw}^{bi}$	50	0.1178	0.8575	4.0094	0.1894	0.8774	0.9608	0.9816
$L_p + M_{occ} + L_{dsc} + W_{aw} + L_{feat}$	50	0.1180	0.9032	4.1571	0.1912	0.8804	0.9601	0.9808
$L_p^{bi} + M_{occ}^{bi} + L_{dsc}^{bi} + W_{aw}^{bi} + L_{feat}^{bi}$	50	0.1155	0.8169	4.0249	0.1876	0.8758	0.9619	0.9830
$L_p^{bi} + M_{occ}^{bi} + L_{dsc}^{bi} + W_{aw}^{bi} + L_{feat}^{bi}$†	50	0.1060	0.7156	3.7449	0.1770	0.8917	0.9663	0.9840

Fig. 7. Visualization analysis of the camera flow occlusion masks. From top to bottom are the original images, the forward camera flows generated from the projection transformation of the reference images to the target images, the backward camera flows synthesized using the differentiable bilinear sampling mechanism, the bidirectional camera flow occlusion masks obtained by checking the consistency of the above two flows, and the estimated depth maps, respectively.

with the corresponding unidirectional method, except *SqRel*, which is generated from *Baseline* and L_p^{bi}, respectively. We hypothesized that this phenomenon might be caused by the presence of occluding objects in the scene. As analyzed above, the absolute trajectory error has a similar trend.

To better understand whether the bidirectional camera flow occlusion masks and adaptive weights play important roles in handling moving objects and occlusions in a scene during inference, we visualized these model components as shown in Fig. 7 and Fig. 8. Both the bidirectional camera flow occlusion masks and the adaptive weights can effectively locate moving objects and occlusions in a scene as seen from Fig. 7 and Fig. 8. In addition, it can be seen from the quantitative experimental results in Tab. IX and Tab. X that the performances of both DepthNet and CameraNet are individually improved. Therefore, the moving objects and occlusions in a scene can be well handled. As a result, the implicit assumptions necessary for view synthesis — that there are no moving objects or occlusions in the scene of interest — can be satisfied. Note that although moving objects and occlusions can be located using these two components of our method, they cannot always be located successfully if only one component is used (see, e.g., Fig. 6). This phenomenon is also supported by the quantitative results shown in Tab. IX. More importantly, the bidirectional

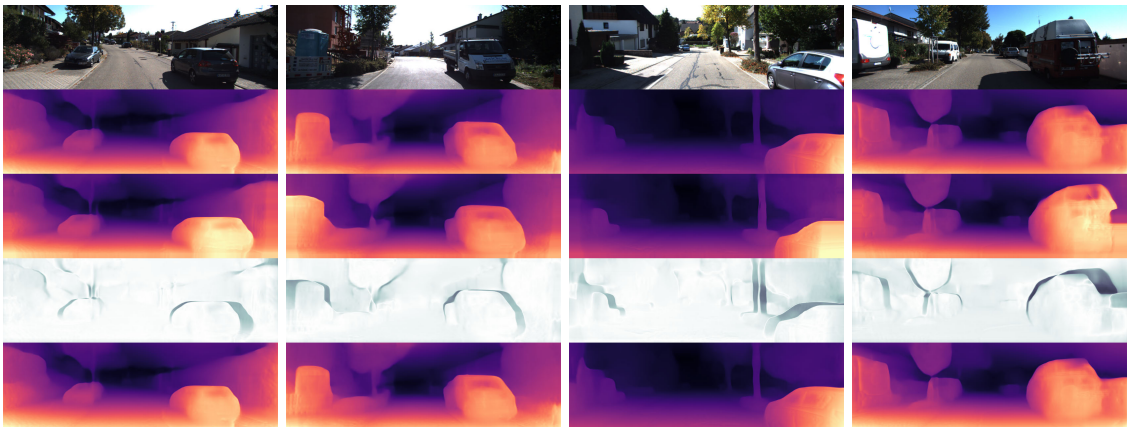

Fig. 8. Visualization analysis of the adaptive weights. From top to bottom are the original images, the depth maps obtained through projection transformation, the depth maps obtained through view synthesis, the adaptive weights obtained by comparing the above two depth maps, and the estimated depth maps, respectively.

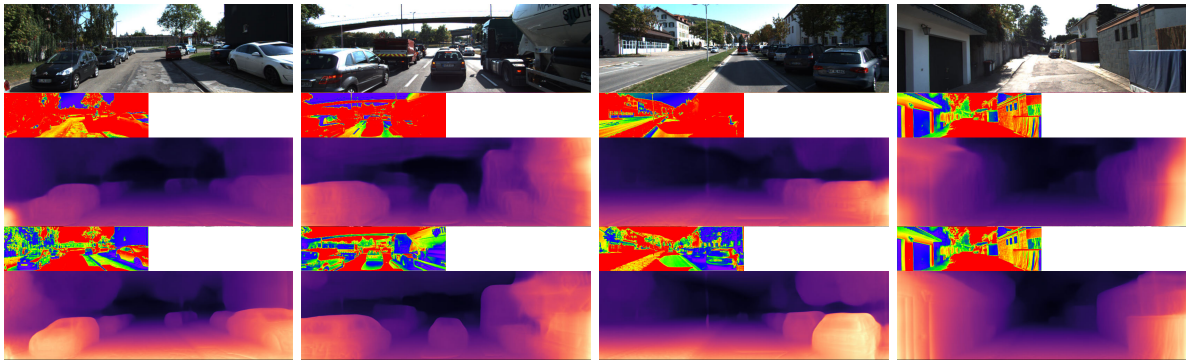

Fig. 9. Visualization analysis of the learned feature maps. Here, we select only the principal feature map for visualization utilizing principal component analysis. From top to bottom are the original images, the feature maps without the bidirectional feature perception loss, the depth maps without the bidirectional feature perception loss, the feature maps with the bidirectional feature perception loss, and the depth maps with the bidirectional feature perception loss, respectively.

TABLE X

ABLATION STUDIES ON CAMERA POSE ESTIMATION. THE RESULTS WERE TESTED ON SEQUENCES 09 AND 10 IN THE KITTI ODOMETRY DATASET. † INDICATES THAT THREE CONSECUTIVE FRAMES WITH A HEIGHT OF 384 AND A WIDTH OF 1280 WERE USED AS A TRAINING SAMPLE FOR DEPTH AND CAMERA POSE ESTIMATION EXPERIMENTS

Method	Seq. 09	Seq. 10
Baseline	0.0369±0.0369	0.0257±0.0271
L_p^{bi}	0.0133±0.0066	0.0120±0.0086
$L_p + M_{occ}$	0.0170±0.0062	0.0144±0.0085
$L_p^{bi} + M_{occ}^{bi}$	0.0130±0.0064	0.0119±0.0086
$L_p + M_{occ} + L_{dsc}$	0.0167±0.0066	0.0142±0.0086
$L_p^{bi} + M_{occ}^{bi} + L_{dsc}^{bi}$	0.0129±0.0067	0.0120±0.0085
$L_p + M_{occ} + L_{dsc} + W_{aw}$	0.0154±0.0057	0.0133±0.0083
$L_p^{bi} + M_{occ}^{bi} + L_{dsc}^{bi} + W_{aw}^{bi}$	0.0123±0.0065	0.0119±0.0083
$L_p + M_{occ} + L_{dsc} + W_{aw} + L_{feat}$	0.0148±0.0061	0.0128±0.0083
$L_p^{bi} + M_{occ}^{bi} + L_{dsc}^{bi} + W_{aw}^{bi} + L_{feat}^{bi}$	0.0120±0.0068	0.0118±0.0081
$L_p^{bi} + M_{occ}^{bi} + L_{dsc}^{bi} + W_{aw}^{bi} + L_{feat}^{bi}$†	0.0084±0.0047	0.0084±0.0064

TABLE XI

COMPARISON OF ACQUISITION METHODS OF THE INVERSE POSE WITH BATCHSIZE = 2. ALL RESULTS ARE EVALUATED ON RTX 3090TI WITH THE SAME SETTING. THE RESOLUTION IS SET TO 256 × 832. THE TIME IS AVERAGED 1000 ITERATIONS ON THE KITTI RAW DATASET. 'CINV/PINV' INDICATES THAT COMPUTE INVERSE POSE/PREDICT INVERSE POSE

CameraNet	Scheme	TrainT(ms)	GPU(M)
PN7	Pinv	116.9	2903
	Cinv(Ours)	74.5	2815
RN18	Pinv	151.3	3935
	Cinv(Ours)	80.1	3429

depth structure consistency constraint employed to obtain the adaptive weights can also improve the quality of the estimated depth maps, as seen in Tab. IX.

To analyze the effects of the bidirectional feature perception loss on the model, we selected the principle feature map formed from the features extracted by the encoder and analyzed it using principal component analysis. The visualization results are shown in Fig. 9, where the original images, the feature maps without bidirectional the feature perception loss, the depth maps without the bidirectional feature perception loss, the feature maps with the bidirectional feature perception loss, and the depth maps with the bidirectional feature perception loss are sequentially shown in the first to fifth rows. As seen from Fig. 9, compared to those learned without the bidirectional feature perception loss, the visual representations learned with the bidirectional feature perception loss show larger variations in textureless regions, such as the pure white/black cars in the first column, the white oil tank in the second column, the the shadows of the cars in the second and third columns, and the wall in the fourth column. The corresponding estimated depth maps are also smoother and sharper, consistent with the quantitative results in Tab. IX.

In Tab. XI, we investigate the differences between acquisition methods of the inverse pose. The results in Tab. XI shows that calculating the inverse pose is better than predicting the inverse pose in terms of both training time and required GPU. Furthermore, this advantage becomes more significant as the depth of the model increases. More importantly, compared with the scheme of predicting the inverse using the network, both the forward and backward transformations are explicitly constrained to be invertible and the scale of both the forward and backward poses is also explicitly constrained to be consistent.

V. Conclusion

In this paper, we have presented an end-to-end self-supervised learning pipeline that utilizes the task of view synthesis to obtain the supervision signal for depth and camera pose estimation from unlabeled monocular video. Our experimental results indicate that the proposed method outperforms previous related work. The proposed bidirectional weighted photometric loss function can fully reveal the information captured by the limited available data and handle dynamic scenes effectively. Second, textureless regions in a scene can be given more attention by using our feature perception loss function. Moreover, we can enforce consistency between depth maps, further improving the quality of the depth estimates. Despite competitive performance in a benchmark evaluation, the scale-drift issue still remains, causing us to need to align our estimation results with the ground truth during evaluation. Additionally, our method assumes that the camera intrinsics are given, thus preventing its application to arbitrary Internet videos acquired with unknown camera types. We plan to address these problems in future work.

References

[1] Y. D. V. Yasuda, L. E. G. Martins, and F. A. M. Cappabianco, "Autonomous visual navigation for mobile robots: A systematic literature review," *ACM Comput. Surv.*, vol. 53, no. 1, pp. 1–34, 2020.

[2] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, "A survey on 3D object detection methods for autonomous driving applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3782–3795, Oct. 2019.

[3] L. Wang, X. Fan, J. Chen, J. Cheng, J. Tan, and X. Ma, "3D object detection based on sparse convolution neural network and feature fusion for autonomous driving in smart cities," *Sustain. Cities Soc.*, vol. 54, Mar. 2020, Art. no. 102002.

[4] C. Eom, H. Park, and B. Ham, "Temporally consistent depth prediction with flow-guided memory units," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4626–4636, Nov. 2020.

[5] J. Chen, X. Yang, Q. Jia, and C. Liao, "DENAO: Monocular depth estimation network with auxiliary optical flow," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 8, pp. 2598–2610, Aug. 2021.

[6] X. Qi, Z. Liu, R. Liao, P. H. S. Torr, R. Urtasun, and J. Jia, "GeoNet++: Iterative geometric neural network with edge-aware refinement for joint depth and surface normal estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 2, pp. 969–984, Feb. 2022.

[7] K. Park, S. Kim, and K. Sohn, "High-precision depth estimation using uncalibrated LiDAR and stereo fusion," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 321–335, Jan. 2020.

[8] W. Su, H. Zhang, Q. Zhou, W. Yang, and Z. Wang, "Monocular depth estimation using information exchange network," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3491–3503, Jun. 2021.

[9] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2002–2011.

[10] G. Wang, C. Zhang, H. Wang, J. Wang, Y. Wang, and X. Wang, "Unsupervised learning of depth, optical flow and pose with occlusion from 3D geometry," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 308–320, Jan. 2022.

[11] Y. Zhang, S. Xu, B. Wu, J. Shi, W. Meng, and X. Zhang, "Unsupervised multi-view constrained convolutional network for accurate depth estimation," *IEEE Trans. Image Process.*, vol. 29, pp. 7019–7031, 2020.

[12] N. Yang, L. von Stumberg, R. Wang, and D. Cremers, "D3 VO: Deep depth, deep pose and deep uncertainty for monocular visual odometry," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1281–1292.

[13] S. Li, X. Wang, Y. Cao, F. Xue, Z. Yan, and H. Zha, "Self-supervised deep visual odometry with online adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6339–6348.

[14] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova, "Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 8001–8008.

[15] A. Ranjan et al., "Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12240–12249.

[16] J.-W. Bian et al., "Unsupervised scale-consistent depth and ego-motion learning from monocular video," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2019, pp. 35–45.

[17] C. Godard, O. M. Aodha, M. Firman, and G. Brostow, "Digging into self-supervised monocular depth estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3828–3838.

[18] Z. Yin and J. Shi, "GeoNet: Unsupervised learning of dense depth, optical flow and camera pose," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1983–1992.

[19] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 270–279.

[20] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1851–1858.

[21] C. Luo et al., "Every pixel counts ++: Joint learning of geometry and motion with 3D holistic understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2624–2641, Oct. 2020.

[22] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1–9.

[23] V. Guizilini, R. Ambrus, S. Pillai, A. Raventos, and A. Gaidon, "3D packing for self-supervised monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2485–2494.

[24] J.-W. Bian et al., "Unsupervised scale-consistent depth learning from video," *Int. J. Comput. Vis.*, vol. 129, no. 9, pp. 2548–2564, Sep. 2021.

[25] J. Watson, M. Firman, G. Brostow, and D. Turmukhambetov, "Self-supervised monocular depth hints," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2162–2171.

[26] J. Zhou, Y. Wang, K. Qin, and W. Zeng, "Unsupervised high-resolution depth learning from videos with dual networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6872–6881.

[27] C. Shu, K. Yu, Z. Duan, and K. Yang, "Feature-metric loss for self-supervised learning of depth and egomotion," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 572–588.

[28] A. Gordon, H. Li, R. Jonschkowski, and A. Angelova, "Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8977–8986.

[29] H. Li, A. Gordon, H. Zhao, V. Casser, and A. Angelova, "Unsupervised monocular depth learning in dynamic scenes," in *Proc. Conf. Robot. Learn.*, vol. 155, pp. 1908–1917. PMLR, 16–18, Nov. 2021.

[30] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 694–711.

[31] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. M. Reid, "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 340–349.

[32] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 2366–2374.

[33] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2024–2039, Oct. 2016.
[34] K. Konda and R. Memisevic, "Learning visual odometry with a convolutional network," in *Proc. 10th Int. Conf. Comput. Vis. Theory Appl.*, 2015, pp. 486–490.
[35] S. Wang, R. Clark, H. Wen, and N. Trigoni, "DeepVO: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 2043–2050.
[36] S. Wang, R. Clark, H. Wen, and N. Trigoni, "End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks," *Int. J. Robot. Res.*, vol. 37, nos. 4–5, pp. 513–542, 2018.
[37] F. Xue, X. Wang, S. Li, Q. Wang, J. Wang, and H. Zha, "Beyond tracking: Selecting memory and refining poses for deep visual odometry," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8575–8583.
[38] A. Beauvisage, K. Ahiska, and N. Aouf, "Multimodal tracking framework for visual odometry in challenging illumination conditions," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 11133–11139.
[39] X. Ju, D. Xu, and H. Zhao, "Scene-aware error modeling of LiDAR/visual odometry for fusion-based vehicle localization," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6480–6494, Jul. 2022.
[40] R. Garg, V. K. Bg, G. Carneiro, and I. Reid, "Unsupervised CNN for single view depth estimation: Geometry to the rescue," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 740–756.
[41] H. Zhan, C. S. Weerasekera, R. Garg, and I. Reid, "Self-supervised learning for single view depth and surface normal estimation," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 4811–4817.
[42] W. Zhao, S. Liu, Y. Shu, and Y.-J. Liu, "Towards better generalization: Joint depth-pose learning without PoseNet," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9151–9161.
[43] M. Klingner, J.-A. Termöhlen, J. Mikolajczyk, and T. Fingscheidt, "Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 582–600.
[44] V. Guizilini, R. Hou, J. Li, R. Ambrus, and A. Gaidon, "Semantically guided representation learning for self-supervised monocular depth," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–14.
[45] S. Jia, X. Pei, X. Jing, and D. Yao, "Self-supervised 3D reconstruction and ego-motion estimation via on-board monocular video," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7557–7569, Jul. 2022.
[46] S. Pillai, R. Ambrus, and A. Gaidon, "SuperDepth: Self-supervised, super-resolved monocular depth estimation," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 9250–9256.
[47] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[48] S. Meister, J. Hur, and S. Roth, "Unflow: Unsupervised learning of optical flow with a bidirectional census loss," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, 2018, pp. 1–9.
[49] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.
[50] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.
[51] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant CNNs," in *Proc. Int. Conf. 3D Vis. (3DV)*, Oct. 2017, pp. 11–20.
[52] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput-Assist. Intervent*. Cham, Switzerland: Springer, 2015, pp. 234–241.
[53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[54] A. Paszke et al., "Automatic differentiation in Pytorch," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–4.
[55] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–19.
[56] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.
[57] J. Ma, X. Lei, N. Liu, X. Zhao, and S. Pu, "Towards comprehensive representation enhancement in semantics-guided self-supervised monocular depth estimation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 304–321.
[58] A. Petrovai and S. Nedevschi, "Exploiting pseudo labels in a self-supervised learning framework for improved monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1578–1588.
[59] M. He, L. Hui, Y. Bian, J. Ren, J. Xie, and J. Yang, "RA-depth: Resolution adaptive self-supervised monocular depth estimation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 565–581.
[60] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.
[61] J. Wang et al., "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, Oct. 2020.
[62] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 8934–8943.
[63] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source slam system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.
[64] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.
[65] Y. Zou, Z. Luo, and J.-B. Huang, "DF-Net: Unsupervised joint learning of depth and flow using cross-task consistency," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 36–53.
[66] R. Mahjourian, M. Wicke, and A. Angelova, "Unsupervised learning of depth and ego-motion from monocular video using 3D geometric constraints," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 5667–5675.

Fei Wang (Student Member, IEEE) is currently pursuing the Ph.D. degree with the University of Chinese Academy of Sciences and the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences. His current research interests include computer vision, structure from motion, robotics, and deep learning.

Jun Cheng (Member, IEEE) received the B.Eng. and M.Eng. degrees from the University of Science and Technology of China, Hefei, China, in 1999 and 2002, respectively, and the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 2006. He is currently with the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China, as a Professor, and the Director of the Laboratory for Human Machine Control. His current research interests include computer vision, robotics, machine intelligence, and control.

Penglei Liu (Student Member, IEEE) is currently pursuing the Ph.D. degree with the University of Chinese Academy of Sciences and the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences. His current research interests include robot control, neural network applications, and machine learning.