remote sensing

MDPI

Article

Switchable-Encoder-Based Self-Supervised Learning Framework for Monocular Depth and Pose Estimation

Junoh Kim [1], Rui Gao [1], Jisun Park [1], Jinsoo Yoon [2] and Kyungeun Cho [3,*]

[1] Department of Multimedia Engineering, Dongguk University-Seoul, 30, Pildongro-1-gil, Jung-gu, Seoul 04620, Republic of Korea; junoh.kim@dongguk.edu (J.K.); gaorui@dongguk.edu (R.G.); jisun@dongguk.edu (J.P.)
[2] Autonomous Driving Research Department, KoROAD (Korea Road Traffic Authority) 2, Hyeoksin-ro, Wonu-si, Gangwon-do 26466, Republic of Korea; nametree@koroad.or.kr
[3] Division of AI Software Convergence, Dongguk University-Seoul, 30, Pildongro-1-gil, Jung-gu, Seoul 04620, Republic of Korea
* Correspondence: cke@dongguk.edu

Abstract: Monocular depth prediction research is essential for expanding meaning from 2D to 3D. Recent studies have focused on the application of a newly proposed encoder; however, the development within the self-supervised learning framework remains unexplored, an aspect critical for advancing foundational models of 3D semantic interpretation. Addressing the dynamic nature of encoder-based research, especially in performance evaluations for feature extraction and pre-trained models, this research proposes the switchable encoder learning framework (SELF). SELF enhances versatility by enabling the seamless integration of diverse encoders in a self-supervised learning context for depth prediction. This integration is realized through the direct transfer of feature information from the encoder and by standardizing the input structure of the decoder to accommodate various encoder architectures. Furthermore, the framework is extended and incorporated into an adaptable decoder for depth prediction and camera pose learning, employing standard loss functions. Comparative experiments with previous frameworks using the same encoder reveal that SELF achieves a 7% reduction in parameters while enhancing performance. Remarkably, substituting newly proposed algorithms in place of an encoder improves the outcomes as well as significantly decreases the number of parameters by 23%. The experimental findings highlight the ability of SELF to broaden depth factors, such as depth consistency. This framework facilitates the objective selection of algorithms as a backbone for extended research in monocular depth prediction.

Keywords: structure from motion; self-supervised learning; monocular depth estimation

check for updates

Citation: Kim, J.; Gao, R.; Park, J.; Yoon, J.; Cho, K. Switchable-Encoder-Based Self-Supervised Learning Framework for Monocular Depth and Pose Estimation. *Remote Sens.* **2023**, *15*, 5739. https://doi.org/10.3390/rs15245739

Academic Editor: Chiman Kwan

Received: 4 September 2023
Revised: 29 November 2023
Accepted: 13 December 2023
Published: 15 December 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Monocular depth prediction studies have demonstrated their significance in various sectors, including autonomous robots, surveillance, healthcare, construction, and production. These studies are fundamental in extending the semantic interpretation of 2D data, facilitating advanced analyses such as 3D object detection, tracking, volumetric prediction, and ground detection for specific industrial requirements. Although monocular cameras can acquire data from a broad spectrum of environments, including car black boxes and CCTV systems, the challenge of dense depth prediction persists, primarily due to information loss during data collection.

The progression of depth prediction research originated from the hypothesis that humans can ascertain relative distances in a single image through learned experience [1,2]. This premise was adapted into a learning-based challenge for artificial intelligence algorithms, yielding remarkable results in depth prediction. Progress in research in stereo camera data suggests a shift from data-driven supervised learning to self-supervised learning, addressing issues such as the high cost of training data production and the scarcity of

depth data from LiDAR sensors [3–6]. Earlier studies aimed to transcend the constraints of data acquisition and the reliance on stereo cameras by incorporating monocular camera data [7–11]. Nonetheless, this shift introduced several new research challenges:

(1) Estimation of relative poses between images;
(2) Absence of ground truth for direct loss calculation, necessitating the creation of synthetic data;
(3) Extraction of data that impedes learning, necessitating the removal of featureless or non-displaced data;
(4) Consideration of biased occlusions caused by moving objects and camera movements [11];
(5) Accommodation for non-common data between two images arising from alterations in camera pose.

Previous research efforts have jointly trained a camera pose network to address problem (1), leveraging feature matching between two consecutive datasets and designing around depth prediction loss functions. Challenges (2), (3), and (5) were tackled by enhancing the synthetic data generation module and refining the loss function for self-supervised learning applications, while problem (4) was addressed using static scene data [7,12]. The separate employment of optical flow [2,8,11] and segmentation prediction algorithms [13–15] has been observed, with recent studies shifting towards loss-function-based methodologies [16,17].

Despite notable advancements in monocular depth prediction research, the field still lacks a comprehensive end-to-end learning framework. Prior studies have primarily concentrated on decoder research, adopting a model where a newly proposed algorithm is pre-trained for feature extraction and then implemented as a foundational element. However, accurately evaluating the distinct contributions of the encoder, user-defined decoder, loss function, synthetic data generation module, training methodology, and camera pose algorithm to the enhancement of depth prediction remains a formidable challenge. The requirement to modify or rebuild all components of self-supervised learning in response to the introduction of a new encoder further complicates extended research in monocular depth predictions.

In our study, we perform an exhaustive review of previous monocular self-supervised learning research and introduce the switchable encoder learning framework (SELF), a self-supervised learning framework designed to promote the progression of monocular depth prediction studies. SELF integrates an adaptable decoder, enabling the immediate application of algorithms with demonstrated efficacy in feature extraction for classification tasks as encoders. The loss function and synthetic data generation processes are tailored to the decoder, eliminating the need for extensive alterations. Although self-supervised learning methods encounter limitations in concurrent depth and pose prediction, our approach enhances pose prediction accuracy through a specialized loss function developed for both camera pose and depth prediction. This method effectively advances the depth factor, known as depth consistency.

The contributions of this study are outlined as follows:

(1) By integrating components from previous self-supervised learning research with the newly introduced decoder into our self-supervised learning framework, the proposed encoder becomes readily applicable to dense prediction tasks;
(2) Adaptive decoders, characterized by standardized long skip connections, facilitate the utilization of variously structured encoders, including pre-trained models, without necessitating adjustments. This permits an unbiased comparison of feature extraction capabilities in dense prediction;
(3) Revolving around the adaptive decoder, each element is constructed with standardized components, enhancing its utility for further 3D research and streamlining the process of selecting a backbone, thereby reducing additional research time.

The structure of this paper is organized in the following manner. Section 2 delineates the components of self-supervised learning in monocular depth prediction, drawing on previous research; this includes data preprocessing, augmentation, and synthetic image generation, as well as the encoder, decoder, loss function, and camera pose prediction. Section 3 describes the proposed self-supervised learning framework, following the sequence established in Section 2. Section 4 involves a comparative analysis of the performance of our learning framework through experiments; it examines the depth performance in relation to the switchable encoder and elucidates how the expressive capability of the decoder and the scale depth loss function within the learning framework contribute to the outcomes.

2. Related Works

SELF primarily encompasses synthetic data for self-supervised learning (Section 2.1), an encoder (Section 2.2), a decoder (Section 2.3), self-supervised loss functions including depth consistency (Section 2.4), and camera pose prediction (Section 2.5). This section provides a comparative analysis of existing research on each of these elements.

2.1. Synthetic Data for Self-Supervised Learning

Self-supervised learning, a method that derives ground truth from provided data, offers a cost-efficient alternative for reducing data labeling expenses and addresses the challenge of diversity in the training dataset [18,19]. In scenarios lacking direct ground truth comparisons, the creation of synthetic input data based on algorithm-predicted data becomes a crucial aspect of monocular depth self-supervised learning [20–22].

Previous studies initially implemented self-supervised learning using stereo camera data. For a pair of data points (I_{Left}, I_{Right}), researchers established a correlation via reprojection, a geometric transformation based on fixed camera intrinsic parameters (K) [23,24]. This method involved employing a loss function by generating a composite image from one image using the data predicted with the algorithm [25,26], further detailed in Section 2.4. Through this process, the algorithm not only interprets the disparity map of one image but also learns the fixed geometric elements of the two cameras. Subsequently, the prediction task involves generating a disparity map that includes the camera characteristics learned using only one image.

This methodology was subsequently expanded to monocular camera data, characterized by inconsistent relative relationships between adjacent images. As a result, monocular depth prediction requires the concurrent prediction of camera pose, leading the learning framework to generate synthetic data based on these poses [17,27]. The generation of geometric synthetic data has become standard in end-to-end learning frameworks, achieved through pixel sampling synthesis algorithms or differentiable bilinear interpolation. To tackle inconsistencies due to luminance errors and camera pose changes during learning, algorithms predict two or more consecutive datasets simultaneously, creating synthetic data for comparison using averages or minimum values [17]. Another approach involves comparing prediction results for three datasets in pairs and then performing cross-comparisons to enhance training stability [27].

Recent efforts have involved the creation of additional synthetic data through separate networks, such as optical flow or classification. However, this method has presented challenges in objectively assessing the monocular depth prediction capabilities of the proposed algorithms. To mitigate the influence of other prediction algorithms on monocular depth prediction performance, recent research has pivoted to generating synthetic data based on a loss function. This approach entails creating synthetic data for areas of adjacent data that are incomparable and affected by camera movement and applying a loss function to minimize interference with learning, thus prioritizing the extraction of available data [8,11,16,17].

Prior research integrated these features for self-supervised learning, complicating the evaluation of monocular depth predictions and hindering the selection of suitable algorithms for more comprehensive studies. This paper addresses these issues by incorporating normalization through the preprocessing of input data and a module for synthetic data

generation based on geometric principles within the learning framework. Consequently, it extracts available data without relying on a separate network, generates weighted data for moving objects, and modularizes it according to the decoder's structure presented in this paper.

2.2. Encoder

Dense prediction tasks in computer vision frequently utilize encoder-decoder structures to reconstruct the dimensions of input data. In this paradigm, encoders often use pre-trained classification networks, commonly referred to as backbones, for feature extraction. Monocular depth prediction follows a similar structure, where the encoder's efficiency in feature extraction and the decoder's ability to reproduce the extracted information significantly impact performance [28,29]. The focus on encoder performance stems from the challenge of retrieving information lost during the feature extraction process.

Initial research in monocular depth prediction explored depth data reconstruction using a VGG network with stacked layers serving as both encoders and decoders [3,4,6,11,23]. Despite this, limitations in detailed expression persisted, leading to subsequent studies aimed at improving depth estimation. Techniques such as learning residuals, introducing skip connections for each layer to reduce parameters and increase learning stability, and adopting the U-Net structure for a more precise depth description were investigated [15,30,31]. The combination of ResNet and U-Net structures has become increasingly prevalent in dense prediction task research [5,16,24].

To address overfitting associated with increased model size in the basic ResNet structure, recent research has concentrated on design modifications. These include integrating full connections to each layer, implementing dropout layers, batch normalization, and average pooling layers [30,32,33]. Subsequently, studies have focused on optimization strategies, encompassing efficient training, model adjustments, and training refinement [34], recently applied to learning methodologies [30,31,35,36]. The model size is characterized by its width, network depth, and input data resolution. Research has been conducted to determine the optimal hyper-parameters using the grid search algorithm [37–40]. Additionally, a method for facilitating transfer learning by segmenting tasks based on a large pre-trained model has been proposed [41,42].

This study examines algorithms that replace Recurrent Neural Network (RNN) structures with self-attention and cross-attention modules [43]. These modules, which segment images into patches similar to RNN input data, have demonstrated superior performance in computer vision tasks, outperforming traditional convolutional models in accuracy [44]. Proposals for multiple heads within the attention block cater to diverse interpretations of data relevance, while residual structures enhance learnability [45,46]. To counteract the loss of multi-head diversity near bottleneck sparsity, techniques involving re-attention and the addition of class tokens to image patches have been introduced [47–50].

This paper departs from the encoder–decoder structure prevalent in recent research, instead focusing on encoder research to augment classification performance and on decoder research that applies these encoders as pre-trained backbones. This novel approach allows for the objective evaluation of encoder efficiency by configuring the decoder to support various encoder architectures, thereby optimizing dense reconstruction performance. Consequently, the study enables the comparative selection of encoder algorithms as backbones in specialized studies on monocular depth prediction tasks, significantly streamlining the research process.

In this paper, we examine various encoder structures that are currently prominent in the field of computer vision and assess their performance through experimental comparisons. This includes the classic ResNet [30], EfficientNet2 [38], the Swin Transformer [51] from the pure self-attention series, and the hybrid convolution and self-attention MPViT [52], all implemented as switchable encoders.

2.3. Decoder

The landscape of dense reconstruction research, especially in monocular depth prediction, has seen a significant divergence into two primary areas: encoder research, focusing on feature extraction, and decoder research, dedicated to dense reconstruction [52–54]. The architectural design of the decoder is focused on enhancing information in each layer by integrating details from the encoder via long skip connections, particularly those associated with the upsampling layers from the compressed receptive field [55–57]. Prior studies aiming to improve dense reconstruction performance have investigated approaches such as integrating a pre-trained high-resolution prediction network into the long skip connection or using an additional network to harness global relationship information from the final layer of the encoder [52]. Another strategy has been the implementation of a pyramid decoder, capitalizing on the fact that each layer transmitted from the encoder to the decoder is proportionally reduced relative to the input data [58–61]. However, there remains a challenge in distinctly attributing advancements in depth prediction performance to either a newly proposed encoder, a bespoke decoder structure, or a specific learning method.

Insights derived from previous research have revealed commonalities that inform effective strategies [58–61]. First, improved performance is attained not by amalgamating global and local features for a final prediction but by initially generating predictions based on individual features from the encoder, followed by their integration. Second, convolution and self-attention series algorithms, similar to those discussed in Section 2.2, create a layer block structured to proportionally reduce in size relative to the input data, thereby addressing training challenges. These findings support the preference for a pyramid structure over a U-Net structure, facilitating the implementation of various techniques mentioned earlier. The proposed method standardizes the input data of the decoder based on features from all encoder layers, defining the layer sizes of the decoder as 1/2, 1/4, 1/8, and so on, relative to the input data [58]. A convolutional block adapts the long skip connections, including the encoder's final receptive field, to the decoder's size and channels [58]. These structural modifications aim to reduce reliance on the encoder structure through a pyramidal network, arranging data from long skip connections into identical channels of varying sizes [58–61]. Additionally, a dense network is utilized for depth prediction across different depth sizes [52–54]. Although pyramid-structured networks are hierarchical, they may impede information flow, prompting the incorporation of dense connections from one dense network layer to another to address this issue [53,62,63]. Building an independent decoder maximizes the acceptance and standardization of the final receptive field, and each layer blocks information from these encoders while also considering backpropagation for monocular depth learning. Notably, the newly introduced encoder requires no modifications, as the loss function and synthetic image generation are dependent on the structure of the decoder.

2.4. Self-Supervised Loss Functions Including Depth Consistency

In the context of monocular camera data, the lack of direct depth ground truth for comparison has necessitated the use of a 2D-plane-based loss function in previous research. This challenge is further exacerbated by the necessity to incorporate the learning of the camera pose algorithm, complicating the application of specialized loss functions focused on features like lines, surfaces, or vanishing points, as well as those designed for generative models [64–66].

The fundamental loss function in this context operates on the 2D image plane, with an enhanced version supplanting optical flow or segmentation networks. This process involves projecting the predicted depth data onto a two-dimensional plane, combined with camera pose predictions, and then calculating the difference with subsequent input data to train the depth network. Past research often refers to these image comparison loss functions as reprojection loss or reconstruction loss [17,27,67–69]. The pixel-level L1 loss is typically utilized to compare the planar image reconstructed through depth prediction with the continuous input image. However, challenges emerge when dealing with data that cannot

fully reconstruct the scene due to camera movement, leading to the adoption of SSIM loss functions for assessing pixel similarities. Additionally, a smoothness loss function, or edge-aware smoothness, is employed to mitigate discontinuities in pixel values that could hinder learning [27,69,70]. The balance of the smoothness loss is crucial, as excessive weighting may obscure vital edge information, necessitating experimental fine-tuning in previous studies.

Furthermore, features extracted from monocular camera data are not uniform, posing difficulties in directly comparing certain elements, such as the sky or scene-consistent data, amidst changes in camera pose. To address this, prior research has selectively extracted valid learning data through a comparison between the synthesized image, based on adjacent frames, and the input image used for prediction [17,69–71].

An expanded loss function relates to occlusions caused by camera pose changes or moving objects, a critical aspect of depth prediction. Traditional methods have utilized appearance loss, inspired by studies using stereo camera data [5,67]. This approach identifies data not appearing in common due to varying camera positions as occlusions, applying a low weight to foster learning on occluded data [17,72–74]. There are two variants of this extended loss function: one is image-based, removing smoothness in the reconstructed data, ideal for depth prediction; the other is depth-data-based, applying smoothness, better suited for camera pose prediction. Both versions share the benefit of addressing occlusion challenges using only input data, negating the need for additional prediction algorithms.

Finally, depending on the decoder design, the final loss function is computed at each pyramid step to train the network, taking into consideration the incremental weights of each step [27,58,63,75].

Camera pose prediction typically involves estimating the relationship between two consecutive datasets with six degrees of freedom (DoF) [76–79]. The primary 2D-plane-based loss function, as discussed earlier, acts as the central mechanism for training the camera pose prediction algorithm, while the extended loss function indirectly addresses occlusions resulting from obscured objects.

The paper focuses on evaluating the performance of camera pose prediction and occlusion processing loss functions used in studies of camera pose.

2.5. Camera Pose Estimation

In the field of unsupervised monocular depth prediction, incorporating a camera pose network is crucial for determining the relative positions of cameras [7–11]. A significant challenge in this area has been achieving accurate six degrees of freedom (DoF) camera pose predictions [76–79]. Two principal methodologies have been explored: one method inputs moving objects and depth data directly, while the other utilizes a visual odometry algorithm based on a distinct mathematical function [9].

The first approach, which aims to simultaneously predict depth and camera pose, faces difficulties in efficiently managing these dual tasks. The second approach, employing visual odometry in conjunction with a pre-trained camera pose model, offers improved performance on given data but often fails to exceed the limitations inherent in mathematical algorithms [17]. This dependence on specific mathematical algorithms also hinders the generalizability of camera pose prediction models. A subsequent study introduced a feature-matching method, which enhanced learning accuracy through an extended loss function [17,27,70]. This technique has become standard practice in modern self-supervised learning research.

Contrary to two-stage learning methods or transfer learning utilized in some previous studies, our research is dedicated to a purely self-supervised learning approach. Through experimentation, we have found that the implementation of the geometric consistency constraint loss function, as adopted in recent camera pose prediction research [17,64,65,70,72], significantly improves prediction performance. The reduction in loss through geometric consistency not only addresses the occlusion issue but also bolsters camera pose prediction.

This enables the accurate forecasting of complete camera trajectories in extended video sequences, overcoming the scale ambiguity present between successive images.

3. Switchable Encoder Self-Supervised Learning Framework

In our self-supervised learning framework, three consecutive images are used as inputs. Two sets, (FirstSet: I_{t-1}, I_t) and (SecondSet: I_t, I_{t+1}), are formed from the images, and the network is trained by calculating the intersection loss ($I_{t-1} \rightarrow I_t$, $I_t \rightarrow I_{t-1}$) within each set and the overlap loss ($I_{t-1} \rightarrow I_t$, $I_{t+1} \rightarrow I_t$) between sets. After data preprocessing and augmentation, SELF employs switchable encoders and adaptive decoders to predict depths at varying resolutions. It then learns the inverse depth (ranging from 0 to 1), which accounts for infinite distances like the sky. Thus, the input image and predicted depth data create a composite of two images and depths based on the predicted camera pose. The loss function is initially calculated by comparing these to extract uncertain data and generate weights for moving object data. The network is then trained through backpropagation with the final loss value. Framework's organization is depicted in Figure 1.

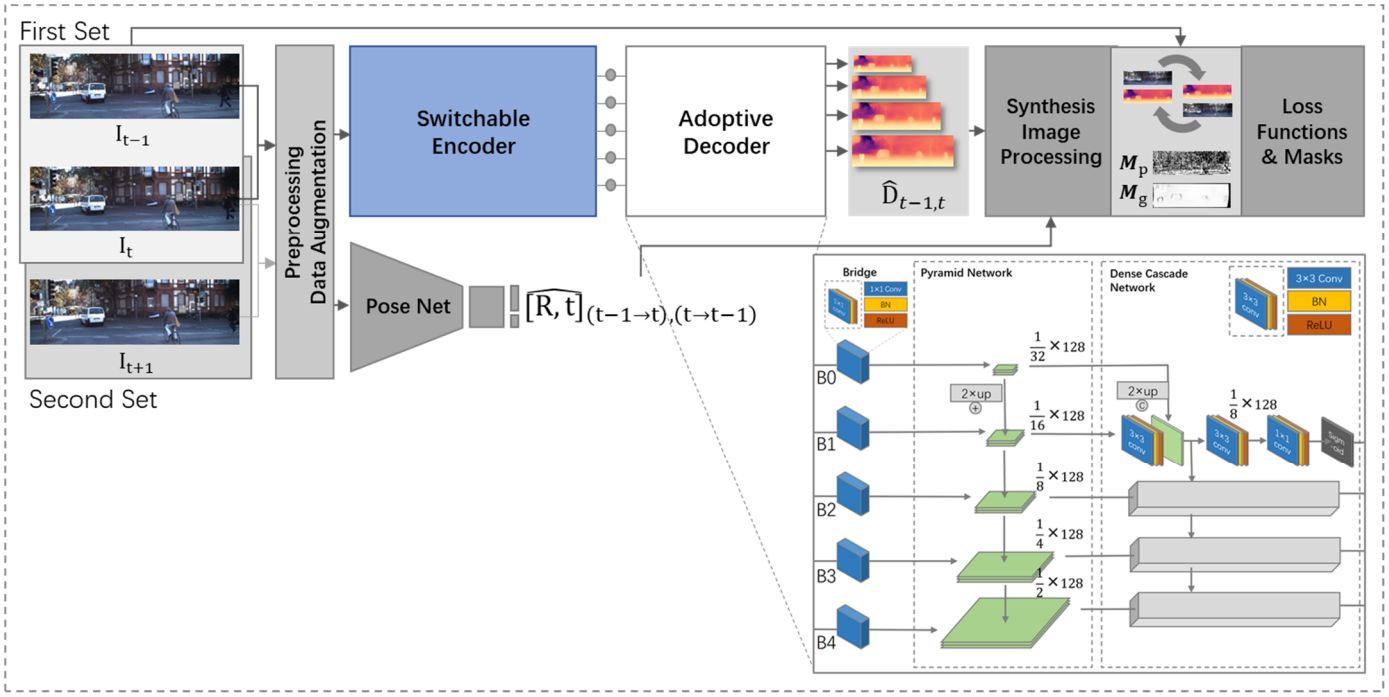


Figure 1. Switchable Encoder Self-Supervised Learning Framework. Two sets of three adjacent images are taken as inputs. They are normalized by preprocessing, and one set (I_{t-1}, I_t) is input to the depth and camera pose networks. By intersecting the two depth maps output by the depth network with the two input images, a matrix [*R*,*t*] of two relative poses ($P_{t-1\rightarrow t}$, $P_{t\rightarrow t-1}$) is extracted to generate a composite image and depth. Thus, based on the input image (I_{t-1}, I_t), predicted depth map ($\hat{D}_{t-1}$, $\hat{D}_t$), and camera pose, we generate a synthetic image ($\tilde{I}_{t-1}$, $\tilde{I}_t$)) and a synthetic depth map ($\tilde{D}_{t-1}$, $\tilde{D}_t$), and we apply the loss function. Further, a photometric loss mask (M_p) and geometric loss mask (M_g) are generated based on the loss function to finally recalculate the loss function.

3.1. Switchable Encoder

The switchable encoders used in this study are models pre-trained on ImageNet1K [80]. The structure of each encoder's connection to the adaptive decoder is detailed in Table 1. We adapted the encoders to minimize structural changes while preserving the performance of networks originally designed for classical classification tasks in computer vision, specifically for image feature extraction. The skip connection chose blocks analogous to the feature size of the standardized decoder based on the input data. The bottleneck (B0) was designed to

be size-independent. The design of long skip connections for decoders is further discussed in Section 3.2.

Table 1. Switchable Encoder. The convolutional block or attention block inside the encoder determines the block to be connected with the decoder. Swin Transformer outputs the predicted depth from the connection (B1) without any bottlenecks in the encoder structure. The numbers of training parameters are 32.5 M, 26.9 M, 26.4 M, and 25.2 M from the left of the table.

ResNet50 [30]			**EfficientNet2-S [38]**			**MPViT [52]**			**Swin Transformer [51]**		
Layer Name	**Dim**	**Bridge**	**Stage**	**Dim**	**Bridge**	**Scale**	**Dim**	**Bridge**	**Scale**	**Dim**	**Bridge**
Conv5	2048	B0	7	1280	B0	MPT Block	288	B0	/	/	B0
Conv4	1024	B1	6	256		MPT Block	288	B1	ST Block	512	B1
Conv3	512	B2	5	160	B1	MPT Block	216	B2	ST Block	256	B2
Conv2	256	B3	4	128		MPT Block	128	B3	ST Block	128	B3
Conv1	64	B4	3	64	B2	Conv-stem	64	B4	ST Block	64	B4
			2	48	B3						
			1	24	B4						
			0	24							

ResNet50 [30] consists of five convolutional blocks, with the latter four featuring a bottleneck structure containing three convolutional layers each. The output from each convolutional block is fed into the long-distance skip connections of the decoder, with the fifth convolutional block in the bottleneck being directly linked to the decoder without skipping. For EfficientNet2 [38], the architecture includes seven stage blocks: 3×3 Conv (1), Fused-MBConv (3), MBConv (3), and 1×1 Conv (1). The first, second, third, and fifth convolutional blocks are concatenated to align with the resolutions of the decoder's skip branches: $1/2$, $1/4$, $1/8$, and $1/16$, respectively, and the final $1/32$ bottleneck layer is directly connected to the decoder.

MPViT [52] is a hybrid encoder that employs two initial 3×3 convolutional layers to produce quarter-size features of the input image, followed by multiscale patch embedding and a stack of multipatch transformer blocks, which consist of a convolutional and transformer encoder for local feature extraction. The MPT blocks are connected to the skip branch of the decoder at sizes $1/4$, $1/8$, and $1/16$, with the initial convolutional block linked to the $1/2$ size branch. The final block, sized at $1/32$, is connected to the decoder's bottleneck layer. The Swin Transformer [51] replaces the convolutional component with self-attention, utilizing a 4×4 sliding window for data embedding. It does not include a convolutional block like MPViT and starts with 48 dimensions ($4 \times 4 \times 3$) at size $1/4$. A $1/2 \times 48$-dimensional Swin Transformer block is added to match the decoder's branch, as demonstrated in a previous study [54], omitting the connection of B0 to the decoder's bottleneck layer.

3.2. Adaptive Decoder

The feature extraction networks, whether convolutional-based or self-attention-based transformer families, have either pyramidal convolutional layers or attention blocks, depending on their size. Drawing inspiration from this, we organized the connections to the decoder through skip connections in a pyramid structure, as outlined in a previous study [60], and utilized a dense network [63] for depth prediction at each size. The feature data from the encoder was converted and upscaled from the preceding pyramid layer to a uniform 128-dimensional pyramid scale. For each layer, a dense cascaded network [55] was implemented for depth estimation. The bridges used were long skip connections with convolutional blocks, and the decoder's organization is depicted in Figure 2.

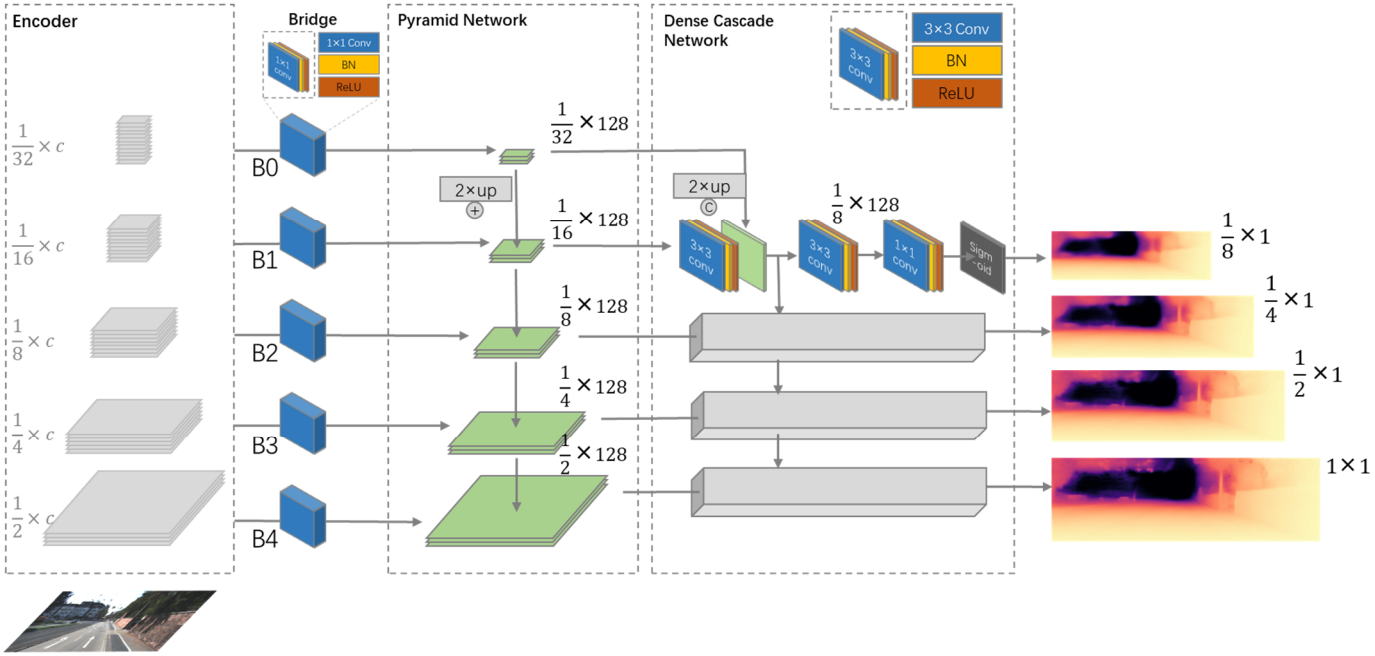


Figure 2. Adaptive Decoder. Pyramid and Dense Cascade Network. A nested pyramid decoder is constructed by maintaining the size of the encoder feature information and unifying the channels, and each pyramid stage has a dense nested connection to estimate the depth result. Since they are proportional to the size of the input image, 1/2, 1/4, 1/8, 1/16, and 1/32 of the pyramids have feature sizes of (416, 128), (208, 64), (104, 32), (52, 16), and (26, 8), respectively. c is the number of channels in the encoder and depends on the switchable encoder type. B0–B4 are bridges.

Initially, up to five bridges were contemplated for the decoder to directly receive information from the encoder. These skip connections were structured based on the feature extraction size of the encoder, as indicated in a previous study [60]. We incorporated five pyramid layers with sizes of 1/2, 1/4, 1/8, and 1/16, proportional to the input data (H,W), and an additional layer of size 1/32 corresponding to the final bottleneck layer of the encoder. To harmonize the feature size for each layer of the switchable encoder, the same convolution block previously described was employed in each long skip connection layer of the bridge. This densely connected decoder structure, similar to that in prior work [54], was expanded to include five skip connections and incorporated a 3D loss function and an extended mask technique for facilitating the interchange of different encoders.

The upsampling process followed the top-down pathway method found in earlier research [60]. Beginning with the bottleneck layer, the feature map of each pyramid layer was upsampled twice, equalizing the number of channels in each layer. Unlike the neighbor interpolation used in previous studies, we adopted a bilinear interpolation method for scaling. However, the upscaling process in pyramid networks, typically connected by a sum, can hinder information flow [55,63]. To ameliorate this, we enhanced information flow using dense cascade connections that directly concatenated all preceding layers to the subsequent layer. As expressed in Equation (1), X_0, $X_1, \ldots,$ X_{i-1} represent each layer, and H_l has a convolutional block that is adapted from $[BN - ReLU - Conv(3 \times 3)]$ to $[Conv(3 \times 3) - BN - ReLU]$ to align with the upsampling of the decoder [25,31]. The aggregated final feature map then proceeded through the convolutional block and sigmoid layer—the chosen activation function—to produce a single-channel output, representing the predicted depth result for each pyramid layer.

In summary, feature maps received through long skip connections from the encoder were vertically upscaled using a pyramid network in the adaptive decoder. This network integrated the cumulative feature map of the previous layer with dense cascade connections to predict the depth map according to pyramid size. For training purposes, we computed

the loss function at each pyramid layer, obtaining a proportionally weighted summed loss function.

$$X_i = H_i([X_0, X_1, \ldots, X_{i-1}]) \tag{1}$$

3.3. Depth Consistency Guaranteed Loss Function

In self-supervised learning, depth prediction from a single image must be learned without labeling depth ground truth, deriving the solution from a sequence of adjacent images. An image, (I_{t-1}), is fed into the depth prediction network, which generates the predicted depth data ($\widetilde{D}_{t-1}$). The image (I_{t-1}) is then synthesized using the predicted depth and camera pose to facilitate learning through comparisons with neighboring images. The camera pose network synthesizes $\widetilde{I}_t$ from the relative camera pose matrix [*R*,*t*] between two images, considering the camera's internal parameters (K), and compares it with the neighboring images. The specific formulation is provided in Equation (2).

$$\widetilde{I}_{t-1\to t} = I_{t-1}\left(Reproject\left(\hat{D}_t, \hat{P}_{t\to t-1}, K\right)\right) \tag{2}$$

There are two main challenges in generating the synthesized image $\widetilde{I}_t$. Firstly, the data values transformed due to changes in camera pose are continuous and may not correspond to integer values. To address this, we employed differentiable bilinear interpolation, blending these values based on the I_{t-1} we aimed to compare, augmented by the distance value to neighboring pixels. Secondly, changes in the camera pose can cause pixel positions in the image to overlap or disappear, potentially altering the overall image size. To counter this, we considered image transformation $\left(\widetilde{x}, \widetilde{y}\right) = T(x, y)$ to inverse warping (T^{-1}: inverse warping) to preserve the original size of the intended image. In our research, the camera pose matrix [*R*,*t*] consistently possessed an inverse function, maintaining a relationship between the neighboring images.

In our self-supervised learning framework, the loss is determined through comparison with the synthesized image, and it is formulated as follows: V (validate) represents the set $\widetilde{I}_t$ of comparable pixels successfully placed on the I_t plane based on the depth estimate, and L_1 is the loss of pixel $\widetilde{I}_t(p)$ in the synthesized image compared with pixel $I_t(p)$ in the neighboring image. However, this formula does not account for real-world factors like variations in light intensity. Even in consecutive images, losses due to light intensity variations can impair learning efficiency. Therefore, we calculated the luminance, contrast, and structure of the two images to assess their similarity. Luminance was compared based on the average brightness of the images, μ_x, and was calculated using $l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$. If μ_x, μ_y are similar, this value is close to 1, and the larger its difference, the more it tends towards 0. Further, C_1 prevents the denominator from approaching zero, $C_1 = (K_1 L)^2$, where K_1 denotes a general constant (usually 0.01) and L indicates the range of pixel values (255). Therefore, we applied $C_1 = (0.01x255)^2 = 6.5025$. Contrast uses σ_x, and the calculation of $c(x, y)$ is the same as luminance. However, $C_2 = (K_2 L)^2$, and K_2 was 0.03; thus, we applied 58.5225. Structure normalizes Image $- \mu_x/\sigma_x$ with μ_x denoting the mean and σ_x representing the standard deviation to obtain the correlation between the two images. Finally, structural similarity (SSIM) was applied by deriving Equation (4) through the product of luminance, contrast, and structure [81].

$$L_{Photometric} = \frac{1}{|V|}\sum_{p\in V}\left\| I_t(p) - \widetilde{I}_t(p)\right\|_1 \tag{3}$$

$$\mathrm{SSIM(x, y)} = \frac{\left(2\mu_{\,x}\mu_y + C_1\right)\left(2\sigma_{\,xy} + C_2\right)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)} \tag{4}$$

Using these two loss functions, we applied the luminous-intensity loss function, as expressed in Equation (5). Here, α was set to 0.15.

$$L_{Photometric-ssim} = \frac{1}{|V|}\sum_{p\in V}((1-\alpha)\left\|I_t(p) - \widetilde{I}_t(p)\right\|_1 + \frac{\alpha}{2}(1 - SSIM(I_t(p), \widetilde{I}_t(p))) \quad (5)$$

This loss function makes uniform comparisons in regions with low texture or similar luminosity challenging, consequently diminishing the efficiency of learning depth and camera pose. To create smooth data, as accomplished in previous studies [5,7,8,17,70], we introduced an edge-aware smoothness loss function, as depicted in Equation (6), prior to normalization.

$$L_{Edge-Aware\ Smoothness} = |\partial_x d_t^*|e^{-|\partial_x I_t|} + |\partial_y d_t^*|e^{-|\partial_y I_t|} \quad (6)$$

In our self-supervised learning framework, we combined odometry, long-range camera poses, and depth, leveraging the proposed switchable encoder. Consequently, we introduced an additional loss function to ensure the consistency of the generated depth map and facilitate the learning of the camera pose, as demonstrated in previous studies [17,72–74]. The three input images were divided into two sets. For each set, depth was estimated by intersecting two images (I_{t-1}, I_t), one of which (I_{t-1}) was computed as a depth map of the other image (I_t) along with the camera pose and was interpolated. Geometric constraints were subsequently applied to both the synthesized and predicted depth maps ($\widetilde{D}_{t-1\to t}$, $\hat{D}_t$). This unsupervised learning framework for monocular depth prediction ensures depth consistency across the entire input sequence, given that the input images are nested in a persisting sequence: ((I_{t-1}, I_t, I_{t+1}), (I_t, I_{t+1}, I_{t+2})). The comparative formulas are articulated in Equation (6), where $\widetilde{D}_{t-1\to t}$ represents the depth map projected onto the I_t image plane via the predicted depth map $\hat{D}_{t-1}$ and the predicted relative camera pose between the two images ($\hat{P}_{t-1\to t}$). We then applied geometric constraints based on valid data (Equation (5)), as outlined in Equation (8). The validity of images is discussed subsequent to the total loss calculation. In other studies [7,43], three images were used to learn implicit scale consistency constraints by comparing the depth of the central image with the other images on two separate occasions. For performance comparisons, a relative evaluation of depth maps generated from a single image is beneficial. Our proposed monocular self-supervised learning framework ensures depth scale consistency across an image sequence through a direct loss function and enhances long-range odometry performance.

$$D_{diff}(p) = \frac{\left|\widetilde{D}_{t-1\to t}(p) - \hat{D}_t(p)\right|}{\widetilde{D}_{t-1\to t}(p) + \hat{D}_t(p)} \quad (7)$$

$$L_{Geometry} = \frac{1}{|V|}\sum_{p\in V} D_{diff}(p) \quad (8)$$

The loss functions applied to the learning framework are defined as in Equation (7). The weights of each loss function are W_1, W_2 and W_3, respectively, and were set to 1.0, 0.1, and 0.5, respectively.

$$L_{TOTAL} = W_1 L_{Photometric} + W_2 L_{Edge-Aware\ Smoothness} + W_3 L_{Geometry} \quad (9)$$

In accordance with the decoder structure detailed in Section 3.2, we applied a scale-specific loss function to the feature map to enhance both the depth reconstruction capabilities of the decoder and the training efficiency of the camera pose network. The final loss function was determined based on the weights assigned to each pyramid level. We calculated the final loss function from the final depth results of the four pyramid layers, reducing the weight to 0.1.

To address challenges in learning from consecutive images within monocular depth estimation tasks, we utilized masks based on the time difference between image acquisitions. Firstly, some pixels remain stationary despite changes in camera pose, such as objects moving at a consistent speed in the same direction as the camera. Secondly, disparities in regions and objects with different depths arise due to the acquisition time difference between the images. For the first issue, we generated a mask based on a photometric loss function, as implemented in previous studies [5,17,27], to isolate invalid data, as depicted in Equation (10).

$$M_{t-1}(p) = \begin{cases} 1 \; if \; \left\| I_{t-1}(p) - \widetilde{I}_{t-1}(p) \right\|_1 < \| I_{t-1}(p) - I_t(p) \|_1 \\ 0 \; otherwise \end{cases} \tag{10}$$

where M_{t-1} denotes a binary mask and $\widetilde{I}_{t-1}(p)$ represents the result of re-projecting the image of I_t through the predicted camera pose ($\hat{P}_{t \to t-1}$). The mask $M_{t-1} = 0$ yields pixels that are valid for training and applied to the loss function.

Second, the depth mismatch between two images, stemming from the geometric error described earlier, is presented in Equation (10). Studies [17] indicate that dynamic objects, occlusions, and data that challenge the definition of relationships between two images can increase the D_{diff} error, indicative of a breach in geometric consistency. D_{diff}, which has a range of 0–1, is weighted, as expressed in Equation (10), forming a mask that assigns lower weights to inconsistent data and higher weights to consistent data. This mask is then used to compute the conventional luminosity loss function. Therefore, upon applying the proposed mask, we derived the final photometric loss function, as outlined in Equation (12).

$$M_w = 1 - D_{diff} \tag{11}$$

$$L_{Photometric}^{Mask} = \frac{1}{|V|} \sum_{p \in V} (M_w(p) \cdot L_{Photometric}(p)) \tag{12}$$

The final loss function was computed by applying a weight (M_w) for geometric consistency to each data point. This weight was derived by extracting valid data (validate) through a photometric mask. We employed masks to alleviate issues caused by moving objects and occlusions, ensuring that regions with inaccurate predictions were assigned lower weights during backpropagation.

3.4. Camera Pose Estimation Network

The camera pose network processes two consecutive images (I_t, I_{t+1}) as inputs and predicts six degrees of freedom (6DoF) through feature matching, aligning with methodologies used in prior studies [8,16,17,27]. It outputs a transformation matrix that captures the rotation and translation of the camera between the images.

Learned concurrently with depth prediction, the camera pose network merges the two images into six channels and calculates the difference value, representing the 6DoF. Utilizing this data, the learning framework produces synthetic images for depth prediction, and the loss function evaluates the loss by downweighting valid data while considering moving objects. In cases where moving objects are prevalent in the dataset, robust features are extracted, which can impede accurate camera pose estimation. To counteract this, the effectiveness of the camera pose network can be augmented by integrating a geometric loss function.

4. Experiment

For comparative purposes, experiments were conducted against self-supervised learning frameworks from earlier studies [17,27]. We adopted the same encoder [30] as a baseline

in our proposed learning framework, as utilized in [17], which offered comparisons for odometry and analyzed the accuracy improvements in depth and odometry.

In evaluating depth, we adhered to established metrics from previous studies [6,11,17,23]. These metrics include the mean absolute relative error (AbsRel), mean log10 error (Log10), root mean squared error (RMS), root mean squared log error (RMSlog), and accuracy under various thresholds ($\delta_i < 1.25^i$, $i = 1, 2, 3$).

Note that in monocular self-supervised learning, the absolute scale is not directly recoverable. Consistent with prior research in this field, we addressed this limitation by scaling the predicted depth maps using a scalar that adjusts the median of the predictions to match that of the ground truth. Additionally, to maintain applicability to specific datasets like KITTI, we limited the predicted depths to a maximum of 80 m/10 m in the respective datasets.

For a balanced comparison, our visual odometry evaluation was benchmarked against both our learning framework and the findings of previous research [17]. The evaluation includes standard metrics such as translational (t_{err}) and rotational errors (r_{err}), calculated as averages over the entire sequence, alongside the absolute trajectory error (ATE). This method conforms to the established practices for visual odometry evaluation in the research community.

To showcase the efficacy of our monocular depth prediction self-supervised learning framework with interchangeable encoders, we conducted a comparative analysis using the monocular depth self-supervised learning framework from a recent study [27]. Our experiments included the ResNet50 encoder [30] from the convolutional family and the EfficientNet2-S, the smallest model of the EfficientNet2 encoders [38], optimized for similar performance with fewer parameters. In the transformer family, we tested the hybrid MPViT encoder [52] and the hierarchical Swin transformer encoder [51], which replaces all convolutions with self-attention, to assess the impact of encoder performance on enhancing monocular depth estimation in our proposed learning framework. We evaluated the performance of our learning framework by comparing the depth prediction capabilities of various switchable encoders, reconstructed using the decoder with the feature information of the replaced encoder, and assessing the consistency of the predicted depth.

The data for training and evaluation of our constructed self-supervised learning framework were sourced from the KITTI dataset [82]. The KITTI dataset assembles and processes data gathered using various sensors, including video footage from a car's journey. It incorporates GPS, LiDAR, monochrome, and color cameras, with the depth information aligned with that from the LiDAR. The KITTI dataset is a widely used benchmark dataset in computer vision, particularly for tasks related to autonomous driving and scene understanding. It includes a diverse set of real-world images collected from a moving platform in urban environments. In our experiments, we utilized a resolution of 832 × 256 pixels. The depth and odometry measurements were divided into training, validation, and testing categories, as detailed in Table 2.

Table 2. KITTI data used for depth and odometry prediction, evaluation, and testing.

	Depth Estimation	**Odometry Estimation**
Training	42,440	Seq. 00–07
Validation	2266	Seq. 08
Test	697	Seq. 09–10

4.1. Comparing the Performance of Self-Paced Learning Frameworks

In this experiment, we evaluated the performance of our proposed learning framework against existing self-supervised learning methods [17]. We utilized the same encoder [30], but with an enhanced decoder, to compare the depth and odometry performances.

For depth performance comparison, the same ResNet50 [38] encoder was employed, reducing the original 34.6 M parameters in the depth prediction network to 32.5 M. This

reduction resulted in an overall improvement in depth prediction performance to 0.113, as detailed in Table 2. From the evaluation results, we observed enhancements in both overall depth prediction and accuracy in critical corners.

The continuous odometry performance of camera poses is compared in Tables 3 and 4. Using our proposed learning framework, there was a noticeable improvement in the overall performance of the camera poses. This demonstrates that even though the camera pose network was not directly linked to the depth network, the depth performance predicted with the learning framework, utilizing the same loss function, influenced the effectiveness of the feature-matching-based camera pose network. The odometry results are illustrated in Figure 3. When compared to our previous study [17], the proposed learning framework managed to reduce the number of training parameters to 93% using the same encoder. Furthermore, it showed an overall enhancement in both depth and camera pose prediction tasks.

Table 3. Results of relative depth estimation and absolute depth estimation using proposed learning network and sparse LiDAR-based depth GT.

	Abs Rel	**Sq Rel**	**RMSE**	**RMSE Log**	**Accuracy under a Threshold (δ)**		
					δ**1**	δ**2**	δ**3**
SC-SFM [17]	0.114	0.813	**4.706**	0.191	**0.873**	**0.960**	0.981
Our (ResNet50)	**0.113**	**0.793**	4.724	**0.187**	0.869	0.959	**0.983**

The bolder text represents a comparative advantage.

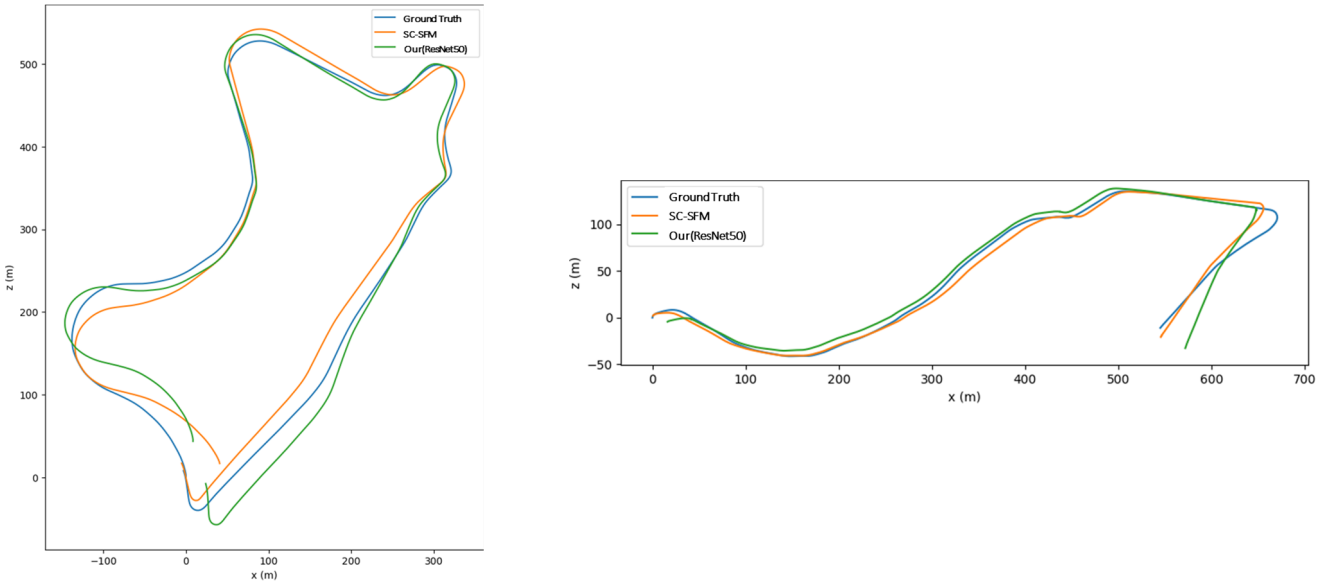
Table 4. Overall performance in predicting camera movement and rotation. Seq. 09's evaluation shows a slight decrease in performance for camera movement prediction but an improvement in rotation and difference from ground truth. Seq. 10 shows performance improvements in both relative translation and rotation between consecutive images.

	Seq. 09					**Seq. 10**				
	Translational Error	**Rotational Error**	**ATE**	**RPE (m)**	**RPE (deg)**	**Translational Error**	**Rotational Error**	**ATE**	**RPE (m)**	**RPE (deg)**
SC-SfM [17]	**7.31**	3.05	23.55	0.11	**0.10**	7.79	4.90	**12.00**	**0.08**	0.11
Our(ResNet50)	8.44	**2.49**	**20.93**	**0.09**	0.11	**6.35**	**4.78**	15.47	0.10	0.11

The bolder text represents a comparative advantage.

4.2. Comparing Depth Estimation Performance of Switchable Encoders

In our research, we evaluated the depth prediction performance by integrating newly proposed encoder algorithms into our learning framework. The experiments confirmed that the performance of the encoder significantly influences depth prediction performance and that our proposed learning framework maintained consistency across various encoder algorithms. The classification performance of each encoder used in the experiment is presented in Table 5. We chose the smallest model from each encoder algorithm to emphasize performance improvements. To ensure equitable comparisons, each network was trained for the same duration of 200 epochs using an A6000 1 GPU, with only the encoder being interchanged. The input image size was set at (832 $\times$ 256), and the depth estimation results are compiled in Table 6.



(**a**) (**b**)

Figure 3. Rotation results. Previous work, SC-SFM [17] over-compensated for rotation, while our framework under-compensated. In Seq. 10, despite the overall performance improvement, an under correction occurred in the last trajectory, which affected the absolute evaluation as shown in Table 3. (**a**) Seq. 09 odometry result; (**b**) Seq. 10 odometry result.

Table 5. Switchable Encoder Performance for Classification Task. The switchable encoder was pre-trained on the ImageNet dataset [80]. Accuracy represents the proportion of correctly classified instances (both positive and negative) out of the total instances.

Switchable Encoder	Training Image Size (ImageNet 1 K)	Classification Accuracy (%)
ResNet 50	224 × 224	79.26
EfficientNetV2-S	128–300 (progressive training)	83.9
MPViT-S	224 × 224	83.0
Swin-S	224 × 224	83.0

Table 6. Depth estimation result. Comparison of depth estimation performance of the encoder replacement based on the input image (832 × 256).

Encoders	Abs Rel	Sq Rel	RMSE	RMSE Log	Accuracy under a Threshold (δ)		
					δ1	δ2	δ3
Monodepth2 [27]	0.115	0.882	4.701	0.190	0.879	0.961	0.982
Our (ResNet) [30]	0.113	0.793	4.724	0.187	0.869	0.959	0.983
Our (EfficientNet2) [38]	0.111	0.837	4.703	0.185	0.876	0.961	0.983
Our (MPViT) [52]	**0.109**	0.848	4.665	**0.183**	**0.881**	0.962	0.982
Our (Swin) [51]	**0.109**	**0.765**	**4.664**	**0.183**	0.878	**0.963**	**0.984**

The bolder text represents a comparative advantage.

The objective of this study was to validate the consistency of the learning framework by examining whether different encoders sustain their performance in the depth prediction task. This comparative analysis was based on [27], which recently served as a benchmark learning framework to assess the depth performance of each encoder.

The depth prediction results are displayed in Table 6. In comparison to the ResNet [30] used in prior experiments, EfficientNet V2 [38] demonstrated superior performance with only about 73% of the parameters. Additionally, the hybrid method [52], which em-

ploys convolution in both feature extraction and self-attention blocks, and the Swin Transformer [51], which utilizes self-attention, showed enhanced depth prediction performance.

The experiments revealed that the feature extraction performance of the encoder also impacts the regression task. Visualizations were conducted to compare the characteristics of the encoder transmitted to the adaptive decoder via the bridge, as illustrated in Figure 4. Post-learning, we present the receptive field differences for each layer, corresponding to the specific algorithmic features. These differences serve as input data for the adaptive decoder we designed. A noticeable pattern emerges towards the final receptive field: the convolutional series displays a collection of local features, whereas the self-attention series focuses on global features. Further experiments, including visualizations of the encoder's layers, final depth prediction results, and depth predictions for each layer of the decoder, facilitate a thorough comparison of the proposed decoder's final and layer-by-layer performance.

Additionally, we examined changes in data concentration within each layer by visualizing the results of camera pose learning. Although direct evidence was not provided as input data, Figure 5 shows that, based on the loss function, the emphasis shifts to peripheral features of the input data, as opposed to the center, where movement is less pronounced.

Interestingly, we noted that the depth prediction results varied based on the family characteristics of the encoders replaced within our learning framework. Figures 6 and 7 display the depth maps, photometric error masks, and weighting masks for moving objects predicted with each encoder. The performance outcomes in Table 4 confirmed that encoders from the convolutional family excelled at extracting local features, such as static and dynamic objects, while the pure self-attention encoders were superior at capturing global feature relationships rather than detailed aspects.

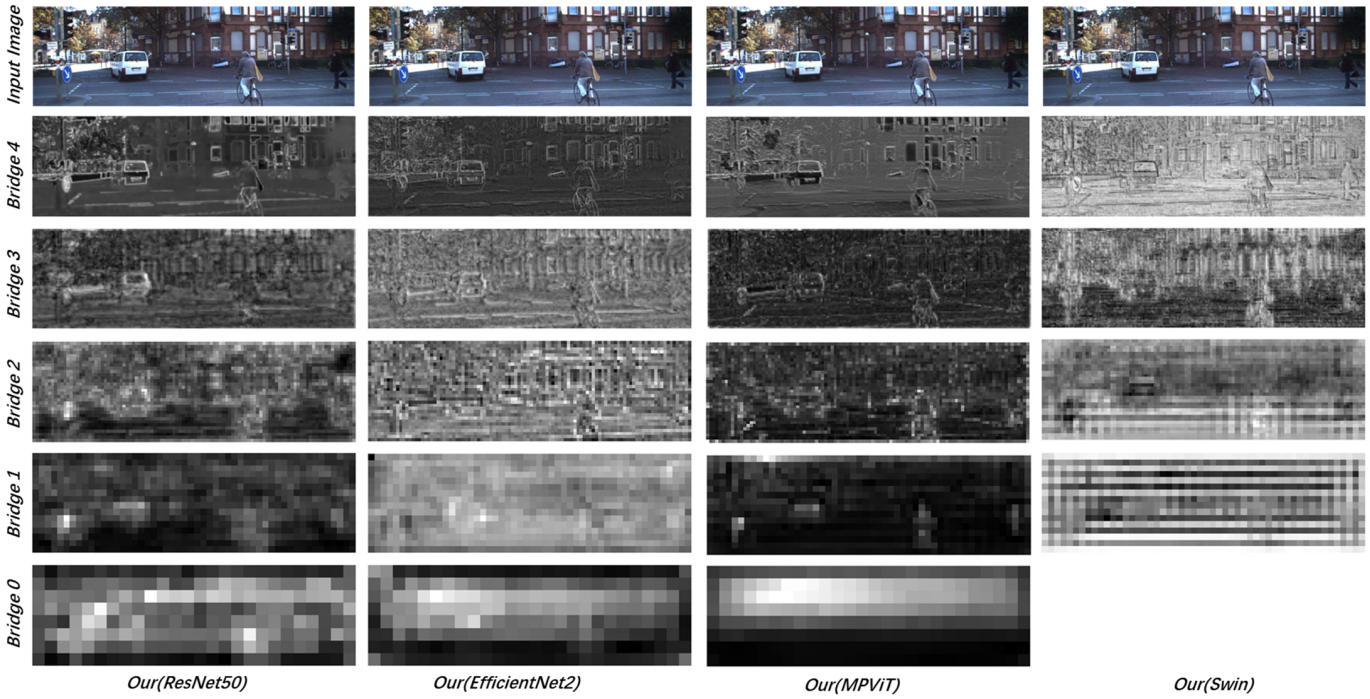


Figure 4. Bridge Visualization. The bridge used as the input of the adaptive decoder delivers feature information from each layer of the encoder. To avoid dependence on the encoder structure, the bridge converts it to the standardized size of the decoder.

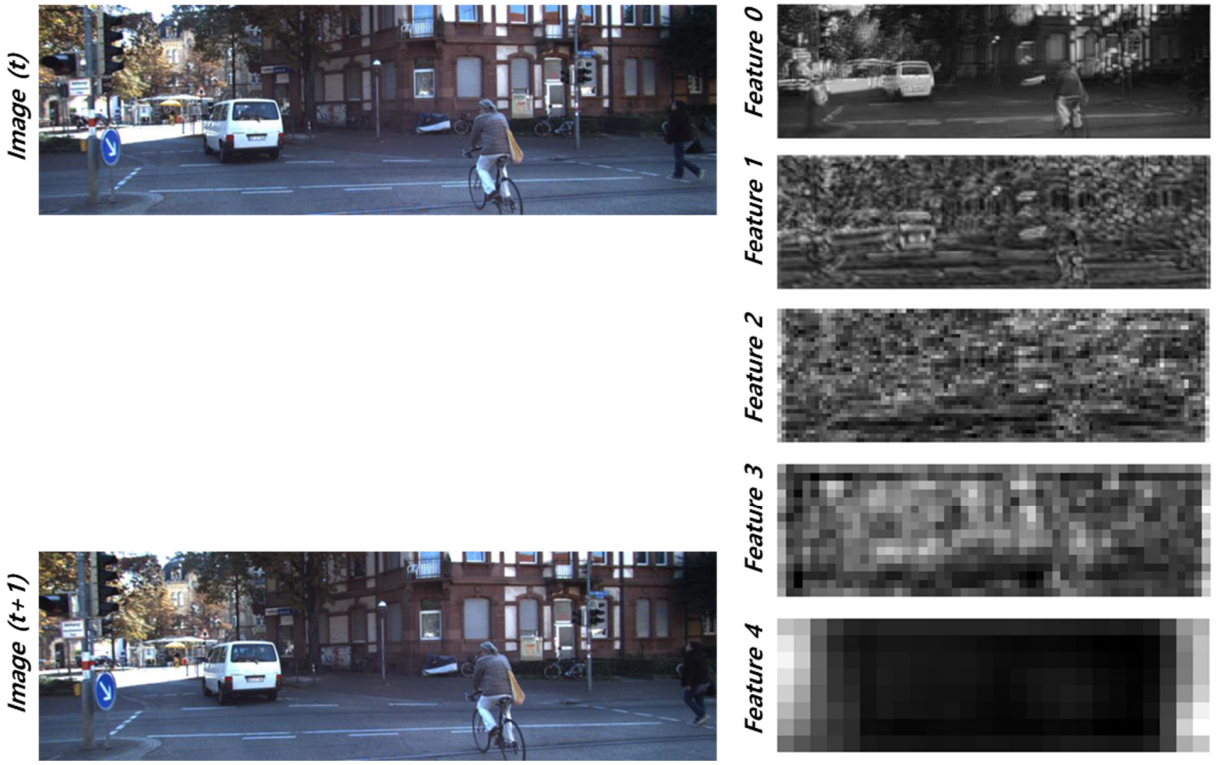


Figure 5. Visualization of Camera Pose Network. The visualization results for each layer of the camera pose network trained through a depth prediction loss function by receiving two consecutive inputs.

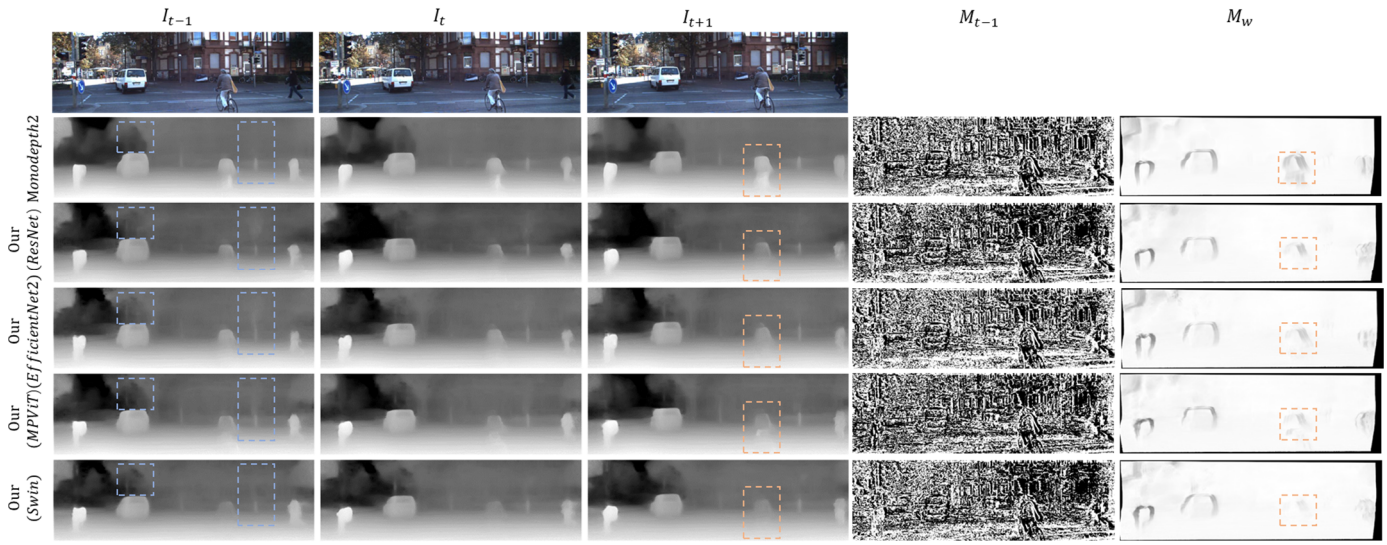


Figure 6. Depth map, validation synthetic data (M_{t-1}), and geometry synthetic data (M_w). Blue-dotted squares show a comparison of local features in the image. The convolutional series had a more detailed description of the local features than the transform series. The red dashed squares show a comparison for a moving bicycle, showing that the convolutional series focuses on features in the depth mask. Previous research MonoDeepth2 [27] applies a 2D-based loss function and the weight of the M_w is constant.

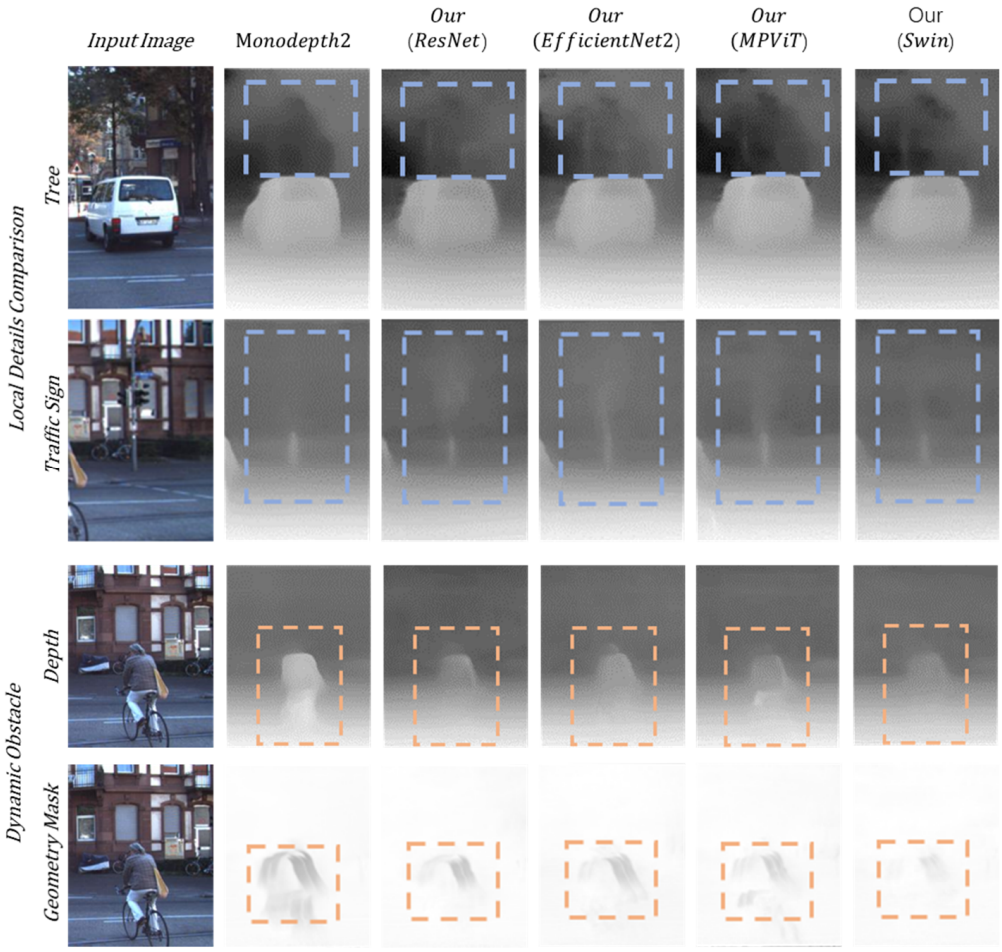


Figure 7. Highlight details of Figure 6.

4.3. Information Reorganization Results

The present experiments investigated whether the high-dimensional feature information from the bottleneck of the encoder and the detailed low-dimensional feature information from the input data were both retained and effectively reproduced using the decoder. The findings indicated that our learning framework consistently reconstructed information for depth estimation, irrespective of the encoder used. Given the challenges of evaluation with sparse depth data like LiDAR, we opted to compare the results by projecting the depth map onto a 2D plane. Figures 8 and 9 showcase the depth estimation outputs from the four encoders, each fed with high-dimensional feature information through the bottleneck connection B1 to estimate depth. We then compared the final depth map, as predicted by the pyramid and dense cascade connections, with the feature information conveyed through the last skip connection (B5). This comparison aimed to ascertain whether the depth map predicted at the bottleneck was preserved across each layer of the decoder and whether the local details, not captured by the high-dimensional information, were accurately integrated to reproduce detailed depth information at the correct locations.

In our experiments, we observed that our learning framework consistently reproduced all necessary information for depth prediction in the decoder based on the high- and low-dimensional feature information from the encoder, even when the encoder was replaced. This demonstrated that our proposed decoder could adapt to depth estimation with minimal degradation, even when a new encoder was introduced.

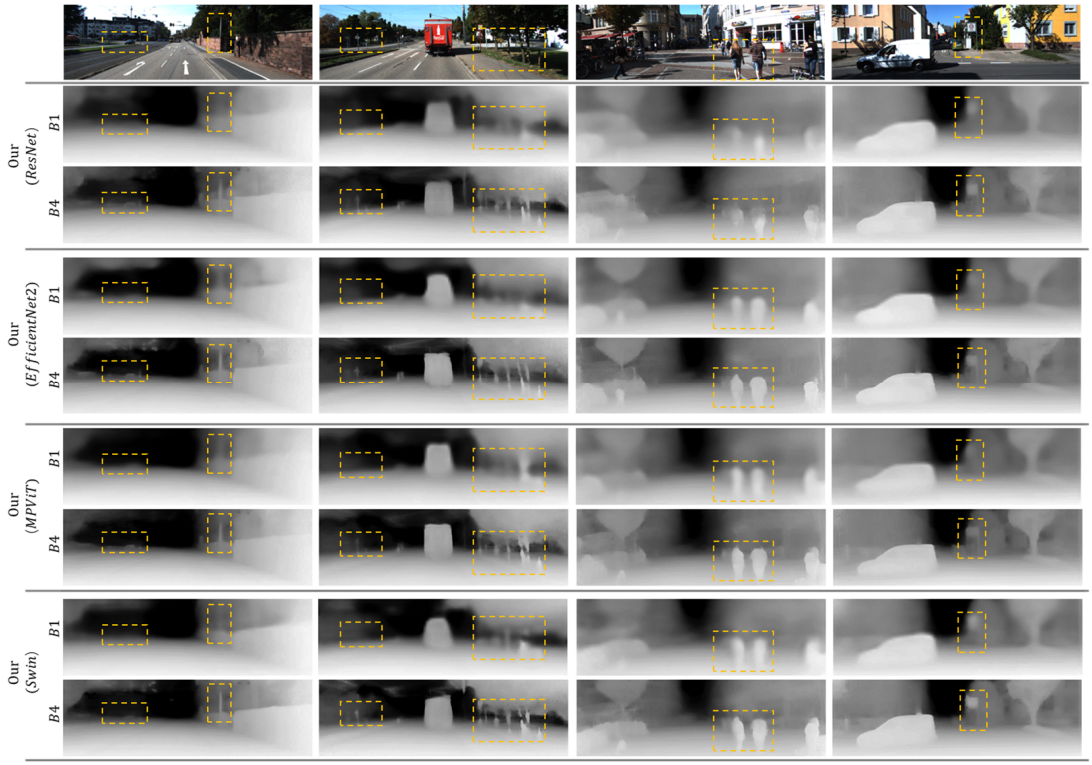


Figure 8. B1 is the result of the top depth map of the pyramid, and B4 is the final depth map predicted by combining all feature information before the pyramid. The yellow dotted boxes represent the details missed by the higher-dimensional information at the right locations in the lower dimensions.

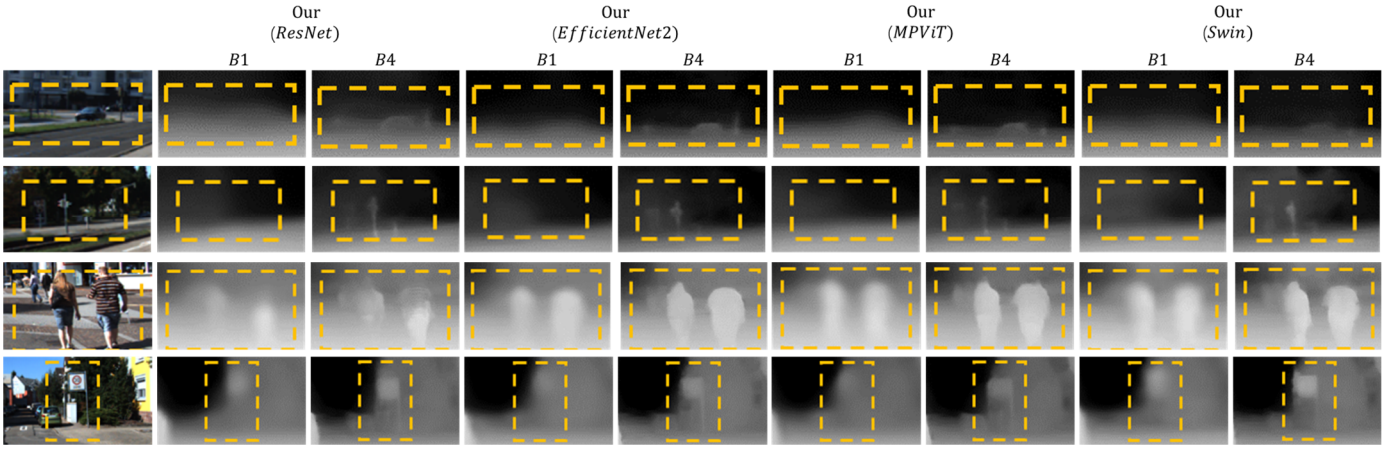


Figure 9. Highlight details of Figure 8.

4.4. Depth Consistency Results

Our final experimental validation focused on the depth consistency of the learning framework. We compared our results with those of a previous study [27] that trained a learning framework without explicit constraints on geometric consistency. The conceptual basis of our experiment was as follows:

- If a significant portion of the data in a series of images featured objects like buildings, the distribution of depth values would vary with changes in the camera pose;
- Conversely, if the images predominantly depicted roads, the distribution of depth values would be similar and exhibit depth consistency.

Figure 10 illustrates the depth values extracted from three consecutive images, compared using distribution plots. We analyzed the depth distribution predicted with the Swin transformer encoder, known for its effective depth prediction. We then contrasted these distributions with those from previous studies [27] that did not directly implement a depth coherence loss function. With alterations in the camera pose, particularly when moving closer or farther from data like buildings that occupied a large portion of the image, the distribution of depth values varied. Our learning framework, however, maintained depth value consistency even with different encoders. This ensured uniform depth estimation across a sequence of data, such as videos, proving beneficial for global point-cloud construction.

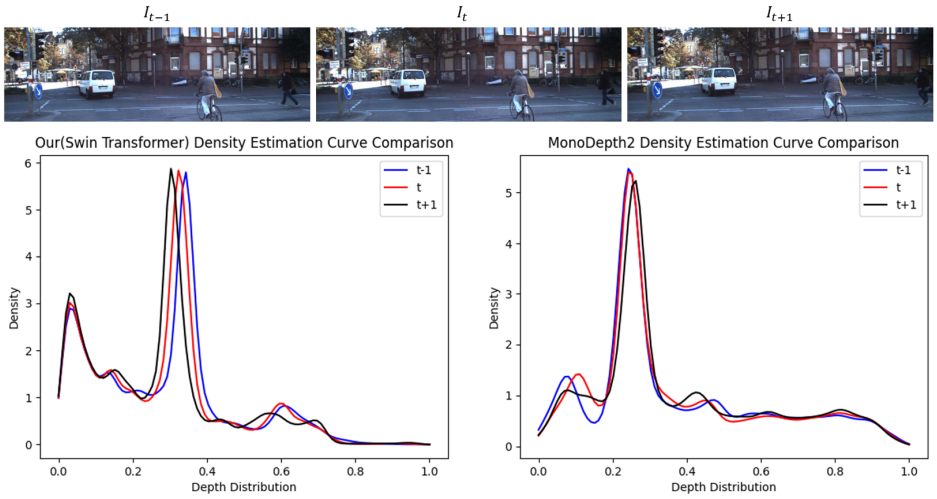


Figure 10. With changes in the pose of the camera in successive images, the distribution of depth values changes owing to the buildings that make up the majority of the data. The graph on the left shows the distribution of depth values trained with our learning framework, while the graph on the right shows the distribution of depth values obtained in a previous study [27].

Figure 11 presents a comparison of depth consistency in images with similar depth distributions. The benchmark for this comparison was the GT depth data from LiDAR. We contrasted the depth value distribution between our self-supervised learning framework, which incorporates a direct geometric loss function, and the findings from the previous study [27].

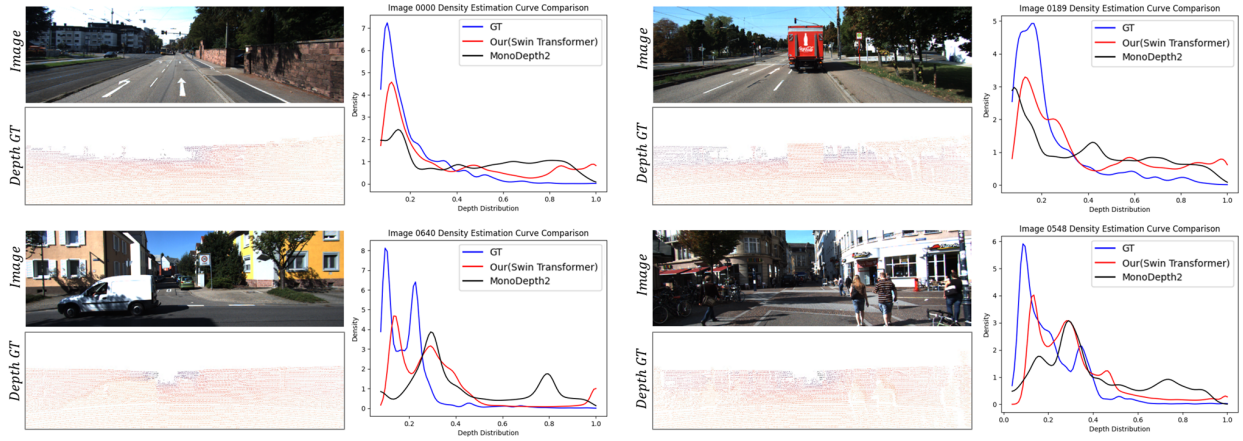


Figure 11. Four different images with similar depth value distributions were compared. Regardless of the performance of the encoder, the depth value distribution of the network trained through our proposed learning framework showed scale consistency and was more similar to the pattern of the ground truth than in past research [27].

Depth consistency not only extends the concept of depth accuracy but is also vitally linked to future research endeavors in global point-cloud construction, odometry, and object distance measurements. Ensuring uniformity in the predicted depth scale, in conjunction with enhancing the performance of the encoder, is crucial for the effective application of our learning framework.

Through our experiments, we verified that SELF is capable of representing dense depth and preserving the scale of the depth with a switchable encoder. However, the basic loss function alone was insufficient to significantly improve the performance of the camera pose network. Therefore, further research is necessary to develop a precise 3D global point cloud, leveraging advancements in the self-supervised learning framework.

5. Conclusions

In conclusion, our exploration delved into the self-supervised learning aspects within the realm of existing monocular depth prediction research, culminating in the introduction of SELF—an end-to-end learning framework featuring an integrated decoder. The ease with which the newly proposed artificial intelligence algorithm can function as an encoder for dense depth prediction tasks, coupled with its compatibility with pre-existing models, underscores the flexibility of our approach. Notably, the absence of any requirement to adjust self-supervised learning elements further streamlines the application of our learning framework. By providing a platform for the objective evaluation of encoder performance, our framework contributes to the advancement of depth-based studies.

Our approach not only achieves a 7% reduction in learning parameters compared to the learning frameworks used in recent studies but also enhances depth and pose prediction performance using the same encoder. Directly applying the newly proposed encoder to our learning framework results in a 23% reduction in learning parameters and a 5% improvement in performance, showcasing the framework's versatility in accommodating various encoders for depth prediction while maintaining a consistent depth scale.

In this study, we conducted experiments using the KITTI dataset under the same conditions as previous studies, yielding comparative results in terms of learning and evaluation. However, despite diverse scenes with buildings and moving objects, our learned model demonstrates limitations in generalization. The camera model used for data acquisition and the refined data tailored to artificial intelligence algorithms are suitable for evaluating performance but pose constraints on accessing the foundational model, representing the ultimate aim of self-supervised learning.

In light of this study, we contemplate broadening our exploration within the self-supervised learning framework. Despite the absence of ground truth, we envision extending preprocessing to re-synthesize input data based on the KITTI camera model, facilitating the incorporation of a more diverse range of training data. Learning outcomes, even without ground truth, become feasible through leveraging the KITTI dataset. By extending research in various domains and utilizing the artificial intelligence algorithm validated in depth prediction evaluations as the backbone through the self-supervised learning framework proposed in this paper, we aim to streamline the research period, reduce learning time, and enable objective evaluations.

Author Contributions: Conceptualization, J.K.; Funding acquisition, K.C.; Investigation, J.K. and R.G.; Methodology, J.K.; Project administration, K.C.; Software, J.K. and R.G.; Supervision, K.C.; Validation, J.K. and R.G.; Visualization, J.K. and R.G.; Writing—original draft, J.K.; Writing—review and editing, R.G., J.P., J.Y. and K.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Korea Institute of Police Technology (KIPoT) grant funded by the Korean government (KNPA) (No.092021D75000000, AI driving ability test standardization and evaluation process development) and the Institute of Information & Communications Technology Planning & Evaluation (IITP) under the Artificial Intelligence Convergence Innovation Human Resources Development (IITP-2023-RS-2023-00254592) grant funded by the Korean government (MSIT),

and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2022R1A2C2006864).

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found here: https://www.cvlibs.net/datasets/kitti/ (accessed on 1 December 2021) [82].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Eigen, D.; Puhrsch, C.; Fergus, R. Depth map prediction from a single image using a multi-scale deep network. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 2366–2374.
2. Mayer, N.; Ilg, E.; Hausser, P.; Fischer, P.; Cremers, D.; Dosovitskiy, A.; Brox, T. A large dataset to train convolutional net-works for disparity, optical flow, and scene flow estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4040–4048.
3. Garg, R.; Bg, V.K.; Carneiro, G.; Reid, I. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Springer International Publishing: Cham, Switzerland, 2016; pp. 740–756.
4. Xie, J.; Girshick, R.; Farhadi, A. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural net-works. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Springer International Publishing: Cham, Switzerland, 2016; pp. 842–857.
5. Godard, C.; Mac Aodha, O.; Brostow, G.J. Unsupervised monocular depth estimation with left-right consistency. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 270–279.
6. Ummenhofer, B.; Zhou, H.; Uhrig, J.; Mayer, N.; Ilg, E.; Dosovitskiy, A.; Brox, T. Demon: Depth and motion network for learning monocular stereo. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5038–5047.
7. Zhou, T.; Brown, M.; Snavely, N.; Lowe, D.G. Unsupervised learning of depth and ego-motion from video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1851–1858.
8. Vijayanarasimhan, S.; Ricco, S.; Schmid, C.; Sukthankar, R.; Fragkiadaki, K. SfM-Net: Learning of structure and motion from video. *arXiv* **2017**, arXiv:1704.07804.
9. Wang, C.; Buenaposada, J.M.; Zhu, R.; Lucey, S. Learning depth from monocular videos using direct methods. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2022–2030.
10. Mahjourian, R.; Wicke, M.; Angelova, A. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5667–5675.
11. Yin, Z.; Shi, J. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1983–1992.
12. Li, Z.; Dekel, T.; Cole, F.; Tucker, R.; Snavely, N.; Liu, C.; Freeman, W.T. Learning the depths of moving people by watching frozen people. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4521–4530.
13. Saha, S.; Obukhov, A.; Paudel, D.P.; Kanakis, M.; Chen, Y.; Georgoulis, S.; Van Gool, L. Learning to relate depth and seman-tics for unsupervised domain adaptation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 8197–8207.
14. Lin, X.; Sánchez-Escobedo, D.; Casas, J.R.; Pardàs, M. Depth estimation and semantic segmentation from a single RGB image using a hybrid convolutional neural network. *Sensors* **2019**, *19*, 1795. [CrossRef] [PubMed]
15. Klingner, M.; Termöhlen, J.A.; Mikolajczyk, J.; Fingscheidt, T. Self-supervised monocular depth estimation: Solving the dy-namic object problem by semantic guidance. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Springer International Publishing: Cham, Switzerland, 2020; pp. 582–600.
16. Wang, G.; Wang, H.; Liu, Y.; Chen, W. Unsupervised learning of monocular depth and ego-motion using multiple masks. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 4724–4730.
17. Bian, J.W.; Zhan, H.; Wang, N.; Li, Z.; Zhang, L.; Shen, C.; Cheng, M.M.; Reid, I. Unsupervised scale-consistent depth learning from video. *Int. J. Comput. Vis.* **2021**, *129*, 2548–2564. [CrossRef]
18. Arora, S.; Khandeparkar, H.; Khodak, M.; Plevrakis, O.; Saunshi, N. A Theoretical Analysis of Contrastive Unsupervised Representation Learning. *arXiv* **2019**, arXiv:1902.09229.
19. Huang, W.; Yi, M.; Zhao, X.; Jiang, Z. Towards the generalization of contrastive self-supervised learning. *arXiv* **2021**, arXiv:2111.00743.
20. Hendrycks, D.; Mazeika, M.; Kadavath, S.; Song, D. Using self-supervised learning can improve model robustness and uncertainty. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 15663–15674.
21. Doersch, C.; Gupta, A.; Efros, A.A. Unsupervised visual representation learning by context prediction. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1422–1430.

22. Hu, J.; Zhang, Y.; Okatani, T. Visualization of convolutional neural networks for monocular depth estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3869–3878.
23. Li, R.; Wang, S.; Long, Z.; Gu, D. Undeepvo: Monocular visual odometry through unsupervised deep learning. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 7286–7291.
24. Zhan, H.; Garg, R.; Weerasekera, C.S.; Li, K.; Agarwal, H.; Reid, I. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 340–349.
25. Chen, P.Y.; Liu, A.H.; Liu, Y.C.; Wang, Y.C.F. Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2624–2632.
26. Ye, X.; Fan, X.; Zhang, M.; Xu, R.; Zhong, W. Unsupervised monocular depth estimation via recursive stereo distillation. *IEEE Trans. Image Process.* **2021**, *30*, 4492–4504. [CrossRef] [PubMed]
27. Godard, C.; Mac Aodha, O.; Firman, M.; Brostow, G.J. Digging into self-supervised monocular depth estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3828–3838.
28. Li, Z.; Chen, Z.; Liu, X.; Jiang, J. Depthformer: Exploiting long-range correlation and local information for accurate monocular depth estimation. *arXiv* **2022**, arXiv:2203.14211. [CrossRef]
29. Ranftl, R.; Bochkovskiy, A.; Koltun, V. Vision transformers for dense prediction. In Proceedings of the IEEE/CVF Interna-tional Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 12179–12188.
30. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
31. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015; Springer International Publishing: Cham, Switzerland, 2015; pp. 234–241.
32. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2014.
33. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef]
34. He, T.; Zhang, Z.; Zhang, H.; Zhang, Z.; Xie, J.; Li, M. Bag of tricks for image classification with convolutional neural net-works. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 558–567.
35. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
36. Howard, A.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
37. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
38. Tan, M.; Le, Q. Efficientnetv2: Smaller models and faster training. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; pp. 10096–10106.
39. Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.
40. Liashchynskyi, P.; Liashchynskyi, P. Grid search random search genetic algorithm: A big comparison for NAS. *arXiv* **2019**, arXiv:1912.06059.
41. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language under-standing. *arXiv* **2018**, arXiv:1810.04805.
42. Palacio, S.; Folz, J.; Hees, J.; Raue, F.; Borth, D.; Dengel, A. What do deep networks like to see? In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3108–3117.
43. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
44. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16×16 words: Transformers for image recognition at scale. In Proceedings of the 9th International Conference on Learning Representations, Virtual Event, 3–7 May 2021.
45. Voita, E.; Talbot, D.; Moiseev, F.; Sennrich, R.; Titov, I. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 5797–5808.
46. Cordonnier, J.B.; Loukas, A.; Jaggi, M. Multi-head attention: Collaborate instead of concatenate. *arXiv* **2020**, arXiv:2006.16362.
47. Levine, Y.; Wies, N.; Sharir, O.; Bata, H.; Shashua, A. The depth-to-width interplay in self-attention. *arXiv* **2020**, arXiv:2006.12467.
48. Zhou, D.; Kang, B.; Jin, X.; Yang, L.; Lian, X.; Jiang, Z.; Hou, Q.; Feng, J. Deepvit: Towards deeper vision transformer. *arXiv* **2021**, arXiv:2103.11886.

49. Yuan, L.; Chen, Y.; Wang, T.; Yu, W.; Shi, Y.; Jiang, Z.H.; Tay, F.E.; Feng, J.; Yan, S. Tokens-to-token vit: Training vision trans-formers from scratch on imagenet. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 558–567.
50. Touvron, H.; Cord, M.; Sablayrolles, A.; Synnaeve, G.; Jégou, H. Going deeper with image transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 32–42.
51. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10012–10022.
52. Lee, Y.; Kim, J.; Willette, J.; Hwang, S.J. Mpvit: Multi-path vision transformer for dense prediction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 7287–7296.
53. Lin, G.; Liu, F.; Milan, A.; Shen, C.; Reid, I. RefineNet: Multi-path refinement networks for dense prediction. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 1228–1242. [CrossRef]
54. Shim, D.; Kim, H.J. SwinDepth: Unsupervised Depth Estimation using Monocular Sequences via Swin Transformer and Densely Cascaded Network. *arXiv* **2023**, arXiv:2301.06715.
55. Li, H.; Galayko, D.; Trocan, M.; Sawan, M. Cascade Decoders-Based Autoencoders for Image Reconstruction. *arXiv* **2021**, arXiv:2107.00002.
56. Majumdar, A. Blind denoising autoencoder. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *30*, 312–317. [CrossRef]
57. Wu, T.; Zhao, W.; Keefer, E.; Yang, Z. Deep compressive autoencoder for action potential compression in large-scale neural recording. *J. Neural Eng.* **2018**, *15*, 066019. [CrossRef] [PubMed]
58. Wang, W.; Xie, E.; Li, X.; Fan, D.P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. Pyramid vision transformer: A versatile back-bone for dense prediction without convolutions. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 568–578.
59. Li, Y.; Luo, F.; Xiao, C. Self-supervised coarse-to-fine monocular depth estimation using a lightweight attention module. *Comput. Vis. Media* **2022**, *8*, 631–647. [CrossRef]
60. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
61. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
62. Zhou, Z.; Siddiquee, M.M.R.; Tajbakhsh, N.; Liang, J. Unet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE Trans. Med. Imaging* **2019**, *39*, 1856–1867. [CrossRef]
63. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
64. Almalioglu, Y.; Saputra, M.R.U.; De Gusmao, P.P.; Markham, A.; Trigoni, N. Ganvo: Unsupervised deep monocular visual odometry and depth estimation with generative adversarial networks. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 5474–5480.
65. Li, J.; Zhao, J.; Song, S.; Feng, T. Unsupervised joint learning of depth, optical flow, ego-motion from video. *arXiv* **2021**, arXiv:2105.14520.
66. Battaglia, P.W.; Hamrick, J.B.; Bapst, V.; Sanchez-Gonzalez, A.; Zambaldi, V.; Malinowski, M.; Tacchetti, A.; Raposo, D.; San-toro, A.; Faulkner, R.; et al. Relational inductive biases, deep learning, and graphnetworks. *arXiv* **2018**, arXiv:1806.01261.
67. El-Shazly, E.H.; Zhang, X.; Jiang, J. Improved appearance loss for deep estimation of image depth. *Electron. Lett.* **2019**, *55*, 264–266. [CrossRef]
68. Gordon, A.; Li, H.; Jonschkowski, R.; Angelova, A. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Long Beach, CA, USA, 15–20 June 2019; pp. 8977–8986.
69. Ranjan, A.; Jampani, V.; Balles, L.; Kim, K.; Sun, D.; Wulff, J.; Black, M.J. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In Proceedings of the IEEE Conference on Com-puter Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 12240–12249.
70. Wang, R.; Pizer, S.M.; Frahm, J.M. Recurrent neural network for (un-) supervised learning of monocular video visual odom-etry and depth. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5555–5564.
71. Mandal, D.; Jain, A. Unsupervised Learning of Depth, Camera Pose and Optical Flow from Monocular Video. *arXiv* **2022**, arXiv:2205.09821.
72. Chen, Y.; Schmid, C.; Sminchisescu, C. Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Long Beach, CA, USA, 15–20 June 2019; pp. 7063–7072.
73. Almalioglu, Y.; Santamaria-Navarro, A.; Morrell, B.; Agha-Mohammadi, A.A. Unsupervised deep persistent monocular visual odometry and depth estimation in extreme environments. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 3534–3541.

74. Zhan, H.; Weerasekera, C.S.; Bian, J.W.; Reid, I. Visual odometry revisited: What should be learnt? In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 4203–4210.
75. Li, A.; Yuan, Z.; Ling, Y.; Chi, W.; Zhang, C. A multi-scale guided cascade hourglass network for depth completion. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Village, CO, USA, 1–5 March 2020; pp. 32–40.
76. Sattler, T.; Zhou, Q.; Pollefeys, M.; Leal-Taixe, L. Understanding the limitations of cnn-based absolute camera pose regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3302–3312.
77. Ng, T.; Lopez-Rodriguez, A.; Balntas, V.; Mikolajczyk, K. Reassessing the limitations of CNN methods for camera pose re-gression. *arXiv* **2021**, arXiv:2108.07260.
78. Meng, L.; Tung, F.; Little, J.J.; Valentin, J.; de Silva, C.W. Exploiting points and lines in regression forests for RGB-D camera relocalization. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 6827–6834.
79. Bleser, G.; Wuest, H.; Stricker, D. Online camera pose estimation in partially known and dynamic scenes. In Proceedings of the 2006 IEEE/ACM International Symposium on Mixed and Augmented Reality, Santa Barbard, CA, 22–25 October 2006; pp. 56–65.
80. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
81. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef] [PubMed]
82. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The kitti dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.