High quality monocular depth estimation with parallel decoder

Jiatao Liu & Yaping Zhang

Monocular depth estimation aims to recover the depth information in three-dimensional (3D) space from a single image efficiently, but it is an ill-posed problem. Recently, Transformer-based architectures have achieved excellent accuracy in monocular depth estimation. However, due to the characteristics of Transformer, the model parameters are huge and the inference speed is slow. In traditional convolutional neural network–based architectures, many encoder-decoders perform serial fusion of the multi-scale features of each stage of the encoder and then output predictions. However, in these approaches it may be difficult to recover the spatial information lost by the encoder during pooling and convolution. To enhance this serial structure, we propose a structure from the decoder perspective, which first predicts global and local depth information in parallel and then fuses them. Results show that this structure is an effective improvement over traditional methods and has accuracy comparable with that of state-of-the-art methods in both indoor and outdoor scenes, but with fewer parameters and computations. Moreover, results of ablation studies verify the effectiveness of the proposed decoder.
Related work
Monocular depth estimation from a single RGB image is an ill-posed problem since one picture can correspond to an infinite number of 3D scenes. Additionally, problems such as lack of scene coverage and semi-transparent or reflective materials can lead to fuzzy situations in which geometry cannot be derived from the appearance. In the past, methods that rely on CNNs can generate high-quality depth maps from a single RGB input image. CNN is a feedforward neural network that includes convolution computation. Compared with general neural networks, CNNs have the advantages that the network structure can better adapt to the image structure, and weight sharing can reduce the parameters of the network.

Monocular depth estimation is a low-cost method of distance measurement. When a supervised method is used to train the neural network, the supervised labels are usually obtained by a distance sensor such as an RGB-D camera or a multi-channel laser scanner. The method of using deep learning for monocular depth estimation started from a two-scale network proposed by Eigen et al. Then, several researchers proposed many efficient methods based on deep learning that use CNNs. Laina et al. used a fully convolutional residual network based on ResNet-50 and replaced the fully connected layer with a series of upsampling blocks. Alhashim et al. introduced skip connections in a simple encoder-decoder network architecture, and trained the model using transfer learning. Lee et al. replaced the standard upsampling layers with the local planar guidance layers to guide the reduced skip connections in a simple encoder-decoder network architecture, and trained the model using transfer learning. Long et al. proposed an adaptive surface normal constraint method that effectively correlates the depth estimation with geometric consistency, which can faithfully reconstruct 3D geometry and is robust to local shape changes. Yin et al. designed a loss term to enforce a simple geometric constraint, which can greatly improve the accuracy of depth estimation. Chen et al. proposed a structure-aware residual pyramid network to exploit multi-scale structures in complex scenes for accurate depth prediction. Fu et al. found that if the depth regression task is transformed into a classification task its performance can be improved. Bhat et al. designed the AdaBins module, which divides the depth range into 256 intervals, with the center value of each interval being the depth value of the pixels falling in the interval, and the final depth of a pixel being a linear combination of the center depth values of the intervals.

Transfer learning is attracting increasingly more attention since it can make full use of the previously marked data while ensuring accuracy of the model on the new task. A recent transfer learning method has shown its effectiveness in many tasks, such as image classification and image semantic segmentation. For monocular depth estimation tasks, several researchers have used a pre-trained model with high accuracy in image-classification tasks as the encoder in the network architecture. Transformer, a self-attention-based architecture, has been widely used in natural language processing (NLP) in recent years. Dosovitskiy et al. applied Transformer architecture from NLP to image classification directly, and the model is particularly successful when it is instantiated as a high-capacity architecture and trained on a very large dataset. Ranftl et al. applied this ViT to monocular depth estimation, and also obtained a high-precision depth estimation model through training on a large number of datasets.

Proposed method
In this section, we detail our method for depth estimation from a single image. Our main idea is to design an end-to-end deep-learning network architecture that can predict the global and local depth information of the input image in parallel. The overall architecture of the proposed model follows encoder–decoder network architecture, and the way to combine local and global information is to use the Transformer encoder, which is able to achieve global self-attention of the image.

Motivation. Even though ViT-based architectures have achieved excellent performance, their problems, including the huge number of parameters and slow inference speed, cannot be ignored; in addition, the model requires a large quantity of data to train. These problems may be acceptable for tasks with low-resolution inputs, but quickly become intractable for higher-resolution inputs. In the depth estimation task, the input image usually has a higher resolution, and the use of CNN-based architecture may have more application significance.

In traditional CNN-based architectures, for the ill-posed problem of monocular depth estimation, many researchers try to explore solutions by synthesizing global and local depth information. The local details of the input image may be lost in CNN-based encoders since they use gradual downsampling to gradually increase the receptive field. We propose to use CNNs to design neural networks in a way that does not lose detailed information as much as possible.

The shallower features of an encoder have higher resolution and contain more details and spatial information. Therefore, we propose to calculate these high-resolution features directly to obtain predictions with rich local details and fuse them with more accurate global predictions obtained from deeper parts of the neural network. Many prior works simply upsampled the deep features and progressively concatenated them with the shallower features. It may be difficult in this method to recover the spatial information lost by the encoder during pooling and convolution, which causes boundary blur in high-resolution predictions. Therefore, we propose a parallel decoder structure that directly uses the multi-resolution feature vectors of the encoder in multiple stages for parallel prediction, and then fuses the multi-stage predictions for output.

Network architecture. Figure 1 shows the overview of the proposed network architecture for depth estimation. For the encoder, we use EfficientNet-B7 pre-trained on ImageNet. The input image is encoded into multi-scale feature vectors at different stages of the encoder. The different feature vectors first go through a series of SENet-based residual blocks proposed by Hu et al., which consists of three convolution layers and a SELayer. SELayer is a channel attention mechanism. It first performs the average pooling operation to shrink feature maps through spatial dimensions, and then calculates the weight of each channel through two fully connected layers.
and a sigmoid operation, and multiplies them with the corresponding channel features, respectively, to reweight the feature maps. After that, the feature vectors perform a series of convolution and resampling operations to obtain features with the same shape as shown in features1 to features5 in Fig. 1, and perform feature fusion to obtain parallel prediction results that focus on global or local as shown in block1 to block4 in Fig. 1. Finally, self-attention-based Transformer is used to fuse the predictions, and the final prediction is output through the convolution head.

Specifically, for the encoder, we choose features with post-encoding resolutions of 1/2, 1/4, 1/8, 1/16, and 1/32 of the input resolution. For the decoder, features with higher resolution go through more SENet-based residual blocks. Note that we add a separate channel attention layer after the last residual block and add a residual connection from the encoder to this layer. The feature maps with different scales are convolved and resampled to have the same number of channels and resolution. Considering the memory limit of the graphical processing unit (GPU) employed, we set the number of the channels to 30 and the resolution to 1/2 of the input resolution. The shallower features contain more local details and spatial information, and the deeper features contain more accurate predictions of global depth information. For block1 to block4, we hope that deeper predictions will focus more on global predictions. The local predictions need to refer to the global predictions, and the reverse is also true. Therefore, as shown in Fig. 1, block1 is obtained by convolution of features1, features2, and features5; block2 is obtained by convolution of features2, features3, and features5; block3 is obtained by convolution of features3, features4, and features1; block4 is obtained by convolution of features4, features5, and features1.

The Transformer module that fuses the predictions of block1 to block4 can be regarded as a simplified version of Adabins proposed by Bhat et al. As shown in Fig. 2, we must convert these 2D tensors into 1D sequences before inputting the predictions into Transformer. The process is as follows:

\[
x \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times 4} \rightarrow x \in \mathbb{R}^{\frac{H}{p} \times \frac{W}{p} \times E} \rightarrow x \in \mathbb{R}^{S \times E}.
\] (1)

Specifically, we first pass these predictions through a convolutional layer with kernel size \(p \times p\), stride \(p\), and output channels \(E\). Thus, the output of this convolution is a tensor of shape \(\frac{H}{p} \times \frac{W}{p} \times E\); we then flatten this tensor to \(S \times E\), where \(S = \frac{HW}{p^2}\). In this way, a set of feature vectors that can be input into the Transformer encoder is obtained. After restoring the output of the Transformer encoder as a 2D tensor, we use it as the weight matrix \(N\). On the other side, we pass the predictions through a convolutional layer with kernel size \(3 \times 3\) and obtain a tensor \(Z\) of shape \(\frac{H}{2} \times \frac{W}{2} \times C\). After doing the pixel-wise dot product of \(N\) and \(Z\), the final predicted depth map is output through the convolution head. In our experiment, \(p\) was set to 16, \(E\) to 4, and \(C\) to 128.

**Loss function.** Pixel-level depth loss for block1 to block4. We first introduced the loss function for block1 to block4 as shown in Fig. 1. Assuming that the maximum depth in the ground truth is \(d_{\text{max}}\) and the mini-
mum depth is $d_{\text{min}}$, we first divided the depth range $[d_{\text{min}}, d_{\text{max}}]$ into $b$ intervals on average; that is, the length of a single interval is $\text{len} = \frac{d_{\text{max}} - d_{\text{min}}}{b}$, and the depth range corresponding to the $i$th ($i \leq b$) interval is $[d_{\text{min}} + (i - 1) \times \text{len}, d_{\text{min}} + i \times \text{len}]$. Then we made a histogram of the ground truth in these $b$ intervals to find the interval that accounts for the largest proportion of the depth of the scene; most of the global information is contained in this interval. Correspondingly, the interval that accounts for the smaller proportion contains more local information. In our experiment, $b$ was set to 10, and we arranged the 10 intervals in descending proportions. Note that most of the depth values in a scene are distributed in the first and second intervals, which is unfavorable for optimizing the model in parallel, so we proposed that block 1 focuses on the fifth to tenth interval, block 2 on the fourth to eighth interval, block 3 on the second to fourth interval, and block 4 on the first and second interval. We then calculated the mean-squared-error loss on these intervals separately:

$$L_{\text{focus}}(y, \text{block1}, \text{block2}, \text{block3}, \text{block4}) = \sum_{i=1}^{4} \left( \frac{\lambda_i}{n_i} \sum_{p} d_{p}^2 \right),$$  \hspace{1cm} (2)

where $y$ is the ground truth, $\lambda_1 = 0.5, \lambda_2 = \lambda_3 = 0.6, \lambda_4 = 1, n_i$ is the total number of pixels after $y$ passes through the mask of the corresponding interval, $d_{p} = y_{p} - \text{block}_i p$, and $y_{p}$ and $\text{block}_i p$ are the depth value of the pixel $p$ in the masked ground truth $y$ and block$i$, respectively.

**Pixel-level depth loss for output.** We define the pixel-level loss for the output as a combination of two parts: one is the mean-squared-error loss of each pixel depth value in log space, and the other is the variance of the error of each pixel depth value in the log space. Note that the second part is used to better predict small details in the image that are more difficult to predict accurately. To improve the sensitivity when the loss value gradually tends to zero, the loss term is a square root of the combination:

$$L_{\text{depth}}(y, \hat{y}) = \sqrt{\frac{1}{n} \sum_{p} d_{p}^2 + \frac{1}{n} \sum_{q} \left( \frac{1}{n} \sum_{p} d_{p} - d_{q} \right)^2},$$  \hspace{1cm} (3)

where $d_{p} = \ln_{p} - \ln_{\hat{y}_p}$, $y_p$ and $\hat{y}_p$ are the depth values of the pixel $p$ in the ground truth $y$ and the predicted depth map $\hat{y}$, respectively, and $n$ is the total number of pixels in the depth map. Therefore, this loss term can not only control the global overall prediction, but also stabilize the local detailed prediction.

**Multi-scale structure similarity loss for output.** The multi-scale structure similarity loss (MS-SSIM) loss function scales the picture from large to small according to certain rules, and then calculates the structural similarity, which is equivalent to considering the resolution of the image and retaining the high-frequency information in the image. This loss term is defined as follows:

$$L_{\text{ms-ssim}}(y, \hat{y}) = \sqrt{1 - \text{MS-SSIM}(y, \hat{y})}.$$  \hspace{1cm} (4)

The training loss is defined as follows:

$$L = \eta L_{\text{focus}} + \alpha L_{\text{depth}} + \beta L_{\text{ms-ssim}},$$  \hspace{1cm} (5)

where $\eta = 0.1, \alpha = 8$, and $\beta = 10$.

**Experiments**

We conducted an extensive set of experiments to verify the effectiveness of our method. First, we describe the datasets used and then present the implementation details of the proposed method and evaluate it qualitatively and quantitatively. Finally, we carried out a series of ablation studies to facilitate discussion of the core factors proposed in detail.
Datasets. NYU Depth v2 is a commonly used dataset in depth estimation tasks. It provides RGB images of different indoor scenes and their corresponding depth maps. The image depth ranges from 0 to 10 m and the image resolution is 480 × 640. The dataset contains approximately 120,000 training samples, and we trained our model on the subset of 50,688 samples supplied by Alhashim et al. For the depth values of some pixels in the original depth map in the dataset are missing due to the hardware of the acquisition device. The dataset uses the method proposed by Levin et al.25 to fill in the missing parts. In addition, we use image random horizontal flip and random color to expand the training data to reduce overfitting and improve the generalization performance of the model. Random color augmentation includes randomly changing the saturation, hue, brightness, gamma and contrast of an image. The resolution of the final output depth map is half of the input image. For evaluation, we use the 654 official testing images as the input of the model, and the output is upsampled by a factor of 2 to match the original depth resolution. In addition, the cropping boundary method proposed by Eigen et al. is used. After the original image and horizontally flipped image are used as the model input, the average of the predicted depth of the two images is taken as the final output of the model and evaluated.

KITTI provides outdoor stereo images and corresponding 3D laser scanning, which are captured by equipment mounted on a moving vehicle. It includes 61 scenes collected from "city", "residential", "road" and "campus" categories, with an image resolution of 1241 × 376, and provides sparse ground truth. For training, the images are randomly cropped to 704 × 352, and the data augmentation of random horizontal flipping and random color is used. For testing, we use 652 test images out of 697 test images split by Eigen et al. to evaluate the model performance (45 images without corresponding ground truth are excluded from the test set). The central cropping scheme proposed by Grag et al. is also adopted for the evaluation. We bilinearly upsample the model prediction to match the ground truth resolution. The final depth is computed in the same way as for NYU Depth v2 dataset.

SUN RGB-D is an indoor dataset that has more diverse scenes and contains approximately 10,000 images collected by four different sensors. For our experiments only the official test set of 5,050 images were used for testing, without training.

DIODE (Dense Indoor/Outdoor DEpth) is a dataset that contains thousands of diverse high resolution color images with accurate, dense, long-range depth measurements. We use the validation set for testing, which contains 325 indoor images and 446 outdoor images.

ETH-3D provides multi-view stereo/3D reconstruction benchmark covering various indoor and outdoor scenes. Ground truth geometry has been obtained by using a high-precision laser scanner. In the training set with ground truth, 219 indoor images in 7 scenes and 235 outdoor images in 6 scenes are used for our cross-dataset testing.

ScanNet is an indoor RGB-D video dataset, containing 2.5 million views in more than 1500 scans, annotated with 3D camera poses, surface reconstructions, and instance-level semantic segmentations. We use 100 test scenes, including 2135 frames, to evaluate cross-dataset generalization ability.

Implementation details. We used the PyTorch framework to implement the proposed network structure, and trained the model on a single NVIDIA GeForce RTX 3090 GPU. When training the model, we used AdamW with a weight decay of 10^-2 as the optimizer. For learning rate, we used the 1-cycle strategy with a maximum learning rate of 2 × 10^-4. In the first 30% of iterations, the learning rate linearly warmed up from 2×10^-4 to 2 × 10^-4, and then cosine- annealed to 2×10^-4. The batch size was set to 4. We trained 20 epochs on NYU Depth v2 dataset and 30 epochs on KITTI dataset, taking 105 min and 49 min per epoch respectively. The proposed model has about 77.0 × 10^6 parameters: 63.8 × 10^6 for the encoder and 13.2 × 10^6 for the decoder.

Evaluation. Quantitative evaluation. Table 1 shows the quantitative comparison between the proposed method and eight state-of-the-art methods on NYU Depth v2 dataset, using six standard metrics introduced by Eigen et al. These metrics are threshold accuracy (δ_i, i = 1, 2, 3), average relative error (REL), logarithmic average error (Log10), and root-mean-square error (RMS), and are defined, respectively, as follows:

| Method          | δ_{0.01} | δ_{0.02} | δ_{0.1} | REL  | RMS  | Log10 |
|-----------------|----------|----------|---------|------|------|-------|
| Eigen et al.    | 0.769    | 0.950    | 0.988   | 0.158| 0.641| -     |
| Laina et al.    | 0.811    | 0.953    | 0.988   | 0.127| 0.573| 0.055 |
| Fu et al.       | 0.828    | 0.965    | 0.992   | 0.115| 0.509| 0.051 |
| Alhashim et al. | 0.846    | 0.974    | 0.994   | 0.123| 0.465| 0.053 |
| Yin et al.      | 0.875    | 0.976    | 0.994   | 0.108| 0.416| 0.048 |
| Lee et al.      | 0.885    | 0.978    | 0.994   | 0.110| 0.392| 0.047 |
| Ilbat et al.    | 0.903    | 0.984    | 0.997   | 0.103| 0.364| 0.044 |
| Ranftl et al.   | 0.904    | 0.988    | 0.998   | 0.110| 0.357| 0.045 |
| Proposed        | 0.905    | 0.987    | 0.997   | 0.101| 0.359| 0.043 |
where in Eq. (6), when \( i = 1, 2, 3 \), \( thr = 1.25, 1.25^2, 1.25^3 \), respectively. Additionally for KITTI dataset, as shown in Table 2, we use the two standard metrics: square relative difference (Sq. Rel) and root-mean-square error in the log space (RMS log), as follows:

\[
\text{REL} = \frac{1}{n} \sum_{p} \frac{|y_p - \hat{y}_p|}{y_p},
\]

\[
\text{Log10} = \frac{1}{n} \sum_{p} |\log_{10}(y_p) - \log_{10}(\hat{y}_p)|,
\]

\[
\text{RMS log} = \sqrt{\frac{1}{n} \sum_{p} (\ln y_p - \ln \hat{y}_p)^2},
\]

where in Eq. (6), when \( i = 1, 2, 3 \), \( thr = 1.25, 1.25^2, 1.25^3 \), respectively. Additionally for KITTI dataset, as shown in Table 2, we use the two standard metrics: square relative difference (Sq. Rel) and root-mean-square error in the log space (RMS log), as follows:

\[
\text{Sq. Rel} = \frac{1}{n} \sum_{p} \frac{||y_p - \hat{y}_p||}{y_p},
\]

\[
\text{RMS log} = \sqrt{\frac{1}{n} \sum_{p} (\ln y_p - \ln \hat{y}_p)^2}.
\]

From the Tables 1 and 2, the proposed method is comparable to the state-of-the-art methods in terms of accuracy and error.

We considered the methods proposed by Bhat et al. and Ranftl et al. as the two most important competitors. Note that while the encoders of both methods and the proposed method are pre-trained on ImageNet, the method presented in Ref. 3 needed extra training data and fine-tuning on NYU Depth v2 dataset to obtain better results. Specifically, the method in Ref. 3 must first be trained on a dataset with \( 1.4 \times 10^6 \) images for 60 epochs and then fine-tuned on NYU Depth v2, but the model of Bhat et al. and that proposed herein only need be trained on a 50,688-image subset of NYU Depth v2 for 25 and 20 epochs, respectively. We compared the complexity of the models, including the amount of parameters, computation and the inference speed. The amount of computation of the model is measured by the MACs (Multiply-Accumulate operations) with an input resolution of \( 480 \times 640 \). As shown in Table 3, the proposed model has fewer parameters and computations than the other two models, and the inference speed is the fastest among the three models. Note that our experiments for inference speed were done on a single NVIDIA GeForce GTX 1660ti GPU, and since the output resolution of

| Method          | Parameters \((\times 10^5)\) | MACs \((\times 10^9)\) | Time (ms) | FPS |
|-----------------|-------------------------------|------------------------|-----------|-----|
| Bhat et al.     | 78.26                         | 186.33                 | 108       | 9.26|
| Ranftl et al.   | 123.15                        | 229.11                 | 199       | 5.03|
| Proposed        | 78.99                         | 79.73                  | 103       | 9.71|

Table 2. Quantitative comparison with prediction results of other models on KITTI dataset. The best results are shown in bold, and the second best results are italics.

| Method          | Parameters \((\times 10^5)\) | MACs \((\times 10^9)\) | Time (ms) | FPS |
|-----------------|-------------------------------|------------------------|-----------|-----|
| Bhat et al.     | 78.26                         | 186.33                 | 108       | 9.26|
| Ranftl et al.   | 123.15                        | 229.11                 | 199       | 5.03|
| Proposed        | 78.99                         | 79.73                  | 103       | 9.71|

Table 3. Model statistics.
respectively. We bilinearly resampled the images to $512 \times 384$ and $640 \times 320$, respectively. For ScanNet dataset, proposed by Grag et al.\(^27\) is adopted. The quantitative comparison of model generalization ability in indoor and outdoor scenes and better in some scenarios.

We performed several ablation studies to analyze several factors that may have an impact on our results. In these studies, except for the different subjects studied, the other strategies were the same in each group of experiments. The factors analyzed were the following.

**Parallel prediction decoder.** We first evaluated the effectiveness of the proposed decoder, which implements parallel prediction of global and local depth information. The structure of the comparison method is similar to\(^6\) and the backbone structure of\(^14\). Specifically, it upsamples the features of the deepest layer of the encoder to the same resolution as the features of the sub-deep layer, and then concatenates the features after resampling with the features of the sub-deep layer so as to gradually reach the shallowest layer. We trained 20 epochs on NYU Depth v2 dataset, and the quantitative comparison results are shown in Table 6. It can be seen from the table that the proposed parallel method has advantages over the serial method in terms of accuracy, error, and parameter size.

Moreover, we replaced different encoders to further study the boosting effect of the proposed parallel decoder. After changing the encoder from Efficientnet-B7 to B2 and B5, we separately used the traditional and proposed parallel methods to decode the features of the sub-deep layer so as to gradually reach the shallowest layer. We trained 20 epochs on NYU Depth v2 dataset, and the quantitative comparison results are shown in Table 6. It can be seen from the table that the proposed parallel method has advantages over the serial method in terms of accuracy, error, and parameter size. And the results are shown in Fig. 7. We recorded the accuracy ($\delta_1$) and error (RMS) of

| Method            | SUN RGBD | DIODE indoor | ETH-3D indoor | ScanNet |
|-------------------|----------|--------------|---------------|---------|
|                   | $\delta_1$ | REL$_1$ | $\delta_1$ | REL$_1$ | $\delta_1$ | REL$_1$ | $\delta_1$ | REL$_1$ |
| Lee et al.$^7$    | 0.448    | 0.528       | 0.307         | 0.380   | 0.332       | 0.424   | 0.267       | 0.367   |
| Bhat et al.$^{14}$| 0.553    | 0.334       | 0.265         | 0.407   | 0.348       | 0.375   | 0.643       | 0.208   |
| Ranftl et al.$^3$ | 0.500    | 0.355       | 0.439         | 0.323   | 0.465       | 0.294   | 0.744       | 0.190   |
| Proposed          | 0.568    | 0.328       | 0.361         | 0.352   | 0.343       | 0.371   | 0.695       | 0.193   |

Table 4. Quantitative comparison of model generalization ability in indoor scenes. The best results are shown in bold, and the second best results are italics.

| Method            | DIODE outdoor | ETH-3D outdoor |
|-------------------|---------------|---------------|
|                   | $\delta_1$ | REL$_1$ | RMS$_1$ | $\delta_1$ | REL$_1$ | RMS$_1$ |
| Lee et al.$^7$    | 0.196    | 0.551 | 0.656 | 10.071   | 0.208   | 0.546 | 0.904 | 5.472   |
| Bhat et al.$^{14}$| 0.169    | 0.534 | 0.679 | 10.184   | 0.127   | 0.515 | 0.886 | 5.872   |
| Ranftl et al.$^3$ | 0.247    | 0.761 | 0.647 | 7.492    | 0.261   | 0.519 | 0.991 | 5.147   |
| Proposed          | 0.174    | 0.512 | 0.699 | 10.453   | 0.123   | 0.560 | 0.819 | 5.659   |

Table 5. Quantitative comparison of model generalization ability in outdoor scenes. The best results are shown in bold, and the second best results are italics.
the training results separately and the comparison of the model parameters is reflected in the figure. The figure demonstrates that the proposed parallel decoder is a significant improvement over the traditional serial decoder.

**Transformer for fusing parallel prediction.** We introduced an experiment to study the effectiveness of using Transformer for fusion in the proposed decoder. The calculation method for comparison is

$$output = \sum_{i=1}^{4} (F_{\text{sigmoid}}(F_{\text{conv}}(\text{blocki})) \times \text{blocki}),$$  \hspace{1cm} (12)
Figure 5. Visualization of decoded features. (a) RGB image and the ground truth. (b) Several channels in (b) features1, (c) features2, (d) features3, (e) features4, and (f) features5.

Figure 6. Depth predicted by the proposed model generates a 3D point cloud result.

Table 6. Quantitative comparison of serial and parallel decoders. The better results are shown in bold.
wherein the block<sub>i</sub> (1 ≤ i ≤ 4) is first convolved, and then the sigmoid function is used to map it to between 0 and 1 and then multiply with the block<sub>i</sub>; finally, all the block<sub>i</sub> are added pixel by pixel to obtain the output. Note that the encoder in this experiment used EfficientNet-B3 pre-trained on ImageNet, which has fewer parameters and is faster to train. We trained 20 epochs on NYU Depth v2 dataset, and the quantitative comparison results are shown in Table 7. It can be seen from the table that using Transformer has advantages in terms of accuracy and error compared with simple calculation fusion, and the parameters do not increase too much.

Number of depth intervals. We then introduced a set of experiments to study the influence of the number of depth intervals b in the pixel-level depth loss for block1 to block4 on the experimental results. We mainly conducted three experiments, with depth interval numbers of 1, 4, and 10. Note that the encoder in this set of experiments also used EfficientNet-B3 pre-trained on ImageNet. We trained 20 epochs on NYU Depth v2 dataset, and the quantitative comparison results are shown in Table 8. The table shows that the number of depth intervals may have little impact on the experimental results, and we thus chose a relatively better 10-interval division in the loss item.

Loss function for pixel-level depth of final prediction. To study the effect of the combination of variance and mean-squared error used in the pixel-level loss in the final prediction, we compared the loss term with the scale invariant (SI) loss introduced by Eigen et al. We trained 20 epochs on NYU Depth v2 dataset, and the quantitative comparison results are shown in Table 9. It can be seen from the table that except for δ₁, the values on the other indicators are the same. Therefore, it can also be seen that using the loss constraint we proposed can better control the small, difficult-to-predict parts of the scene.

Loss function for pixel-level depth loss for block1 to block4. To study the effectiveness of the loss term that we designed for block1 to block4, we removed the loss term and conducted an experiment. We trained 20 epochs on NYU Depth v2 dataset, and the quantitative comparison results are shown in Table 10. It can be seen from the table that using this loss term during training can obtain better results.

Weight of each loss term. The weight of the loss item will directly affect the model training process and final result. To roughly find a suitable set of weight combinations, we conducted ablation studies with different values.

### Table 7. Quantitative comparison of different fusion methods. The better results are shown in bold.

| Method   | δ₁↑ | δ₂↑ | δ₃↑ | REL↓ | RMS↓ | Log₁₀↓ | Param↓ |
|----------|-----|-----|-----|------|------|--------|--------|
| Sigmoid  | 0.867 | 0.978 | 0.996 | 0.117 | 0.401 | 0.049  | 21.02 M |
| Transformer | 0.877 | 0.979 | 0.995 | 0.112 | 0.398 | 0.048  | 21.14 M |

### Table 8. Quantitative comparison of different numbers of depth intervals. The better results are shown in bold.

| Intervals | δ₁↑ | δ₂↑ | δ₃↑ | REL↓ | RMS↓ | Log₁₀↓ |
|-----------|-----|-----|-----|------|------|--------|
| 1         | 0.875 | 0.979 | 0.996 | 0.115 | 0.392 | 0.052  |
| 4         | 0.872 | 0.980 | 0.996 | 0.113 | 0.400 | 0.049  |
| 10        | 0.877 | 0.979 | 0.995 | 0.112 | 0.398 | 0.048  |

Figure 7. Performance comparisons of (a) accuracy and (b) error of serial and parallel decoders using different encoders.
for $\lambda_1, \lambda_2, \lambda_3$, and $\lambda_4$ in Eq. (2) and $\eta, \alpha$, and $\beta$ in Eq. (5). We trained 20 epochs on NYU Depth v2 dataset and recorded the RMS error as shown in Fig. 8, where the numbers in the legend, reading across, represent the values of $\eta, \lambda_1, \lambda_2, \lambda_3, \lambda_4, \alpha$, and $\beta$, respectively. It can be seen that, when $\eta = 0.1, \lambda_1 = 0.5, \lambda_2 = \lambda_3 = 0.6, \lambda_4 = 1, \alpha = 8$, and $\beta = 10$, the error that the model can achieve after training is the lowest.

### Conclusions

In this work, we propose a novel end-to-end supervised monocular depth estimation network and achieve state-of-the-art results. We design a decoder that realizes parallel prediction and then fuses the predictions for output. We deploy this decoder into multiple stages of an encoder, combine a loss term based on depth interval partitioning to obtain significant improvements, and present a number of experimental results on challenging benchmarks to validate it. In future work, we aim to apply this work to specific applications, such as 3D reconstruction.

### Data availability

All data generated or analyzed during this study are included in this published article. We provide the proposed model code and pre-training parameters at: https://github.com/jt-liu/PDNet, accessed on 25 September 2022.

Received: 10 May 2022; Accepted: 20 September 2022
Published online: 05 October 2022

### References

1. Eigen, D., Puhrsch, C. & Fergus, R. Depth map prediction from a single image using a multi-scale deep network. arXiv:1406.2283 (2014).  
2. Dosovitskiy, A. et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv:2010.11929 (2020).  
3. Ranftl, R., Bochkovskiy, A. & Koltun, V. Vision transformers for dense prediction. In Proceedings of the IEEE/CVF International Conference on Computer Vision 12179–12188 (2021).  
4. Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention 234–241 (Springer, 2015).

---

**Table 9.** Quantitative comparison of different loss functions for pixel-level depth of final prediction. The better results are shown in bold.

| Loss   | $\delta_1$↑ | $\delta_2$↑ | $\delta_3$↑ | REL↓ | RMS↓ | Log↑↓ |
|--------|-------------|-------------|-------------|------|------|-------|
| SI     | 0.898       | 0.984       | 0.996       | 0.102| 0.362| 0.043 |
| Ours   | 0.905       | 0.984       | 0.996       | 0.102| 0.362| 0.043 |

**Table 10.** Quantitative comparison of different loss functions for pixel-level depth of block1 to block4. The better results are shown in bold.

| Method         | $\delta_1$↑ | $\delta_2$↑ | $\delta_3$↑ | REL↓ | RMS↓ | Log↑↓ |
|----------------|-------------|-------------|-------------|------|------|-------|
| Without loss term | 0.895       | 0.985       | 0.997       | 0.105| 0.366| 0.044 |
| With loss term   | 0.905       | 0.984       | 0.996       | 0.102| 0.362| 0.043 |

**Figure 8.** Comparison of the effect of using different loss function weights on training.
5. Liu, F., Zhang, Z. & Zhou, R. Automatic modulation recognition based on CNN and GRU. Tsinghua Sci. Technol. 27, 422–431 (2021).
6. Hartpence, B. & Kwapisinski, A. CNN and MLP neural network ensembles for packet classification and adversary defense. Intell. Converged Netw. 2, 66–82 (2021).
7. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F. & Navab, N. Deeper depth prediction with fully convolutional residual networks. In 2016 Fourth international conference on 3D vision (3DV) 239–248 (IEEE, 2016).
8. Alhashim, I. & Wonka, P. High quality monocular depth estimation via transfer learning. arXiv:1812.11941 (2018).
9. Lee, J.H., Han, M.-K., Ko, D.W. & Suh, I.H. From big to small: Multi-scale local planar guidance for monocular depth estimation. arXiv:1907.10326 (2019).
10. Long, X. et al. Adaptive surface normal constraint for depth estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision 12849–12858 (2021).
11. Yin, W., Liu, Y., Shen, C. & Yan, Y. Enforcing geometric constraints of virtual normal for depth prediction. In Proceedings of the IEEE/CVF International Conference on Computer Vision 5684–5693 (2019).
12. Chen, X., Shen, L. & Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 4009–4018 (2021).
13. Fu, H., Gong, M., Wang, C., Batmanghelich, K. & Tao, D. Deep ordinal regression network for monocular depth estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2002–2011 (2018).
14. Bhat, S.F., Alhashim, I. & Wonka, P. Adabins: Depth estimation using adaptive bins. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2022 (2022).
15. Shaha, M. & Pawar, M. Transfer learning for image classification. In 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA) 656–660 (IEEE, 2018).
16. Krishna, S. T. & Kalluri, H. K. Deep learning and transfer learning approaches for image classification. Int. J. Recent Technol. Eng. (IJRTE) 7, 427–432 (2019).
17. Wurm, M., Stark, T., Zhu, X. X., Weigand, M. & Taubenbock, H. Semantic segmentation of slums in satellite images using transfer learning on fully convolutional neural networks. ISPRS J. Photogramm. Remote Sens. 150, 59–69 (2019).
18. Brown, T.B. et al. Language models are few-shot learners. arXiv:2005.14165 (2020).
19. Liu, Y. et al. Roberta: A robustly optimized bert pretraining approach. arXiv:1907.11692 (2019).
20. Tan, M. & Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In International Conference on Machine Learning 6105–6114 (PMLR, 2019).
21. Deng, J. et al. Imagenet: A large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition 248–255 (IEEE, 2009).
22. Hu, J., Shen, L. & Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 7132–7141 (2018).
23. Wang, Z., Simoncelli, E.P. & Bovik, A.C. Multiscale structural similarity for image quality assessment. In The Thirteenth Asilomar Conference on Signals, Systems & Computers, 2003 Vol. 2 1398–1402 (IEEE, 2003).
24. Silverman, N., Hoiem, D., Kohli, P. & Fergus, R. Indoor segmentation and support inference from rgbd images. In European Conference on Computer Vision 746–760 (Springer, 2012).
25. Levin, A., Lischinski, D. & Weiss, Y. Colorization using optimization. In ACM SIGGRAPH 2004 Papers 689–694 (2004).
26. Geiger, A., Lenz, P., Stiller, C. & Urtasun, R. Vision meets robotics: The kitti dataset. Int. J. Robot. Res. 32, 1231–1237 (2013).
27. Garg, R., Berg, V.K., Carneiro, G. & Reid, I. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In European Conference on Computer Vision 740–756 (Springer, 2016).
28. Song, S., Lichtenberg, S.F. & Xiao, J. Sun rgb-d: A rgb-d scene understanding benchmark suite. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 567–576 (2015).
29. Xiao, J., Owens, A. & Torralba, A. Sun3d: A database of big spaces reconstructed using sfm and object labels. In Proceedings of the IEEE International Conference on Computer Vision 1625–1632 (2013).
30. Janoch, A. et al. A category-level 3d object dataset: Putting the kitnet to work. In Consumer Depth Cameras for Computer Vision 141–165 (Springer, 2013).
31. Vasič, I. et al. Diode: A dense indoor and outdoor depth dataset. arXiv:1908.00463 (2019).
32. Schops, T. et al. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 3260–3269 (2017).
33. Dai, A. et al. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 5828–5837 (2019).
34. Paszke, A. et al. Pytorch: An imperative style, high-performance deep learning library. Adv. Neural. Inf. Process. Syst. 32, 8026–8037 (2019).
35. Loshchilov, I. & Hutter, F. Decoupled weight decay regularization. arXiv:1711.05101 (2017).
36. Smith, L.N. & Topin, N. Super-convergence: Very fast training of neural networks using large learning rates. In Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications Vol. 11006 1100612 (International Society for Optics and Photonics, 2019).

Author contributions
J.L.: conceptualization, software, investigation, visualization, validation, writing, revision.Y.Z.: conceptualization, methodology, writing, revision, supervision, financial support.

Funding
Yaping Zhang was funded by National Natural Science Foundation of China (Grant no. 61863037) and Ten Thousand Talent Plans for Young Top-notch Talents of Yunnan Province.

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to Y.Z.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.
