Improving Depth Gradient Continuity in Transformers: A Comparative Study on Monocular Depth Estimation with CNN

Jiawei Yao[1]
jwyao@uw.edu
Tong Wu[1]
tw96@uw.edu
Xiaofeng Zhang[2]
framebreak@sjtu.edu.cn

[1] School of Engineering and Technology,
University of Washington,
Tacoma, USA

[2] School of Electronic Information and
Electrical Engineering,
Shanghai Jiao Tong University,
Shanghai, China

Abstract

Monocular depth estimation is an ongoing challenge in computer vision. Recent progress with Transformer models has demonstrated notable advantages over conventional CNNs in this area. However, there's still a gap in understanding how these models prioritize different regions in 2D images and how these regions affect depth estimation performance. To explore the differences between Transformers and CNNs, we employ a sparse pixel approach to contrastively analyze the distinctions between the two. Our findings suggest that while Transformers excel in handling global context and intricate textures, they lag behind CNNs in preserving depth gradient continuity. To further enhance the performance of Transformer models in monocular depth estimation, we propose the Depth Gradient Refinement (DGR) module that refines depth estimation through high-order differentiation, feature fusion, and recalibration. Additionally, we leverage optimal transport theory, treating depth maps as spatial probability distributions, and employ the optimal transport distance as a loss function to optimize our model. Experimental results demonstrate that models integrated with the plug-and-play Depth Gradient Refinement (DGR) module and the proposed loss function enhance performance without increasing complexity and computational costs on both outdoor KITTI and indoor NYU-Depth-v2 datasets. This research not only offers fresh insights into the distinctions between Transformers and CNNs in depth estimation but also paves the way for novel depth estimation methodologies.

1 Introduction

Monocular depth estimation aims to perceive the depth of each pixel in a 2D image, playing a pivotal role in understanding the three-dimensional spatial construction of scenes. Historically, depth maps were acquired using high-end sensors, but their expensive cost and scene limitations hindered widespread adoption. Consequently, extracting depth information from 2D images using monocular cameras has become a research hotspot.

With the advancement of convolutional networks, backpropagation learning of features has replaced early handcrafted features. Eigen et al. [8] pioneered the use of neural networks for depth feature learning, setting the stage for the rapid development of monocular depth estimation. Hu et al. [13] built upon CNNs and employed L1 loss, gradient loss, and normal loss to address depth map boundary distortions. Bhat et al. [4] utilized EfficientNet [25] as the encoder and introduced Transformer modules during decoding to enhance global feature correlations, achieving state-of-the-art results. With the success of ViT [7] in image classification, recent works have explored replacing CNNs with Transformers for feature extraction. Yang et al. [29] combined CNNs with Transformers, merging local information from CNNs and global insights from Transformers, offering a novel perspective on global-local information aggregation. Yuan et al. [31] adopted the Swin-Transformer [16] as the encoder and leveraged the Transformer's self-attention mechanism with CRF to integrate global information.

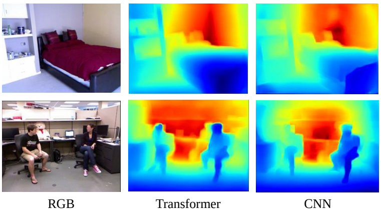


Figure 1: Visualization comparison of depth estimation using Transformers and CNNs. From left to right: RGB, depth prediction using a Transformer encoder, and depth prediction using CNN. Both results are obtained under identical data processing and loss conditions. Depth maps estimated by the Transformer method exhibit clearer scene structures than those by the CNN method, while CNNs provide smoother depth estimations at object boundaries.

While Transformers have significantly advanced monocular depth estimation, surpassing CNN models, the underlying reasons for their effectiveness are not fully understood. Investigating this will deepen our grasp of how Transformers process depth, advancing the field. Humans use visual cues like size-distance relationships and occlusions to gauge depth in images[15, 19, 26]. Similarly, this study explores what cues Transformers leverage and how these can be optimized to enhance performance. We utilize visualization techniques to probe whether Transformers and CNNs prioritize the same image regions and how these preferences affect outcomes, applying the visualization method developed by Hu et al. [14].

Our experiments demonstrate that Transformers are particularly sensitive to gradient information in images, especially gradients beneficial for scene depth information. However, Transformers lag behind CNNs in handling the continuity of depth gradients. Given these observations, we introduce the Depth Gradient Refinement (DGR) module, a plug-and-play component designed to enhance Transformer performance in depth estimation tasks. We also propose a novel loss function derived from optimal transport theory. Our findings show that these innovations markedly boost the performance of Transformer-based models, setting new benchmarks in the field. The main contributions of this work are:

- We provide a comprehensive comparison between Transformer and CNN models in monocular depth estimation through visualization, offering an interpretable analysis of their focal regions and operational principles.

- To address the challenges of depth gradient continuity in Transformers, we introduce the novel Depth Gradient Refinement (DGR) module. This paper also presents a unique perspective by treating depth maps as spatial probability distributions and employs optimal transport distance as a loss function for model optimization.

- Our proposed method, designed as a plug-and-play component, seamlessly integrates with existing Transformer-based monocular depth estimation models. When combined with leading Transformer-based models, our approach achieves breakthrough performance, surpassing existing benchmark.

2 Related Work

2.1 CNNs and Transformers for Monocular Depth Estimation

As neural networks have evolved, CNNs have become the backbone for depth estimation, object detection, and semantic segmentation due to their translational invariance [27] and robust feature representation. However, their limited receptive field has historically impeded long-range dependency modeling, a gap addressed by the introduction of Transformers. These networks utilize a sequence-based approach, granting them a global receptive field and making them increasingly dominant in image processing, often surpassing CNNs. Innovations such as the lightweight hybrid model by Zhang et al. [32], which integrates Consecutive Dilated Convolutions and Local-Global Features Interaction, and the convolution-free SwinDepth by Shim and Kim [23], along with Rahman et al.'s DwinFormer [17], reflect continuous advancements. Despite their advantages, Transformer-based models still often fall short of CNNs in fine-grained detail filtering [11].

2.2 Visualization in Deep Learning Models

Recent efforts have focused on understanding the mechanisms behind CNNs in image classification. Selvaraju et al. [20] used gradient-based techniques to analyze how changes in input affect model outputs. Techniques like Class Activation Mapping (CAM) and its enhanced version, Grad-CAM, have been instrumental, the latter integrating gradients to address CAM's limitations. However, studies on the interpretability of Transformers in the visual domain are sparse. Abnar et al. [1] explored linear relationships within attention mechanisms, constructing saliency maps via self-attention, though their approach does not adequately capture the impact on decision-making across different categories. Chefer et al. [5] applied deep Taylor decomposition to trace local correlations through layers, including attention and residuals.

Importantly, most of these methods cater to image classification and are not directly applicable to depth estimation, where the output is a 2D depth map rather than class probabilities. This discrepancy implies significant differences in how features are emphasized in depth estimation compared to classification. While Hu et al. [14] shed light on CNNs' selection of depth cues in images, the workings of Transformers in this context remain largely unexplored. Given that Transformers often excel over CNNs with the same data conditions, understanding these advantages is critical for advancing the field.

3 Methodology

We adopt the visualization approach proposed by Hu et al [14]. for the interpretability experiments in monocular depth estimation. The underlying assumption is that the Transformer network can extract depth information from a selected set of pixels. If a subset of pixels in an image can approximate the entire image with results within an acceptable range, we can identify the regions of interest for the network. By analyzing the commonalities among these regions, we can deduce the cues the Transformer network uses for depth prediction. Build-

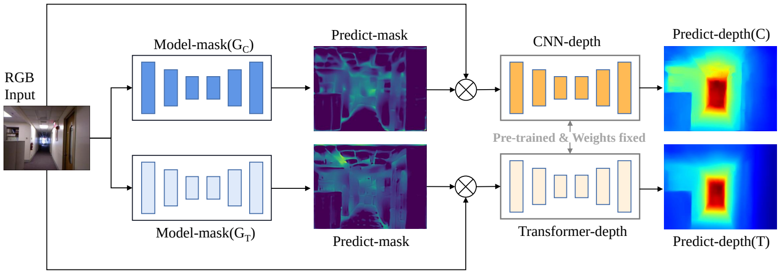


Figure 2: Network architecture visualizes Depth Estimation differences between CNN and Transformer models. This dual-pathway architecture processes RGB input through two parallel branches, each with a model-mask network (G_C and G_T) that produces predictive masks. These masks are multiplied element-wise with the RGB input to highlight relevant features. The modified inputs are then processed by pre-trained, weight-fixed depth prediction networks ("CNN-depth" and "Transformer-depth"), which independently generate depth maps (Predict-depth(C) and Predict-depth(T)). The model-mask networks are specifically trained to exclude irrelevant regions, ensuring that the predicted depth maps align more closely with the ground truths (GTs).

ing on this, we introduce the Depth Gradient Refinement (DGR) module and the Optimal Transport Depth Loss (OTDL).

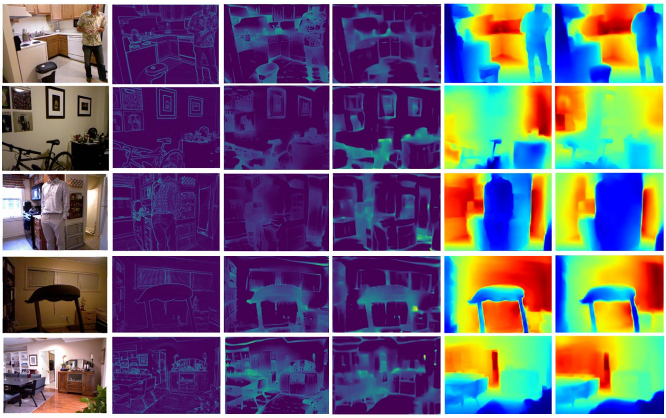


Figure 3: Visualization of differences. From left to right: RGB, image gradient, masks predicted by Transformer and CNN, and depth predictions post-sparsification.

3.1 Visualization of Monocular Depth Estimation

The experimental network's architecture is depicted in Figure 2, utilizing two networks, N_t (Transformer) and N_c (CNN), to predict depth maps Y_t and Y_c from an input image I. During the training phase of both networks, we employ consistent data processing techniques and utilize the same loss function. We use a sparse pixel mask M to determine focal areas by partially occluding I. The depth predictions $\hat{y}_t$ and $\hat{y}_c$ from these occluded inputs suggest important regions for each model.

$$\hat{y}_t = N_t(I * M_t) \tag{1}$$

$$\hat{y}_c = N_c(I * M_c) \tag{2}$$

Here, M_t and M_c are the sparse pixel selections for each network. The masks undergo optimization to minimize the loss L_{dif} between the full and sparse image depth predictions, incorporating $L1$ regularization to promote mask sparsity:

$$\min_{M} L_{dif}(Y, \hat{Y}) + \lambda \frac{1}{n} ||M||_1 \tag{3}$$

where L_{dif} computes the loss between the depth predictions obtained from the full image and those from the sparse image. n denotes the total number of sparse pixels, $||M||_1$ represents the $L1$ regularization of M, and λ is a hyperparameter used to control the sparsity level.

We employ two additional networks, G_t and G_c, to predict two sets of sparse pixels, $mask_t$ and $mask_c$, respectively. More specifically, we consider the following optimization:

$$\min_{G} L_{dif}(Y, N(I * G(I))) + \lambda \frac{1}{n} ||G(I)||_1 \tag{4}$$

For network G, we limit its output range to between 0 and 1 by using a sigmoid activation function. Consequently, $I * G(I)$ symbolizes a weighted selection of the input image I. This method allows the network to assign lower weights to less critical regions while emphasizing areas considered important.

To evaluate focus areas of CNN and Transformer models, we input $I * mask_c$ and $I * mask_t$—regions selected by CNN and Transformer, respectively—into the opposite models. We calculate the Root Mean Square Error (RMSE) for $Y_t = N_t(I * mask_c)$ and $Y_c = N_c(I * mask_t)$. Stable RMSE values indicate similar focal regions, while notable discrepancies suggest differing areas of interest. Both networks utilize binarized masks, $mask_c$ and $mask_t$, with elements set to 0 or 1, resulting in a sparsified image. Through experimental comparisons, we investigate these regions to discern the similarities and differences between the models.

Extensive testing on the NYU-depth-V2 dataset [24] clarifies how the Transformer differs from the CNN in feature extraction. As shown in Figure 3, the Transformer focuses notably on object boundaries and peripheral regions, which are crucial for depth perception in scenes. However, it struggles with the continuity of depth gradients compared to the CNN, often resulting in unnatural transitions in smoother regions of depth maps. Further details and analyses are available in the Appendix.

3.2 Depth Gradient Refinement Module

Transformer models excel in global context within monocular depth estimation but struggle with significant gradient changes at object edges, leading to less distinct boundary depth estimations. To address this, we introduce the Depth Gradient Refinement (DGR) module, enhancing depth continuity in transformers. We consider using higher-order derivatives to better capture the depth map's intricate variations, particularly where object boundaries show rapid intensity transitions. We compute the second and third-order derivatives of the depth map D as:

$$\nabla^2 D = D * \mathcal{L} \tag{5}$$

$$\nabla^3 D = D * \mathcal{H} \tag{6}$$

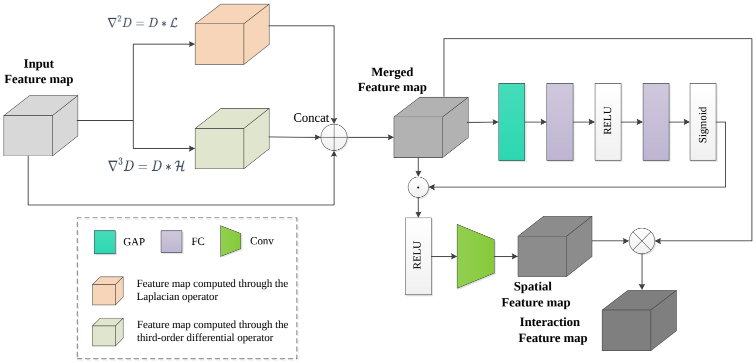

$\nabla^2 D = D * \mathcal{L}$
Input Feature map
$\nabla^3 D = D * \mathcal{H}$
Concat
Merged Feature map
RELU
Sigmoid
RELU
Spatial Feature map
Interaction Feature map
GAP
FC
Conv
Feature map computed through the Laplacian operator
Feature map computed through the third-order differential operator

Figure 4: Schematic of the Depth Gradient Refinement (DGR) module. Features processed by the transformer encoder serve as inputs to the DGR module, subsequently undergoing higher-order derivative computation, feature concatenation, and feature recalibration.

where $\mathcal{L}$ denotes the Laplacian operator, and $\mathcal{H}$ represents the corresponding third-order differential operator. The symbol “$*$" signifies the convolution operation.

Subsequently, leveraging feature concatenation, we amalgamate these higher-order derivative features with the original depth features, yielding a feature representation enriched with multifaceted depth information:

$$F_{\text{merged}} = \text{Concat}(\mathrm{F_D}, \nabla^2\mathrm{D}, \nabla^3\mathrm{D}) \tag{7}$$

Feature recalibration submodule is used to optimize the merged features for depth estimation tasks. We apply channel attention mechanism to weight features based on channel significance, enhancing spatial adaptivity to improve feature continuity. Furthermore, we include an outer product and dimensionality reduction to capture higher-order feature interactions. This approach ensures that information from both higher-order derivatives and original features is effectively utilized and amplified, providing a richer context for monocular depth estimation:

$$w_c = \sigma\left(\text{FC}_2\left(\text{ReLU}\left(\text{FC}_1\left(\text{GAP}(F_{\text{merged}})\right)\right)\right)\right) \tag{8}$$

$$F_{\text{spatial}} = \text{Conv}\left(\text{ReLU}\left(F_{\text{merged}} \odot w_c\right)\right) \tag{9}$$

$$F_{\text{interaction}} = \text{Conv}_{1\text{x}1}\left(F_{\text{spatial}} \otimes F_{\text{merged}}\right) \tag{10}$$

The DGR module is strategically placed after each encoder block in the Transformer. This arrangement supports progressive refinement of depth features while addressing potential feature discontinuities introduced by self-attention mechanisms and feed-forward networks within each block. By positioning the DGR module post each encoder block, we ensure maintained continuity across the network.

3.3 Optimal Transport Depth Loss

Transformers, devoid of local convolution operations, can occasionally produce depth maps with unnatural depth jumps in regions expected to be smooth. This observation propels the need for a more nuanced loss function that can address this continuity challenge. To bridge this gap, we introduce the Optimal Transport Depth Loss (OTDL). Drawing inspiration from the optimal transport theory, this loss offers a meticulous comparison between predicted and true depth maps, emphasizing the preservation of depth distribution variance and continuity.

To begin with, depth maps must be represented as normalized distributions. Given a predicted depth map P and its corresponding ground truth depth map Q, normalization is carried out as:

$$P' = \frac{P}{\sum_{i,j} P(i,j)} \tag{11}$$

$$Q' = \frac{Q}{\sum_{i,j} Q(i,j)} \tag{12}$$

where P' and Q' are interpreted as probability distributions.

Central to optimal transport is the cost matrix, which details the “expense" of transporting “mass" between positions. In the depth map context, the depth values provide this positional information. Thus, our cost matrix M has entries:

$$M_{ij} = |i-j|^2 \tag{13}$$

with i and j being depth values. The matrix entry M_{ij} denotes the cost of transitioning from depth i to depth j.

The core of our proposed loss rests on solving the optimal transport problem:

$$OT(P',Q') = \min_{T\in\Pi(P',Q')} \sum_{i,j} T_{ij} M_{ij} \tag{14}$$

where T signifies a joint distribution such that the marginals of T align with P' and Q'. The ensemble $\Pi(P',Q')$ contains all feasible T distributions that fulfill this criteria.

From the above discussions, the Optimal Transport Depth Loss (OTDL) is articulated as:

$$L_{\text{OTDL}}(P,Q) = OT(P',Q') \tag{15}$$

For monocular depth estimation, the conventional Mean Squared Error (MSE) loss, denoted as L_{MSE} , is typically employed:

$$L_{\text{MSE}}(P,Q) = \frac{1}{N}\sum_{i=1}^{N}(P_i - Q_i)^2 \tag{16}$$

To harness both the global depth estimation capability of MSE and the depth distribution preservation of OTDL, we combine them to formulate the final Loss:

$$L(P,Q) = L_{\text{MSE}}(P,Q) + \lambda_{OTDL} \cdot L_{\text{OTDL}}(P,Q) \tag{17}$$

where λ_{OTDL} is the hyperparameter that regulates the influence of the respective loss components.

4 Experiments

Implementation Details Our proposed method is implemented using the PyTorch framework on an RTX3090 GPU. We train our model for 200 epochs with a patch size of 256×256. The Adam optimizer is employed with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 1e^{-6}$. The weight decay factors for the encoder and decoder are set to 0.01 and 0, respectively. We adopt a polynomial decay for learning rate scheduling, starting with an initial rate of 10^{-4} and a power of $p = 0.9$, decaying until the rate reaches 10^{-5}. For the NYU-Depth-V2 dataset, the input/output resolution during training is set to 416×544.

Method	Venue	Abs Rel↓	RMS↓	$\text{Log}_{10}\downarrow$	$\delta_1\uparrow$
DORN [9]	CVPR'18	0.115	0.509	0.051	0.828
Yin et al. [30]	ICCV'19	0.108	0.416	0.048	0.872
Adabins [1]	CVPR'21	0.103	0.364	0.044	0.903
DPT [18]	ICCV'21	0.110	0.367	0.045	0.904
TransDepth [29]	ICCV'21	0.106	0.365	0.045	0.900
SwinDepth [6]	IEEE SENS J'21	0.100	0.354	0.042	0.909
DepthFormer [2]	ICIP'22	0.100	0.345	-	0.911
NeWCRFs [31]	CVPR'22	0.095	0.334	0.041	0.922
PixelFormer [3]	WACV'23	0.090	0.322	0.039	0.929
NDDepth [21]	ICCV'23	0.087	0.311	0.038	0.936
IEBins [22]	Arxiv'23	0.087	0.314	0.038	0.936
Adabins + DGR	Ours	0.097	0.347	0.041	0.918
DPT + DGR	Ours	0.104	0.348	0.042	0.914
TransDepth + DGR	Ours	0.101	0.348	0.043	0.911
SwinDepth + DGR	Ours	0.094	0.336	0.040	0.920
DepthFormer + DGR	Ours	0.096	0.329	-	0.922
PixelFormer + DGR	Ours	**0.086**	**0.310**	**0.036**	**0.937**

Table 1: Experimental results on NYU-Depth-V2. Bold text indicates the best performance.

Datasets We conduct training and visualization evaluations of Transformer and CNN on the NYU-Depth-V2 dataset [24] and assess the performance of our proposed method on this dataset. Then, we further evaluated the proposed method on the KITTI [10] dataset.

Models To delve into the differences, we select the ResNet50 [12] and SegFormer [28] network models as our target models. For a fair comparison, both models are implemented under identical data processing, training loss functions, and iteration cycles. For the loss function of the target networks, we employ the loss function proposed in [13] for training.

$$L_{dif} = L_{depth} + L_{grad} + L_{normal} \tag{18}$$

$$Loss = \frac{1}{n}\sum_i d_i^2 - \frac{1}{2n^2}\sum_i d_i^2 \tag{19}$$

where $L_{depth} = \frac{1}{n}\sum_{i=1}^{n} F(e_i)$, $L_{normal} = \frac{1}{n}\sum_{i=1}^{n}(1 - cos\partial_i)$, $L_{grad} = \frac{1}{n}\sum_{i=1}^{n}(F(\nabla_x(e_i)) + F(\nabla_y(e_i)))$, $F(e_i) = \ln(e_i + 0.5))$.

To assess the efficacy of DGR, we embed it into the state-of-the-art Transformer-based monocular depth estimation models for evaluation.

4.1 Quantitative results

We analyze the performance of DGR module across various Transformer-based state-of-the-art methods for monocular depth estimation and compare these results with prominent CNN-based SOTA models. Our experiments are conducted on two benchmark datasets: NYU-Depth-V2 and KITTI.

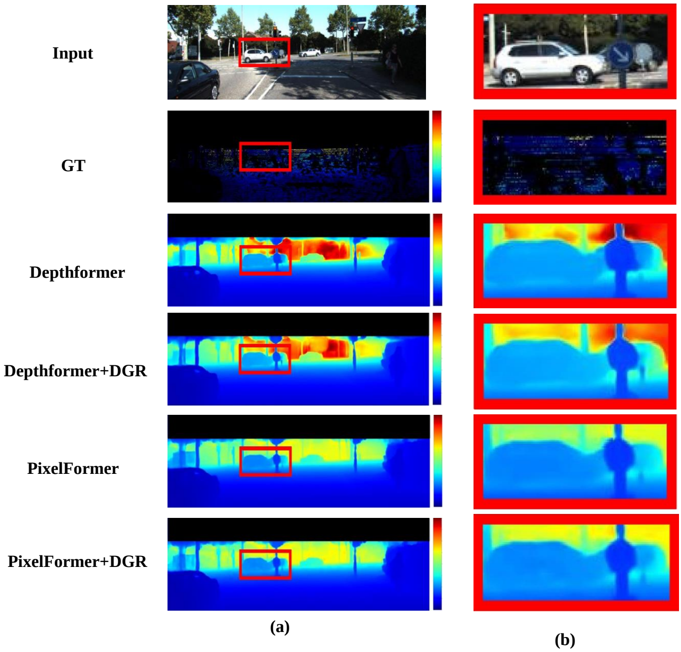
NYU-Depth-V2 Results Table 1 presents the comparative results on NYU-Depth-V2 dataset. Our proposed Adabins + DGR shows a significant improvement over the baseline Adabins model, reducing the Absolute Relative Difference (Abs Rel) from 0.103 to 0.097 and RMSE from 0.364 to 0.347. This improvement demonstrates the efficacy of the DGR module in refining depth predictions. Notably, PixelFormer integrated with DGR (PixelFormer+DGR) outperforms all other methods, achieving the best performance across most metrics, specifically lowering Abs Rel to 0.086 and RMSE to 0.310. This result underscores the compatibility of our DGR module with different transformer-based

Loss function		Abs Rel↓	RMS↓	$\text{Log}_{10}\downarrow$	$\delta_1\uparrow$
$L_{\text{MSE}}(P,Q)$	L_{OTDL}				
✓		0.088	0.320	0.038	0.931
	✓	0.088	0.319	0.037	0.934
✓	✓	**0.086**	**0.310**	**0.036**	**0.937**

Table 2: Performance of models trained with different loss functions on NYU-Depth-V2.

architectures, enhancing their depth estimation capabilities.

KITTI Results As shown in Table 3, on KITTI dataset, the Depthformer + DGR and PixelFormer + DGR again demonstrate superior performance, with Depthformer + DGR achieving the lowest Abs Rel of 0.050 and PixelFormer + DGR obtaining the best RMSE of 2.041. These results are particularly noteworthy given the challenging nature of KITTI dataset, known for its diverse and dynamic outdoor scenes. The consistent improvements across different base models when integrated with DGR highlight the module's adaptability and effectiveness in various contexts.



(a)

(b)

Figure 5: Comparative Visualization of Depth Estimation on KITTI Dataset with DGR Enhancement. (b) is a close-up view of the red frame in (a).

Loss Function We further evaluated the performance of the PixelFormer + DGR model trained with different loss functions, as shown in Table 2. When trained solely with the $L_{\mathrm{MSE}}(P,Q)$ loss function, the model achieved commendable performance on Abs Rel, RMS, and Log10, with a δ_1 accuracy of 0.931. This suggests that the mean squared error loss already provides a stable optimization target for the model, enabling accurate depth prediction in most cases. When trained solely with the L_{OTDL} loss function, the model's performance is similar to that trained only with the mean squared error loss, hinting at a potential complementary relationship between the two. When combining both loss functions, the model achieved the best results on all evaluation metrics.

4.2 Qualitative results

Figure 6 shows the visual comparison on NYU-Depth-V2 dataset. Examining the edge definition, the integration with the DGR module appears to enhance the edge smoothness, particularly around object boundaries. In addition, the detail preservation in areas of intricate

Method	Venue	Abs Rel↓	RMS↓	$\text{Log}_{10}\downarrow$	$\delta_1\uparrow$
DORN [9]	CVPR'18	0.072	2.727	0.120	0.932
Yin et al. [30]	ICCV'19	0.072	3.258	0.117	0.938
Adabins [1]	CVPR'21	0.058	2.360	0.088	0.964
DPT [18]	ICCV'21	0.062	2.573	0.092	0.959
TransDepth [29]	ICCV'21	0.064	2.755	0.098	0.956
SwinDepth [6]	IEEE SENS J'21	0.106	4.510	0.182	0.890
DepthFormer [2]	ICIP'22	0.052	2.143	0.079	0.975
NeWCRFs [31]	CVPR'22	0.052	2.129	0.079	0.974
PixelFormer [3]	WACV'23	0.051	2.081	0.077	0.976
NDDepth [21]	ICCV'23	0.050	**2.025**	0.075	0.978
IEBins [22]	Arxiv'23	0.051	2.370	0.076	0.974
Adabins + DGR	Ours	0.055	2.357	0.083	0.967
DPT + DGR	Ours	0.060	2.568	0.088	0.963
TransDepth + DGR	Ours	0.061	2.748	0.094	0.962
SwinDepth + DGR	Ours	0.098	4.485	0.151	0.894
DepthFormer + DGR	Ours	0.050	2.124	**0.074**	0.979
PixelFormer + DGR	Ours	**0.049**	2.041	0.075	**0.979**

Table 3: Experimental results on KITTI. Bold text indicates the best performance.

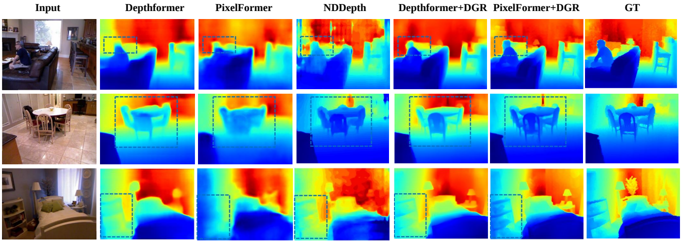


Figure 6: Qualitative comparison of different models on indoor dataset NYU-Depth-V2.

geometry, like the patterned chair backs and room corners, is visibly better in the DGR-enhanced models.

Figure 5 demonstrates the visualization results on the KITTI [10] dataset. The incorporation of the DGR module leads to more precise object boundaries and a clearer representation of the depth differences between objects at varying distances. This improvement allows for more accurate estimation of contours for distant objects like trees and people, as well as cars. In Figure 5(a), a red box highlights two signposts next to a white car, with one being farther away than the other. Figure 5(b) reveals that after adding the DGR module, the depth information of these two signposts is predicted more accurately.

5 Conclusion

In this study, we explored the application and challenges of the Transformer architecture in monocular depth estimation. Through visual comparisons with CNN models, we noted the Transformer's superior depth cue selection and identified issues like unnatural depth transitions in smooth regions. To improve Transformer performance, we introduced two novel approaches: the Depth Gradient Refinement (DGR) module and the Optimal Transport Depth Loss (OTDL). The DGR module uses higher-order derivatives to better capture variations at object edges, enhancing depth map continuity and edge sensitivity. The OTDL refines the loss function, focusing on preserving depth variance and continuity. Together with advanced Transformer-based models, our methods not only set new performance benchmarks but also provided deeper insights into the Transformer's mechanisms, laying a solid foundation for future research and applications in this field.

References

[1] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928*, 2020.

[2] Ashutosh Agarwal and Chetan Arora. Depthformer: Multiscale vision transformer for monocular depth estimation with global local information fusion. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 3873–3877. IEEE, 2022.

[3] Ashutosh Agarwal and Chetan Arora. Attention attention everywhere: Monocular depth prediction with skip attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5861–5870, 2023.

[4] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4009–4018, 2021.

[5] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 782–791, 2021.

[6] Zeyu Cheng, Yi Zhang, and Chengkai Tang. Swin-depth: Using transformers and multi-scale fusion for monocular-based depth estimation. *IEEE Sensors Journal*, 21 (23):26912–26920, 2021.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[8] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014.

[9] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2002–2011, 2018.

[10] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11): 1231–1237, 2013.

[11] Ali Hassani, Steven Walton, Nikhil Shah, Abulikemu Abuduweili, Jiachen Li, and Humphrey Shi. Escaping the big data paradigm with compact transformers. *arXiv preprint arXiv:2104.05704*, 2021.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] Junjie Hu, Mete Ozay, Yan Zhang, and Takayuki Okatani. Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries. In *2019 IEEE winter conference on applications of computer vision (WACV)*, pages 1043–1051. IEEE, 2019.

[14] Junjie Hu, Yan Zhang, and Takayuki Okatani. Visualization of convolutional neural networks for monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3869–3878, 2019.

[15] Pierre Lebreton, Alexander Raake, Marcus Barkowsky, and Patrick Le Callet. Measuring perceived depth in natural images and study of its relation with monocular and binocular depth cues. In *Stereoscopic Displays and Applications XXV*, volume 9011, pages 82–92. SPIE, 2014.

[16] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[17] Md Awsafur Rahman and Shaikh Anowarul Fattah. Dwinformer: Dual window transformers for end-to-end monocular depth estimation. *arXiv preprint arXiv:2303.02968*, 2023.

[18] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021.

[19] Ashutosh Saxena, Jamie Schulte, Andrew Y Ng, et al. Depth estimation using monocular and stereo cues. In *IJCAI*, volume 7, pages 2197–2203, 2007.

[20] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[21] Shuwei Shao, Zhongcai Pei, Weihai Chen, Xingming Wu, and Zhengguo Li. Nddepth: Normal-distance assisted monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7931–7940, 2023.

[22] Shuwei Shao, Zhongcai Pei, Xingming Wu, Zhong Liu, Weihai Chen, and Zhengguo Li. Iebins: Iterative elastic bins for monocular depth estimation. *arXiv preprint arXiv:2309.14137*, 2023.

[23] Dongseok Shim and H Jin Kim. Swindepth: Unsupervised depth estimation using monocular sequences via swin transformer and densely cascaded network. *arXiv preprint arXiv:2301.06715*, 2023.

[24] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*, pages 746–760. Springer, 2012.

[25] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[26] Antonio Torralba and Aude Oliva. Depth estimation from image structure. *IEEE Transactions on pattern analysis and machine intelligence*, 24(9):1226–1238, 2002.

[27] Zijie J Wang, Robert Turko, Omar Shaikh, Haekyu Park, Nilaksh Das, Fred Hohman, Minsuk Kahng, and Duen Horng Polo Chau. Cnn explainer: learning convolutional neural networks with interactive visualization. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1396–1406, 2020.

[28] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021.

[29] Guanglei Yang, Hao Tang, Mingli Ding, Nicu Sebe, and Elisa Ricci. Transformer-based attention networks for continuous pixel-wise prediction. In *Proceedings of the IEEE/CVF International Conference on Computer vision*, pages 16269–16279, 2021.

[30] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5684–5693, 2019.

[31] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. Neural window fully-connected crfs for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3916–3925, 2022.

[32] Ning Zhang, Francesco Nex, George Vosselman, and Norman Kerle. Lite-mono: A lightweight cnn and transformer architecture for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18537–18546, 2023.