Computational Visual Media
https://doi.org/10.1007/s41095-023-0350-8 Vol. 10, No. 3, June 2024, 505–522

Research Article

DepthGAN: GAN-based depth generation from semantic layouts

Yidi Li[1], Jun Xiao[1] (✉), Yiqun Wang[2] (✉), and Zhengda Lu[1]

Abstract Existing GAN-based generative methods are typically used for semantic image synthesis. We pose the question of whether GAN-based architectures can generate plausible depth maps and find that existing methods have difficulty in generating depth maps which reasonably represent 3D scene structure due to the lack of global geometric correlations. Thus, we propose DepthGAN, a novel method of generating a depth map using a semantic layout as input to aid construction, and manipulation of well-structured 3D scene point clouds. Specifically, we first build a feature generation model with a cascade of semantically-aware transformer blocks to obtain depth features with global structural information. For our semantically aware transformer block, we propose a mixed attention module and a semantically aware layer normalization module to better exploit semantic consistency for depth features generation. Moreover, we present a novel semantically weighted depth synthesis module, which generates adaptive depth intervals for the current scene. We generate the final depth map by using a weighted combination of semantically aware depth weights for different depth ranges. In this manner, we obtain a more accurate depth map. Extensive experiments on indoor and outdoor datasets demonstrate that DepthGAN achieves superior results both quantitatively and visually for the depth generation task.

Keywords depth map generation; generative model; transformer; scene generation

1 Introduction

With the rapid development of the technologies of computer vision and computer graphics, 3D scene generation has become important in a variety of downstream applications, such as virtual scene construction, AR, and VR, etc.

However, existing 3D generation methods mainly consider generating a single object, represented by point clouds [1, 2], voxels [3], meshes [4, 5], or implicit representations [6, 7]. Alternatively, they may optimize scene layout of retrieved 3D models for scene construction [8, 9]. The limited fitting capability of 3D generation methods and the complexity of object relations in 3D scenes make it extremely challenging to directly generate 3D representations of scenes containing diverse objects. Moreover, optimizing existing 3D models is computationally easier but lacks flexibility. Hence, generating complex 3D scenes still remains an open problem.

Compared to manually building 3D scenes with multiple objects, visual designers typically prefer controllable and simple input, such as 2D semantic layouts [10, 11] or sketches [12–14]. However, due to the lack of input information, it is impractical to straightly construct 3D scenes from such simplified 2D inputs as above. Inspired by work on depth estimation [15, 16], we believe the depth map to be a viable 2.5D medium: it measures the distance between the objects and the camera in stereoscopic space, and it can be regarded as providing a transition from 2D images to 3D scenes.

Therefore, we focus on a new task of generating an accurate and reasonable depth map from a simple semantic layout as input, to assist in constructing a 3D scene for visual designers. To the best of our knowledge, this is the first work to explore depth generation only using a semantic layout as input for

1 School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China. E-mail: Y. Li, liyidi19@mails.ucas.ac.cn; J. Xiao, xiaojun@ucas.ac.cn (✉); Z. Lu, luzhengda@ucas.ac.cn.

2 College of Computer Science, Chongqing University, Chongqing, China. E-mail: yiqun.wang@cqu.edu.cn (✉).

Manuscript received: 2023-02-03; accepted: 2023-04-11

constructing a 3D scene. Given camera parameters, a 3D scene can be precisely constructed once a reasonable depth map has been generated, as shown in Fig. 1. Since the depth map provides accurate geometric relationships, the 3D scene can be fully represented within this lower-dimensional space.

For this purpose, we first considered generating depths using previous Conv-based conditional image generation models, but this gave unsatisfactory results, which included incorrect depth intervals and improper depth structures. The receptive field of the convolution architecture is limited to a local scope [17] and feature aggregation is confined to pixels inside the scope. Hence, most existing Conv-based methods for depth generation cannot accurately predict global geometric correlations between different objects, making the generated depths visually incoherent. Furthermore, existing GAN-based conditional generative models adopt a simple structure for convolution layers and nonlinear activation to obtain the output image from generated high-resolution features. Since depth maps have more structural regularity than color images, such a simple layer cannot fully model the depth distribution, leading to stretched or squeezed depth intervals and less-than-smooth depth maps.

Accordingly, to address the limitations above, we propose DepthGAN, which redefines depth generation as a feature generation and depth synthesis task. In the feature generation part, to better generate global features in the semantically-guided generation, we propose a semantically aware transformer block

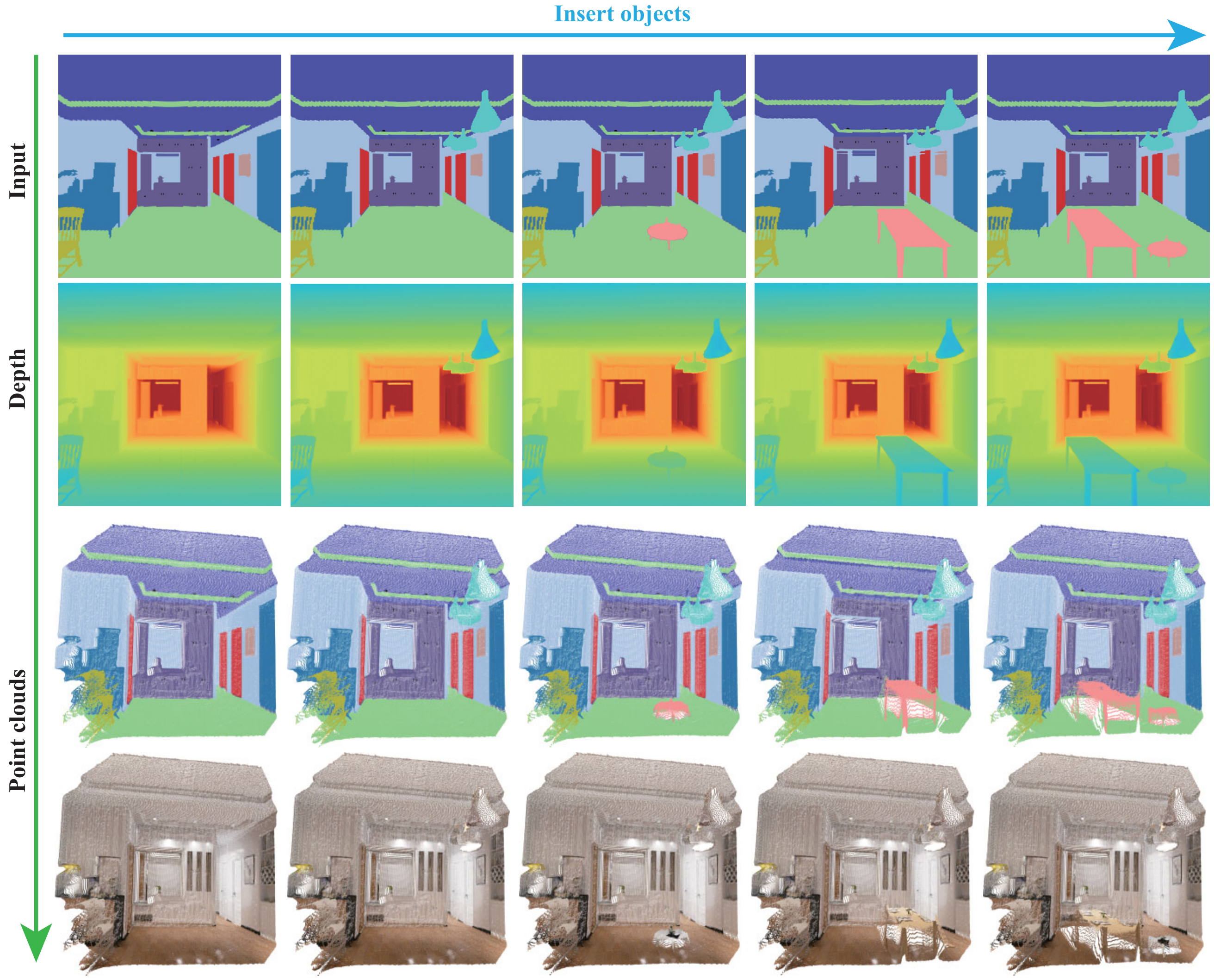


Fig. 1 We manipulate an input semantic layout (top row), to generate different depth maps (second row; blue is closer to the viewer and red is further). Given a fixed camera in the center of the room and the corresponding appearances of the scene and edited objects, we further construct point clouds from the generated depth maps, coloring them either with class labels (third row) or visual appearances (bottom row).

with mixed attention and semantically aware layer normalization, efficiently improving the consistency between the generated feature and the semantic input. Furthermore, we replace the output layer in previous generative methods by a semantically weighted depth synthesis module to generate an accurate depth map. We first predict the depth intervals within a scene and then synthesize the final depth map via weighted combination to integrate local and global features with semantic information. Equipped with the above modules, the proposed DepthGAN achieves state-of-the-art results on both indoor and outdoor scene datasets, demonstrating the effectiveness of our approach to depth generation. Our DepthGAN furthermore permits scene manipulation via simple modification of the input, as shown in Fig. 1. We generate the appearance using novel semantically aware transformer blocks together with depth generation. Thus, we can generate scenes from simply handcrafted semantic input layout, as shown in our video in the Electronic Supplementary Material (ESM).

Overall, our contributions are in summary:

- a novel generation–synthesis approach for a depth generation task which uses only semantic layout as input; our approach provides an effective and controllable solution for 3D scene generation,
- a semantically aware transformer block with mixed attention and semantically aware layer normalization to take advantage of rich global information about depth and semantic layout for generating depth features, and
- a semantically weighted depth synthesis scheme to generate the final depth map using as input generated depth features; it provides superior quantitative results and visual effects.

2 Related works

2.1 GAN-based semantic image synthesis

Generative adversarial networks [18] have achieved impressive results in unconditional [19–21] and conditional [22–24] image generation. Semantic image synthesis is a task that takes a semantic layout as input, which provides pixel-level class labels, and outputs a natural image with semantic guidance.

Pix2Pix [22] was first to introduce an encoder–decoder architecture and a patch-based discriminator to handle this problem. SPADE proposed a method to modulate the activations in the normalization layers using the semantic input to guide generation, which frees the encoder block, enabling coarse-to-fine generation. Later works including Refs. [25–27] learned normalization layers using style, semantic, or instance input. CC-FPSE [28] predicted conditional separated convolution kernels from the input semantic layout, and introduced a feature pyramid semantic-embedding discriminator for semantic alignment. OASIS [29] re-designed the discriminator with a semantic segmentation network for semantic alignment. LGGAN [30, 31] proposed a local class-specific and a global image-level generator to learn local–global feature generation. SCGAN [32] learned semantic vectors to parameterize spatially conditional convolution and normalization. As depth generation requires greater global feature awareness, we introduce a cascade of transformer-based blocks for coarse-to-fine depth feature generation.

2.2 Monocular depth estimation

A depth map measures the spatial structure of a scene, a low-dimensional but efficient representation of the 3D scene. Monocular depth estimation [33–36], mainly focuses on regressing dense depth maps from images. Poor edge quality and lack of global information are common problems of CNN-based depth estimation models. Ref. [37] explicitly introduced a pre-trained semantic segmentation network to guide depth boundaries, using the high quality of edges in the semantic map. In addition to CNN-based structures, generative models [38, 39] and vision transformers [40] have also been applied to depth estimation tasks. Recently, Refs. [41, 42] performed a global statistical analysis on depth bins to further predict the depth map in a classification–regression manner.

Using the output of depth estimation, a dense depth map, we may reconstruct a 3D scene. However, a color image is typically used as input to create this dense depth map. It difficult to meet the requirements of visual designers for simple, easily modified input using such an approach. Instead, in our new approach, we only use a semantic layout as input.

2.3 Vision transformers

The seminal work [43] proposed a pure transformer [44]-based architecture for discriminative vision tasks; it enables global feature aggregation and extraction from images. CvT [45] introduced convolutions into vision transformers to enhance local attention. Swin transformer combines local and global attention by calculating attention in a local shifting window, leading to a huge improvement in vision transformers. Recently, researchers begin to explore using vision transformers in GANs as a means to generating better global features in complex images. Refs. [46, 47] have rapidly improved image generation tasks due to the superior global feature aggregation capability of multi-head self-attention blocks (MSAs). However, the generative quality of these methods is not proportional to the time taken due to the quadratic complexity by default of vision transformers, which makes high-resolution generation difficult. Recent works [48–50] have proposed calculating MSAs in local windows, leading to linear computational complexity. Ref. [51] demonstrated the feasibility of using block-wise attention for unconditional high-resolution image generation. In this work, we observe that exploiting vision transformers with more global information suits the new conditional depth generation task.

3 Method

Figure 2 presents our novel depth generation architecture, DepthGAN, which consists of a depth feature generation stage (see Section 3.1) and a depth map synthesis stage (see Section 3.2). Starting from a one-hot semantic layout $S \in \mathbb{N}^{H\times W\times C}$, we first adopt a cascade of semantically aware transformer blocks to generate the depth feature $F \in \mathbb{R}^{H\times W\times E}$. Then we utilize F to generate the adaptive depth interval and apply a semantically-weighted combination to obtain the final depth map $D \in \mathbb{R}^{H\times W}$, which is semantically aligned with the semantic layout S.

3.1 Depth feature generation

3.1.1 *Approach*

Unlike appearance generation, depth map generation mainly focuses on global features, particularly the geometric and spatial structure within the scene. In order to capture global information, we construct an architecture comprising a series of Swin transformer [49] blocks as our baseline to better generate global attention features. It takes the downsampled low-resolution semantic layout as its input and generates the depth feature using upsampling in a coarse-to-fine manner.

However, the baseline method cannot effectively align the generated features with the input semantic layout due to the lack of semantic constraints in the generation process. To address this issue, we propose a semantically aware transformer (SAT) block, which introduces a semantic positional encoding (SPE), a mixed attention module, and a semantically aware layer normalization (SALN) module to guide feature generation, as shown in Fig. 3(a).

3.1.2 *Semantic positional encoding*

In the SAT block, we first aim to better inform the layers about the semantic position information at each input scale. Thus, we utilize a learned semantic embedding from the semantic layout as a positional encoding, as shown in Fig. 3(a).

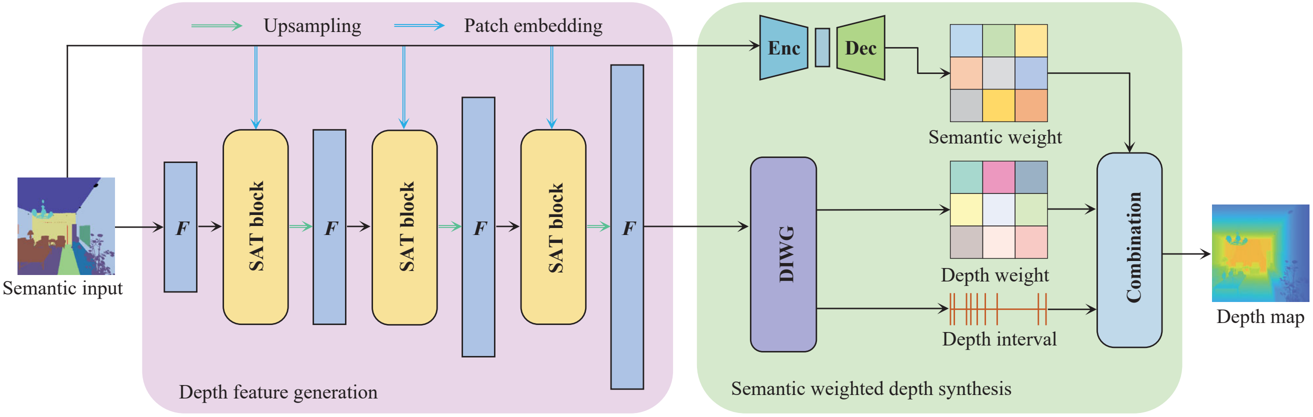


Fig. 2 DepthGAN. Our framework has two stages: (i) depth feature generation and (ii) semantically weighted depth synthesis. A cascade of semantically aware transformer (SAT) blocks generates depth features with semantic alignment. An encoder–decoder then generates the semantic weight map, and a DIWG module generates the depth interval and the depth weight map. Finally, the depth map is synthesized through a semantically weighted combination.

For the input feature maps F^* of shape $\mathbb{R}^{H_\mathrm{F}\times W_\mathrm{F}\times E_\mathrm{F}}$ at each input scale, where $E_\mathrm{F}, H_\mathrm{F}, W_\mathrm{F}$ are the spatial resolution, we embed the one-hot semantic input S of shape $\mathbb{N}^{H\times W\times C}$ at the same scale as F^* by learned *convolution kernels* with different stride parameters, denoted S^*:

$$S^* = \mathrm{Conv}(S) \tag{1}$$

Thus the SPE adapts to different feature scales. We then add the embedded semantic input S^* to the input feature maps F^*, enabling the SAT block to perceive global semantic information at each pixel. Unlike the learned positional encoding in standard transformer blocks which encodes the relative positions of pixels, our SPE can incorporate semantic information and thus improve feature generation quality and semantic alignment.

3.1.3 Mixed attention

Although the baseline utilizes self-attention by calculating queries, keys, and values from the features, this method ignores the interaction between features and semantics. To address this problem, we propose a simple yet effective strategy, *mixed attention*, as shown in Fig. 3(b). Instead of calculating the attention between tokens of features, we adopt additional semantic queries:

$$\mathrm{MixedAttn} = \mathrm{Softmax}\left(\frac{(Q_\mathrm{F}+Q_\mathrm{S})K_\mathrm{F}^\mathrm{T}}{\sqrt{d_k}} + E\right)V_\mathrm{F} \tag{2}$$

where $Q_\mathrm{F}, K_\mathrm{F}, V_\mathrm{F}$ represent the query, key, and value matrices projected by the features, and Q_S is the query matrix from the semantic input. The relative positional encoding E is added as a bias term.

Unlike self-attention in Swin transformer, our mixed attention enables feature aggregation between features and semantics at the same time, leading to more semantically aware outputs.

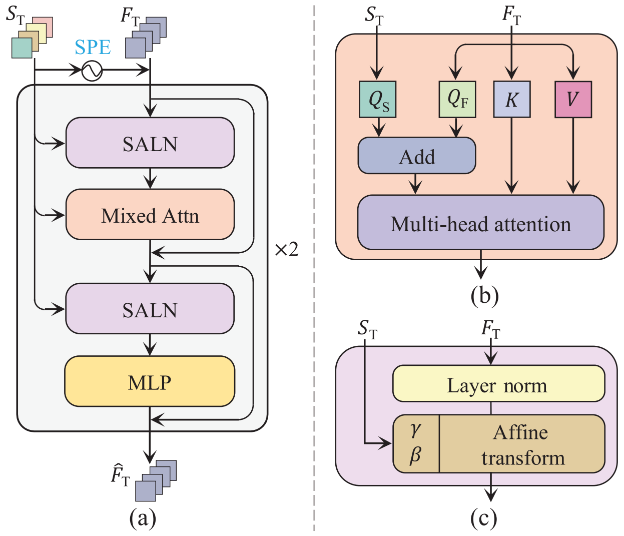


Fig. 3 (a) Semantically aware transformer (SAT) block. (b) Mixed attention block. (c) Semantically aware layer normalization (SALN).

3.1.4 Semantic-aware layer normalization

To better match semantic features and depth features in the SAT block, we propose *semantically aware layer normalization* to learn a parameterized affine transformation and fuse the semantic information with the features, as shown in Fig. 3(c). Given the input feature tokens F_T, the output tokens $\hat{F}_\mathrm{T}$ are calculated as

$$\hat{F}_\mathrm{T} = \frac{\gamma(S_\mathrm{T})}{\sigma} \odot (F_\mathrm{T} - \mu) + \beta(S_\mathrm{T}) \tag{3}$$

where γ, β are vectors learned by a simple MLP–ReLU–MLP architecture with semantic tokens S_T. Here $\odot$ denotes element-wise multiplication between two vectors; μ and σ denote the mean and standard deviation of F_T, respectively.

With learned scaling and bias vectors γ, β, the affine transformation adapts to the semantic input and varies with respect to different token positions, facilitating the matching of different features while training remains stable.

3.2 Semantic weighted depth synthesis

A depth map typically has structural regularity in its feature distribution. However, the simple output approach of previous GAN-based image generation methods lacks the capability to model an accurate depth map. Inspired by Adabins from the depth estimation task, we propose a *semantically weighted depth synthesis* (SWDS) stage, which generates a depth interval from the depth features of the previous stage and performs semantically weighted depth synthesis to obtain the final depth map, as shown in Fig. 2.

In this stage, we first use a *depth interval and weight generation* (DIWG) module to enable depth interval and weight generation for the scene. Meanwhile, we use an encoder–decoder architecture to compute a semantic weight map W from S, to better utilize the semantic input. Finally, we utilize a semantically weighted combination module to fuse them, to synthesize the final depth map.

As Fig. 4 shows, the DIWG module first embeds the input feature F and the semantic input S into patches,

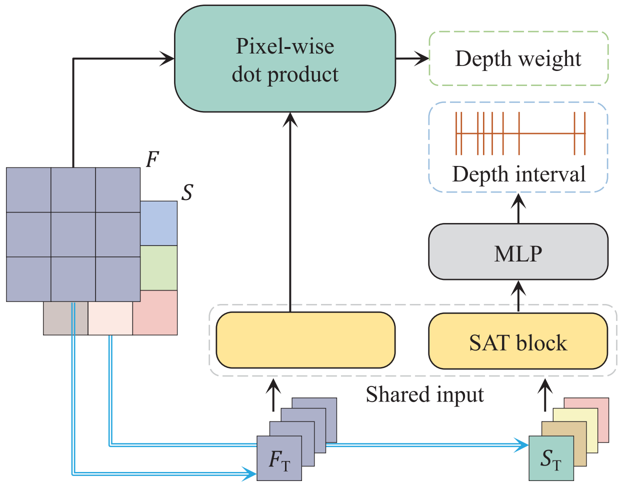


Fig. 4 Depth interval and weight generation module. The module takes the feature map and semantic map as input and outputs the depth interval and depth weight map.

denoted F_T and S_T. Then we adopt two SAT blocks to enable semantically aware feature generation. Note that we do not add a global positional encoding here, since the window size in the SAT block is set to be the same as in the embedded feature map, so the relative positional encoding here can be regarded as a global one. The output embedding from each SAT blocks is projected by a linear perceptron with Softmax to yield an N-bin length vector b. As in Adabins, the bin centers $c(b)$ are calculated via a post-process:

$$c(b_i) = d_{\min} + (d_{\max} - d_{\min}) \left(\frac{b_i}{2} + \sum_{j=1}^{i-1} b_j \right) \tag{4}$$

where $c(b_i)$ is the center depth of the ith bin. $d_{\max}$ and $d_{\min}$ are the maximum and the minimum depths in the dataset, respectively.

We obtain the depth weight map via a pixel-wise dot product between the generated feature embedding of another SAT block and the input feature F. Note that the depth weight map contains rich local–global feature similarities while serving for a key–query process.

On the other hand, we compute the semantic weight map W with the encoder–decoder architecture. We then apply element-wise multiplication between W and the depth weight map to obtain a semantically aware depth weighted map, which aggregates additional semantic information for weighted generation. Next, the semantically aware depth weighted map is converted to a weighted probability depth distribution map P^W using Softmax. Finally, the depth value for each pixel is calculated by weighted combination with the corresponding probability distribution: $\hat{d} = \sum_i c(b_i) p_i^W$.

In the SWDS stage, we fuse the semantic information with the depth map and disentangle bin generation from depth weight map generation using two separate SAT blocks, enabling more accurate and reasonable depth maps.

3.3 Loss functions

The generator and the discriminator are trained alternatively, adopting hinge loss in the discriminator for distinguishing real from fake. The generator is optimized by multiple losses, including hinge-based adversarial loss, discriminator feature matching loss $L_{FM}(\hat{x}, x)$, and perceptual loss $L_P(\hat{x}, x)$, following Ref. [52]:

$$\begin{cases} L_D = -\mathbb{E}_{x,S}[H(D(x,S))] - \mathbb{E}_{\hat{x},S}[H(D(\hat{x},S))] \\ L_G = -\mathbb{E}_{\hat{x},S}[D(\hat{x},S)] + \lambda_P \mathbb{E}_{\hat{x},S} L_P(\hat{x},x) \\ \qquad + \lambda_{FM} \mathbb{E}_{\hat{x},S} L_{FM}(\hat{x},S) \end{cases} \tag{5}$$

where x is a real depth map, $\hat{x}$ is a generated depth map, and S is the semantic layout. λ_P, λ_{FM} denote weights for perceptual loss and feature matching loss, respectively. H is the hinge function; $\lambda = 1$ if I is a real image and -1 if I is a generated image:

$$H(I) = \min(0, -1 + \lambda I) \tag{6}$$

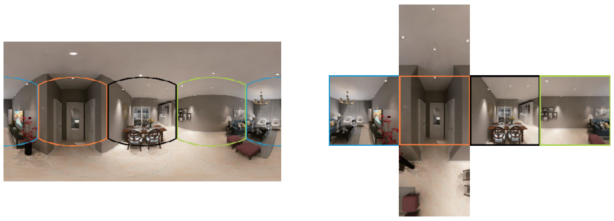
4 Experiments

4.1 Setting

4.1.1 Datasets

We benchmark our approach using Structured3D [53], Stanford2D3D [54], and Visual KITTI (VKITTI) [55] datasets, which are detailed below:

- *Structured3D* contains synthetic scenes rendered as panoramic images. The geometric structure is distorted in the panoramic images on the sphere grid, and accurate depth generation is difficult using conventional convolution kernels [56]. Therefore, we re-project the panoramic images in Structured3D to perspective views by reverse gnomonic projection [57], as shown in Fig. 5. Following the recommended split, we use scenes 0–2999 for training, scenes 3000–3249 for validation, and scenes 3250–3499 for testing, giving 109,494 training images and 10,122 testing images of virtual indoor scenes.



(a) Panorama (b) CubeMap

Fig. 5 Re-projection of Structured3D images. (a) Panoramic image on a sphere grid. (b) Cubemap produced by reverse gnomonic projection. Colored boxes in (a, b) highlight the same part of the scene as spherical and perspective views.

- *Stanford2D3D* contains real-world scenes scanned with RGB-D cameras, shown as both perspective and panorama images. Following the recommended split, we chose perspective images in areas 1–4 and 6 for training and area 5 for testing. Stanford2D3D contains 52,093 training images and 17,593 testing images of real-world indoor scenes.
- *VKITTI* is a photo-realistic synthetic video dataset designed to learn and evaluate computer vision models for several video understanding tasks: object detection and multi-object tracking, scene-level and instance-level semantic segmentation, optical flow, and depth estimation. We chose scenes 0, 2, 18, and 20 for training and scene 6 for testing, giving 18,560 training images and 2700 testing images of outdoor scenes. Semantic labels were obtained from the provided instance labels by the color mapping in each scene provided in the dataset.

The minimum depth value was set to 0, while the maximum depth value was 655.35 m for VKITTI, and 10 m for the other indoor datasets.

4.1.2 Evaluation metrics

We adopt Fréchet Inception Distance (FID) [58] to measure the Wasserstein-2 distance between the distribution of generated depth maps and corresponding ground truth. Seven standard metrics for depth estimation tasks [59] were evaluated for depth accuracy, including mean absolute error (MAE), root mean square error (RMSE), absolute relative error (AbsRel), square relative error (SqRel) and threshold percentage (δ^n):

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|d_i - \hat{d}_i|$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|d_i - \hat{d}_i|^2}$$

$$\mathrm{AbsRel} = \frac{1}{N}\sum_{i=1}^{N}|d_i - \hat{d}_i|/\hat{d}_i$$

$$\mathrm{SqRel} = \frac{1}{N}\sum_{i=1}^{N}|d_i^2 - \hat{d}_i^2|/\hat{d}_i$$

$$\delta_n = \text{percentage of pixels satisfying}$$
$$\max(d_i/\hat{d}_i, \hat{d}_i/d_i) < 1.25^n, n \in \{1,2,3\}$$

where d and $\hat{d}$ are ground truth depth and generated depth respectively. Additionally, we calculate the PSNR of the generated depth maps:

$$\mathrm{PSNR} = 20\log_{10}\frac{\mathrm{MAX}_d}{\mathrm{RMSE}} \tag{7}$$

where MAX_d is the maximum depth value of the dataset used for depth generation.

4.1.3 Training and testing details

Different from the task of image generation, the depth values are stored as 32-bit float values, which are then normalized to [0, 255.0] by dividing by the maximum scene depth value. During testing, FID is calculated directly from the normalized depth values ranging from 0 to 255.0, while other metrics are calculated by re-scaling the generated float depth values back to the original format of the dataset without loss of accuracy.

For the discriminator, we apply the Spectral Norm to all layers. We adopt the Adam optimizer [60] with learning rate 0.0001 for the generator and 0.0004 for the discriminator following TTUR [58], and set $\beta_1 = 0$ and $\beta_2 = 0.999$. The weight for the perceptual loss is 10. Our models were trained on 8 TITAN RTX 24 GB GPUs, with a batch size of 32. The training and generated resolution is 256×256 for Structured3D and Stanford2D3D datasets, and 256 × 512 for the VKITTI dataset. All results presented are obtained after training for 50 epochs. During inferencing, it takes 0.067 s to generate a sample.

4.1.4 Network architecture

The detailed model architecture for 256 × 256 resolution depth generation is as shown in Table 1.

4.2 Comparisons

In this section, we provide quantitative and visual comparisons to demonstrate the effectiveness of DepthGAN. We compare it to previous semantic image synthesis methods including Pix2pixHD,

Table 1 Architecture of DepthGAN. Input size and Dim are the shapes of the input feature map and semantic embedding in the SAT block. SAT-8 means an SAT block with input resolution 8 × 8. In the SAT block, h is the number of heads in MSAs, d is the depth of the SAT blocks, and w is the window size for mixed attention. In the SWDS module, p is the patch size of the patch embedding for both the feature map and the semantic layout, and MLP-256 is the 256-dimensional MLP for the depth interval. We use bilinear upsampling for the upsampling layers

Input size	Dim	Module	Architecture	Up
8 × 8	512	SAT-8	{h-16, d-2, w-8}	✓
16 × 16	512	SAT-16	{h-16, d-2, w-8}	✓
32 × 32	512	SAT-32	{h-16, d-2, w-8}	✓
64 × 64	256	SAT-64	{h-16, d-2, w-8}	✓
128 × 128	128	SAT-128	{h-8, d-2, w-8}	✓
256 × 256	64	SAT-256	{h-4, d-2, w-8}	
256 × 256	64	SWDS	p-16 SAT-16, MLP-256 SAT-16	

SPADE, CC-FPSE, LG-GAN, SEAN, OASIS, and SAFM using the same training strategy. For OASIS, we use the authors' default setting without 3D noise to avoid randomness in the generated depth map, to improve accuracy.

Note that in the depth generation, we use a one-hot semantic layout as input for accurate evaluation, unlike the input noise map commonly used in semantic image synthesis tasks for multi-style generation. Thus we can learn a unique depth distribution from an input semantic layout without randomness. Furthermore, using a semantic layout instead of noise as input permits fully controllable depth generation. As shown in Fig. 1 and the video in the ESM, the generated depth map only changes the edited objects, while remaining parts remain unchanged.

4.2.1 Quantitative comparison

Table 2 compares metrics for the depth maps generated by our approach and its competitors on the proposed new tasks. Our approach provides a decisive improvement and performs consistently better than previous approaches, which demonstrates the effectiveness of the proposed approach. Using a generate–synthesize strategy, we generate depth maps with more accurate depth values and higher PSNR. In particular, our method improves MAE by around 20% and PSNR by around 5% over the second-best competitor, averaged over the three datasets: the proposed strategy is more capable of generating accurate depth than the convolution-nonlinear approach of previous methods. Moreover,

Table 2 Comparison to previous approaches on various datasets. Best results are in bold

Dataset	Method	MAE ↓	AbsRel ↓	SqRel ↓	RMSE ↓	δ^1 ↑	δ^2 ↑	δ^3 ↑	PSNR ↑	FID ↓
Structured3D	Pix2pixHD	0.1587	0.1325	0.1162	0.2062	84.52	93.73	96.64	21.63	128.20
	SPADE	0.1366	0.1447	0.0781	0.1536	86.61	94.51	96.93	22.42	119.59
	CC-FPSE	0.0946	0.0903	0.0353	0.1297	91.46	97.68	99.04	27.19	87.62
	LGGAN	0.1362	0.1229	0.0893	0.1489	88.03	95.63	97.56	23.41	114.02
	SEAN	0.1037	0.0863	0.0481	0.1268	89.47	97.05	98.75	26.69	75.34
	OASIS	0.1199	0.1173	0.0532	0.1631	87.47	95.83	98.12	24.40	166.51
	SAFM	0.0981	0.0826	0.0419	0.1187	90.64	97.45	98.61	27.79	61.58
	Ours	**0.0613**	**0.0590**	**0.0228**	**0.0888**	**95.37**	**98.67**	**99.40**	**30.53**	**37.38**
Stanford2D3D	Pix2pixHD	0.5424	0.2985	0.5507	0.7801	69.36	84.97	90.97	17.25	335.98
	SPADE	0.5820	0.2981	0.3662	0.7910	60.54	83.07	92.10	19.37	201.53
	CC-FPSE	0.3662	0.1822	0.1466	0.5385	76.66	92.70	97.20	23.88	185.87
	LGGAN	0.3381	0.1866	0.1435	0.5637	77.65	92.92	97.25	22.23	254.65
	SEAN	0.4208	0.2068	0.1883	0.6057	72.28	90.99	96.56	21.67	157.37
	OASIS	0.4037	0.2441	0.2635	0.6252	71.23	89.79	97.43	22.79	172.55
	SAFM	0.3788	0.1927	0.1604	0.5559	74.17	91.86	96.98	22.19	238.06
	Ours	**0.2831**	**0.1380**	**0.1168**	**0.4898**	**83.90**	**95.21**	**98.19**	**23.95**	**130.45**
VKITTI	Pix2pixHD	24.689	0.3989	31.784	53.373	52.47	81.16	92.92	21.74	668.24
	SPADE	20.014	0.3384	13.675	38.607	55.64	80.95	91.49	24.60	510.08
	CC-FPSE	18.760	0.2869	11.559	35.376	64.63	84.90	92.67	25.40	764.08
	LGGAN	15.089	0.2605	12.331	34.091	67.45	88.34	94.26	25.70	470.11
	SEAN	18.719	0.2996	16.393	38.919	66.52	84.48	91.21	24.55	569.56
	OASIS	13.214	0.2657	8.439	30.263	64.40	86.71	93.17	26.68	493.30
	SAFM	15.220	0.2548	10.754	31.702	64.17	87.15	93.42	26.40	454.41
	Ours	**10.973**	**0.2315**	**7.181**	**26.388**	**69.47**	**89.16**	**94.55**	**27.02**	**291.78**

our generated depth maps improve on the competitors by 28% on the FID score, showing that the distributions of generated depth maps are closer to the ground truth distributions.

4.2.2 Visual comparison

Figures 6–8 compellingly show the ability of our approach to generate more accurate depth maps with reasonable structure correlation, and a better match to the ground truth depth distribution. With the global depth features generated by the SAT block, our generated depth maps can better represent the structure of complex scenes, such as the chairs in the first scene of Fig. 7. Even for small and far semantic regions, our approach can still generate correct depth values: see the plants in the first scene in Fig. 6 and the door in the second scene in Fig. 7.

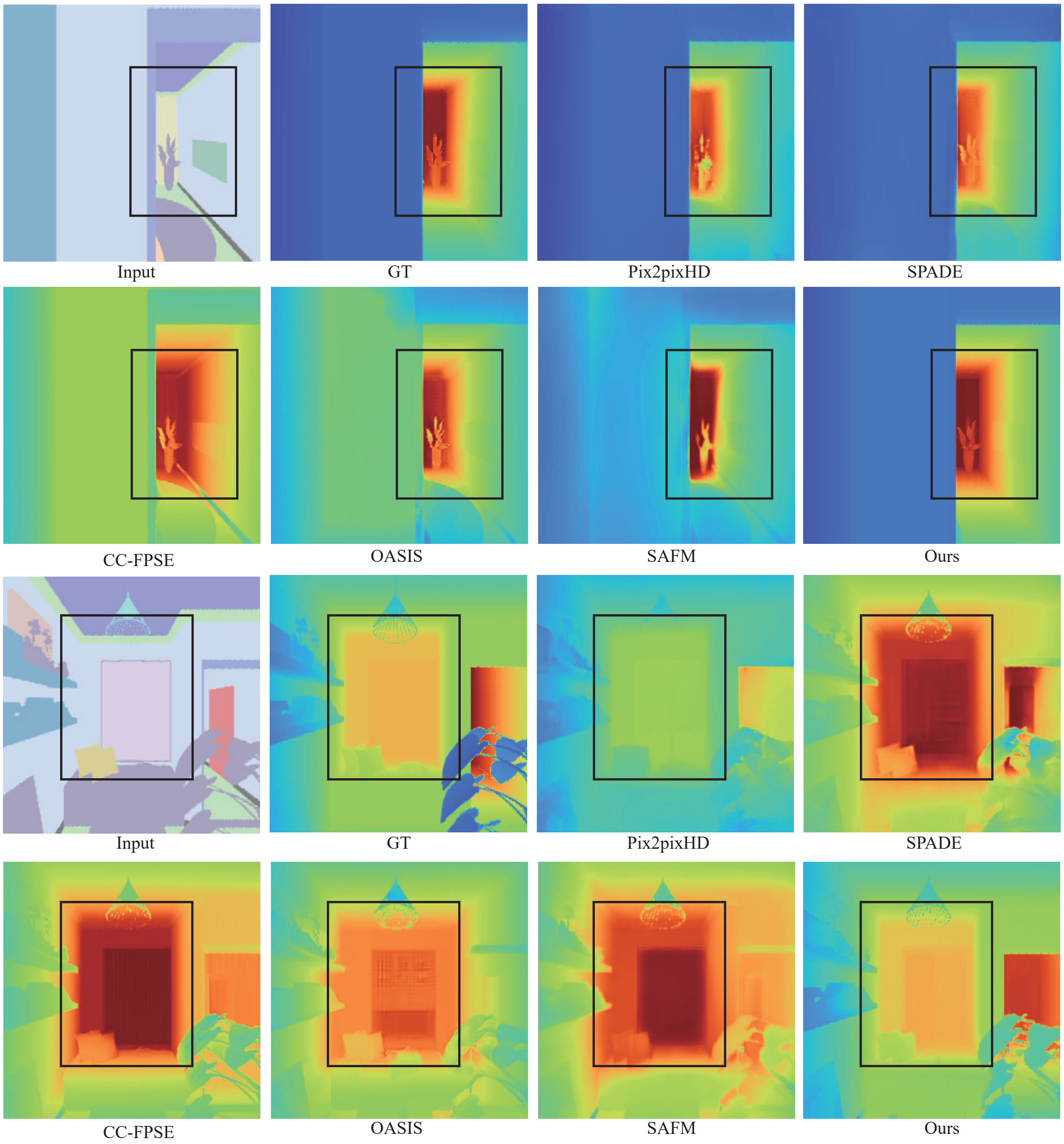


Fig. 6 Depth map comparisons, for two scenes from the Structured3D dataset, showing input, ground truth, and output from various methods. Blue is closer to and red further from the viewer.

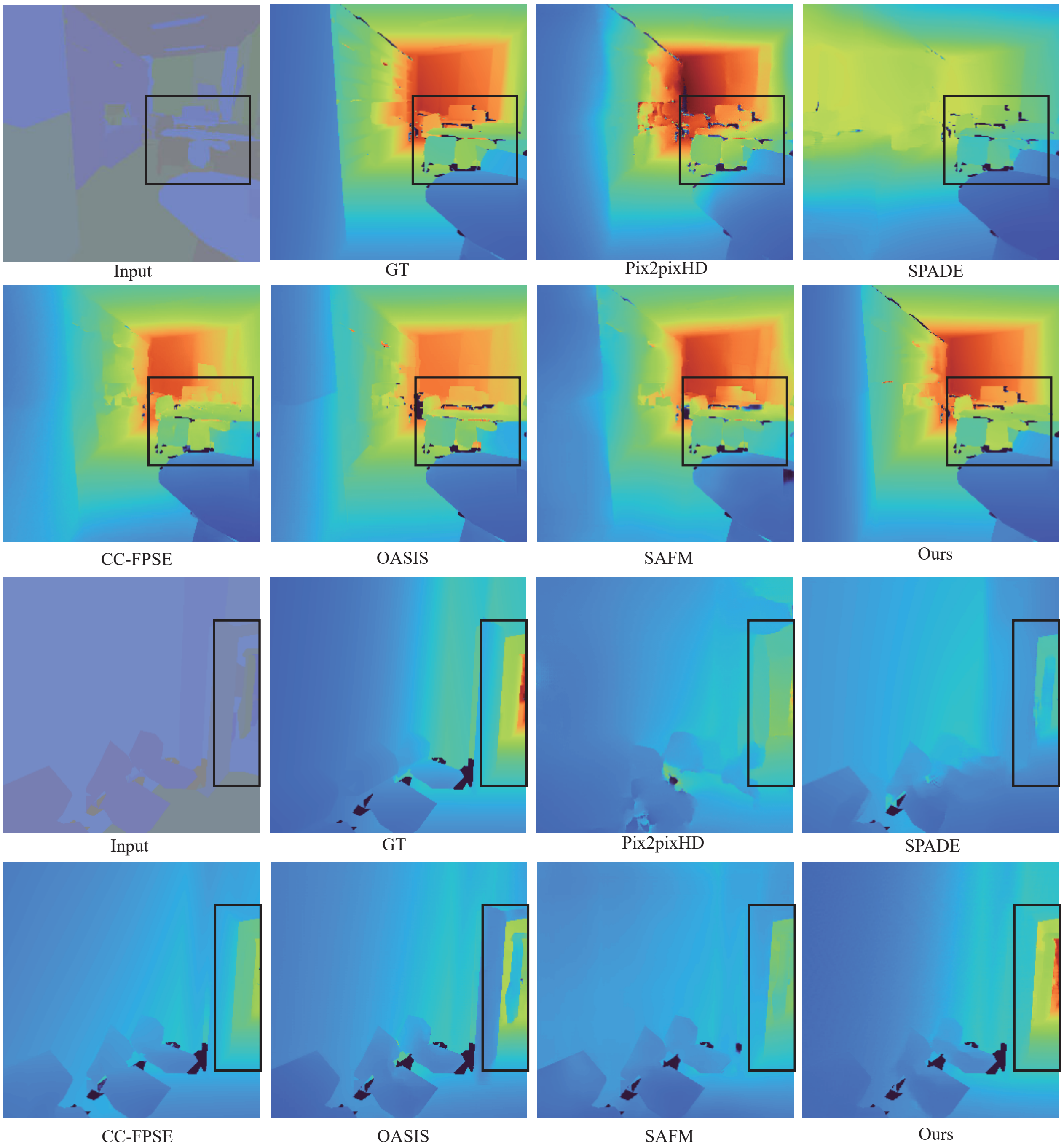


Fig. 7 Depth map comparisons for two scenes from the Stanford2D3D dataset, showing input, ground truth, and output from various methods. Blue is closer to and red further from the viewer.

Moreover, since we generate the depth interval for the scene and utilize a semantically aware weighted combination, our generated depth maps show more accurate geometric correlation and can better capture depth variation within the semantic region: see the second scene in Fig. 6 and the depths of trees in Fig. 8.

4.3 Ablation and alternatives

We conducted experiments on the Structured3D dataset to verify the effectiveness of each component of our method.

4.3.1 Overview

As shown in Table 3, starting from a cascade of Swin blocks as our baseline method, we gradually

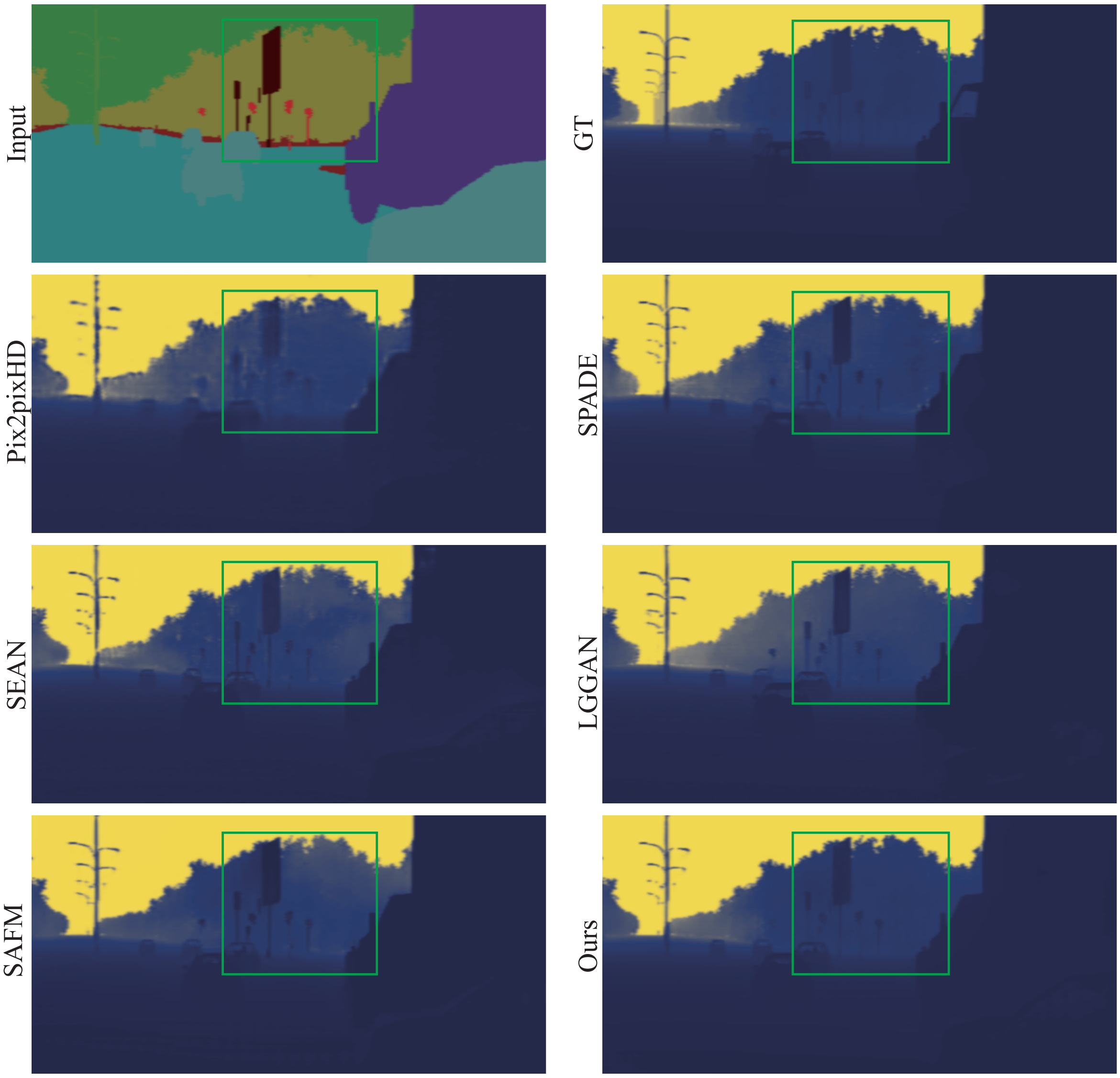


Fig. 8 Depth map comparison for a scene from VKITTI, showing input, ground truth, and output from various methods. Dark blue is closer to and yellow further from the viewer.

Table 3 Ablation study. Starting from the baseline architecture, we demonstrate the effectiveness of each proposed component of our network

Method	MAE ↓	AbsRel ↓	PSNR ↑	FID ↓
Baseline	0.1709	0.2086	17.65	307.16
+ SPE	0.1382	0.1749	20.81	226.94
+ SALN	0.0843	0.1015	26.90	77.24
+ Mixed Attn	0.0755	0.0826	29.58	50.11
+ SWDS (ours)	**0.0613**	**0.0590**	**30.53**	**37.38**

add each component to the framework. Compared to the baseline, adding semantic position embedding (SPE) at each scale brings improvements because it encodes extra semantic positions. The semantically aware layer normalization (SALN) greatly improves performance and training stability by matching semantic and depth features. The mixed attention module enables feature aggregation among different features at the same time. Finally, replacing the output layer by semantically weighted depth synthesis (SWDS) makes the generated depth values more accurate.

4.3.2 Discriminator

In Table 4, we explore choice of discriminator for depth generation. We obtain quantitatively better results using a multiscale design. FPSE and OASIS perform worse because the pixel-wised semantic alignment in the discriminator leads to clear semantic boundaries while introducing a drastic change in the

Table 4 Choice of discriminator

Method	MAE ↓	AbsRel ↓	PSNR ↑	FID ↓
Multiscale	**0.0613**	**0.0590**	**30.13**	**37.38**
FPSE	0.0736	0.0739	29.02	52.39
OASIS	0.9640	0.0878	26.59	66.41

depths of adjacent objects, as shown in the first scene of Fig. 6.

4.3.3 SWDS

In Table 5, we replace the output layer in various semantic image synthesis approaches by our proposed SWDS. The improvements indicate the effectiveness of the SWDS module. Furthermore, we observe that, the worse the performance of the original approach, the greater improvement SWDS provides.

4.3.4 Adabins

We next compare our proposed SWDS to the original Adabins design for depth synthesis.

We note that there are two main differences. On the one hand, we disentangle bin generation from depth weight map generation. In detail, we utilize an SAT block and the following MLP to generate the depth interval and use another SAT block to generate the depth map. Adabins simply uses a transformer to predict both depth bins and weights, which leads to entanglement of different features. We also use a semantic weight map in the encoder–decoder architecture for semantically weighted depth synthesis. The semantic weight map is especially suitable for synthesizing depth maps using semantic layout as input, which is not so for Adabins. As a result, our generated depth maps provide more accurate depth intervals, and thus better depth evaluation metrics, as shown in Table 6. Moreover, the proposed semantically weighted synthesis also enhances the quality of semantic alignment in the generated depth maps, giving a better FID score.

Table 5 Replacing the output layer of various methods by SWDS, denoted †

Method	MAE ↓	AbsRel ↓	PSNR ↑	FID ↓
SPADE	0.1366	0.1447	22.42	119.59
SPADE†	0.1225	0.1208	24.96	108.35
OASIS	0.1199	0.1173	24.40	166.51
OASIS†	0.1084	0.1145	25.82	104.67
SEAN	0.1037	0.0863	26.69	75.34
SEAN†	0.0921	0.0789	28.46	62.57

Table 6 Comparison of SWDS and Adabins

Method	MAE ↓	AbsRel ↓	PSNR ↑	FID ↓
Adabins	0.0756	0.0671	29.69	48.24
SWDS	**0.0613**	**0.0590**	**30.53**	**37.38**

4.3.5 Loss functions

Various loss functions commonly adopted in depth estimation tasks for depth generation were assessed as alternatives: see Table 7. Replacing perceptual loss by structural similarity loss [61] gives worse results in our method: the local window size in SSIM loss leads to block artifacts. Adding scale-invariant loss [62] introduces a threadlike lack of smoothness during adversarial training, again reducing quality.

4.4 Multiple outputs

We can readily extend our method to generate both depth and appearance from semantic input. Specifically, we first introduce appearance supervision to our approach and train a model with two branches for depth and appearance generation respectively. Then, we simply adopt the same design for the two branches with cascaded SAT blocks and upsampling. As Fig. 9 shows, we share the SAT blocks used in

Table 7 Choice of loss function. Per, SSIM, and SI are perceptual loss, structural similarity loss, and scale-invariant loss respectively

Method	MAE ↓	AbsRel ↓	PSNR ↑	FID ↓
SSIM	0.0807	0.7781	28.69	45.89
Per	**0.0613**	**0.0590**	**30.53**	**37.78**
Per+SI	0.0781	0.0689	29.36	56.95

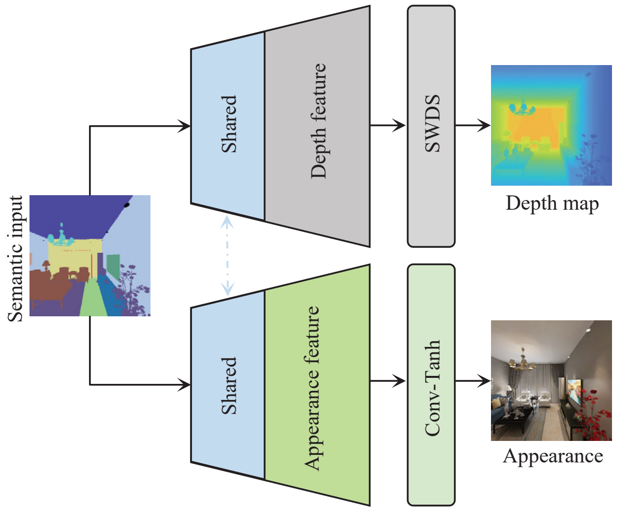


Fig. 9 Depth–appearance generation architecture. We take as input a semantic layout and output a depth map and its appearance simultaneously.

low-resolution generation since the salient features such as edges are mainly generated at low resolutions. Finally, we use the SWDS module for depth synthesis and a Conv-Tanh layer for appearance generation. Figure 10 shows generated 3D point cloud scenes with various appearances using only a simple semantic layout as input, which is easy for visual designers to use. Moreover, shared generation can supervise the depth features with appearance features, also helping improve the depth generation quality, as shown in Table 8. More results using handcrafted semantic layouts as input are shown in the video in the ESM.

In addition, to further verify the influence of the appearance branch on quality of the generated depth, we tried different discriminators in the appearance branch, with results shown in Table 9. The multiscale discriminator in the appearance branch provides better depth quality, while FPSE and OASIS discriminators perform similarly to our single-branch model (see Table 8) even with the help of appearance. The reason is that over-emphasis on semantic boundaries in the shared blocks leads to a depth disparity at object edges (see Table 4). Using the multiscale discriminator in the appearance branch allows our depth generation approach to achieve more continuous depth quality. The results also show the difference between depth generation and image generation.

4.5 3D scene generation

Apart from generating depth maps, our method can generate the 3D scene point cloud, further demonstrating the effectiveness of our depth generation approach. Given the depth map of a perspective view and a fixed camera intrinsic parameter, we can simply generate a point cloud using a *pinhole camera model* as shown in Fig. 1. For better visualization, we project the generated depth maps within a scene in the Structured3D dataset back to the panorama image and then construct the whole 360° scene with the given camera parameters. The accurate geometric details and the flatness of the walls in Fig. 11 show the compelling quality of our generated depth maps.

Table 8 Performance for depth generation alone (D) and for depth–appearance generation (D–A)

Method	MAE ↓	AbsRel ↓	PSNR ↑	FID ↓
D	0.0613	0.0590	30.53	37.38
D–A	**0.0598**	**0.0582**	**31.01**	**32.35**

Table 9 Comparison of our appearance branch to other appearance discriminators

Method	MAE ↓	AbsRel ↓	PSNR ↑	FID ↓
FPSE	0.0614	0.0591	31.15	36.26
OASIS	0.0618	0.0596	**31.25**	36.47
Multiscale	**0.0598**	**0.0582**	31.01	**32.35**

4.6 Depth estimation versus depth generation

Here we explain the key differences between depth generation and depth estimation. Depth estimation models extract rich features from the input image and utilize these features to guide depth prediction. But with a semantic layout as input, they cannot extract sufficient features. Thus adversarial training is needed to guide the model to generate the features. We have tried to retrain depth estimation models, such as Adabins, Midas [63], and DPT using a semantic layout as input, but failed to obtain satisfactory output.

Instead, depth generation is quite different from depth estimation. We cannot use a depth estimation model to predict a dense depth map from a semantic layout. With the help of adversarial training, the depth generation model can generate a depth map from sparse features in the semantic layout in a coarse-to-fine manner.

4.7 Limitations

For one thing, using 256 × 256 resolution depth generation, the constructed sparse point clouds contain 65,536 points. Generating point clouds with higher resolution is time-consuming for training. For another thing, the depth map measures the distance between the surface of the objects and the camera. Regions that are not visible to the camera cannot be constructed by our method, resulting in partial point clouds. To further address the occlusion issue for 3D modeling, it would be interesting future work to further include a point cloud completion module taking as input the partial point clouds generated by our method.

Another issue concerns the user interface: the shapes of drawn objects may not match their standard appearances in the training set, especially for indoor objects. This will also lead to artifacts. In future we hope to increase the robustness of the model to irregular objects.

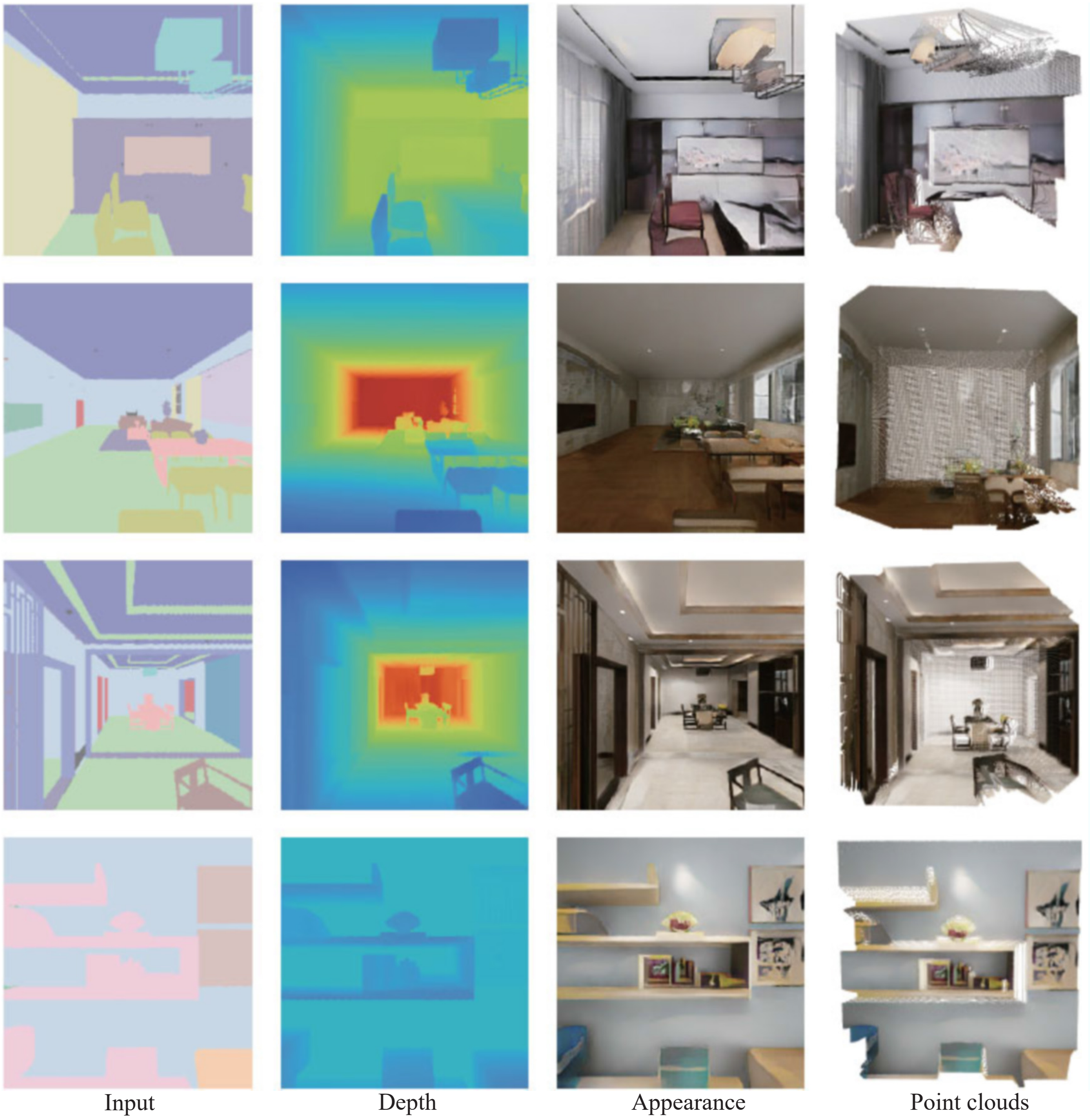


Fig. 10 Using our two-branch model to generate appearance and depth at the same time. The point clouds are constructed with the generated appearance. In the depth map, blue is close and red is far.

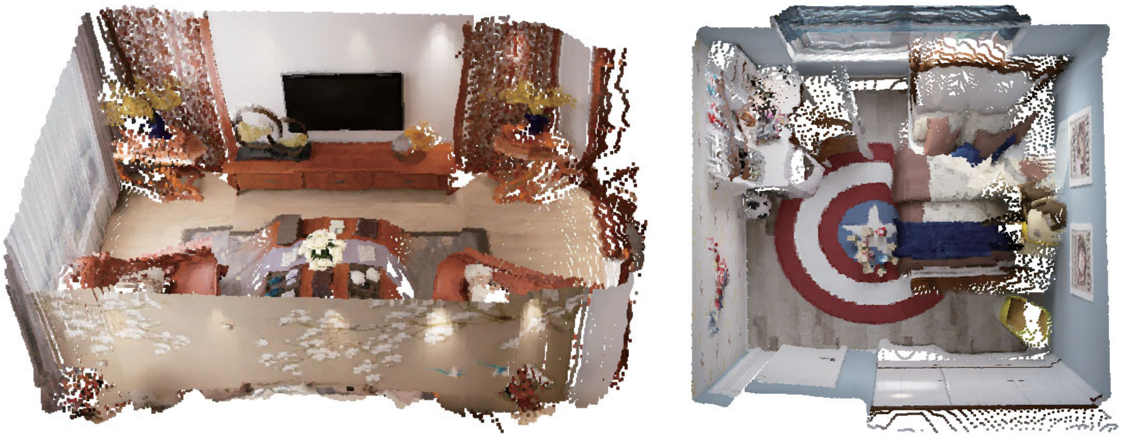


Fig. 11 Point clouds constructed from our generated depth maps. Appearances are taken from the dataset, with ceilings cropped for visualization.

5 Conclusions

We have proposed a novel method, DepthGAN, to solve a proposed depth generation task whose input is a semantic layout. It provides an effective and controllable solution for complex 3D scene generation for the first time. First, we build a cascade of semantically aware transformer blocks with semantically aware layer normalization and mixed attention, enabling semantically-based depth feature generation. The generated depth features are then utilized to synthesize the depth map using our proposed semantically weighted depth synthesis module. Extensive evaluations on multiple datasets verify both quantitatively and qualitatively that our approach provides valid, meaningful depth maps and 3D scenes. Furthermore, our method permits scene manipulation by simply editing the input layout, which is crucial for visual designers. We hope to explore generating further 3D representations such as meshes and implicit functions from a semantic layout in future work.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (U21A20515, 62102393, 62206263, 62271467), and Beijing Natural Science Foundation (4242053).

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

Electronic Supplementary Material

Electronic supplementary material is available in the online version of this article at `https://doi.org/10.1007/s41095-023-0350-8`.

References

[1] Xie, J.; Xu, Y.; Zheng, Z.; Zhu, S. C.; Wu, Y. N. Generative PointNet: Deep energy-based learning on unordered point sets for 3D generation, reconstruction and classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 14971–14980, 2021.

[2] Li, R.; Li, X.; Hui, K. H.; Fu, C. W. SP-GAN: Sphere-guided 3D shape generation and manipulation. *ACM Transactions on Graphics* Vol. 40, No. 4, Article No. 151, 2021.

[3] Zhou, L.; Du, Y.; Wu, J. 3D shape generation and completion through point-voxel diffusion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 5806–5815, 2021.

[4] Wen, C.; Zhang, Y.; Li, Z.; Fu, Y. Pixel2Mesh++: Multi-view 3D mesh generation via deformation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 1042–1051, 2019.

[5] Wei, X.; Chen, Z.; Fu, Y.; Cui, Z.; Zhang, Y. Deep hybrid self-prior for full 3D mesh generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 5785–5794, 2021.

[6] Mittal, P.; Cheng, Y. C.; Singh, M.; Tulsiani, S. AutoSDF: Shape priors for 3D completion, reconstruction and generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 306–315, 2022.

[7] Genova, K.; Cole, F.; Sud, A.; Sarna, A.; Funkhouser, T. Local deep implicit functions for 3D shape. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4856–4865, 2020.

[8] Luo, A.; Zhang, Z.; Wu, J.; Tenenbaum, J. B. End-to-end optimization of scene layout. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 3753–3762, 2020.

[9] Dhamo, H.; Manhardt, F.; Navab, N.; Tombari, F. Graph-to-3D: End-to-end generation and manipulation of 3D scenes using scene graphs. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 16332–16341, 2021.

[10] Park, T.; Liu, M. Y.; Wang, T. C.; Zhu, J. Y. Semantic image synthesis with spatially-adaptive normalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2332–2341, 2019.

[11] Lv, Z.; Li, X.; Niu, Z.; Cao, B.; Zuo, W. Semantic-shape adaptive feature modulation for semantic image synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 11204–11213, 2022.

[12] Chen, W.; Hays, J. SketchyGAN: Towards diverse and realistic sketch to image synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 9416–9425, 2018.

[13] Ghosh, A.; Zhang, R.; Dokania, P.; Wang, O.; Efros, A.; Torr, P.; Shechtman, E. Interactive sketch & fill: Multiclass sketch-to-image translation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 1171–1180, 2019.

[14] Brodt, K.; Bessmeltsev, M. Sketch2Pose: Estimating a 3D character pose from a bitmap sketch. *ACM Transactions on Graphics* Vol. 41, No. 4, Article No. 85, 2022.

[15] Yao, Y.; Luo, Z.; Li, S.; Fang, T.; Quan, L. MVSNet: Depth inference for unstructured multi-view stereo. In: *Computer Vision – ECCV 2018. Lecture Notes in Computer Science, Vol. 11212*. Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 785–801, 2018.

[16] Yin, W.; Zhang, J.; Wang, O.; Niklaus, S.; Mai, L.; Chen, S.; Shen, C. Learning to recover 3D scene shape from a single image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 204–213, 2021.

[17] Luo, W.; Li, Y.; Urtasun, R.; Zemel, R. Understanding the effective receptive field in deep convolutional neural networks. In: Proceedings of the 30th International Conference on Neural Information Processing Systems, 4905–4913, 2016.

[18] Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A. C.; Bengio, Y. Generative adversarial nets. In: Proceedings of the Annual Conference on Neural Information Processing Systems, 2672–2680, 2014.

[19] Brock, A.; Donahue, J.; Simonyan, K. Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint* arXiv:1809.11096, 2018.

[20] Karras, T.; Laine, S.; Aila, T. A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4396–4405, 2019.

[21] Esser, P.; Rombach, R.; Ommer, B. Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 12868–12878, 2021.

[22] Isola, P.; Zhu, J. Y.; Zhou, T.; Efros, A. A. Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 5967–5976, 2017.

[23] Reed, S.; Akata, Z.; Yan, X.; Logeswaran, L.; Schiele, B.; Lee, H. Generative adversarial text to image synthesis. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning, Vol. 48, 1060–1069, 2016.

[24] Johnson, J.; Gupta, A.; Li, F. F. Image generation from scene graphs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 1219–1228, 2018.

[25] Zhu, P.; Abdal, R.; Qin, Y.; Wonka, P. SEAN: Image synthesis with semantic region-adaptive normalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5103–5112, 2020.

[26] Tan, Z.; Chen, D.; Chu, Q.; Chai, M.; Liao, J.; He, M.; Yuan, L.; Hua, G.; Yu, N. Efficient semantic image synthesis via class-adaptive normalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 44, No. 9, 4852–4866, 2022.

[27] Tan, Z.; Chai, M.; Chen, D.; Liao, J.; Chu, Q.; Liu, B.; Hua, G.; Yu, N. Diverse semantic image synthesis via probability distribution modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 7958–7967, 2021.

[28] Liu, X.; Yin, G.; Shao, J.; Wang, X.; Li, H. Learning to predict layout-to-image conditional convolutions for semantic image synthesis. *arXiv preprint* arXiv:1910.06809, 2019.

[29] Sushko, V.; Schönfeld, E.; Zhang, D.; Gall, J.; Schiele, B.; Khoreva, A. You only need adversarial supervision for semantic image synthesis. *arXiv preprint* arXiv:2012.04781, 2020.

[30] Tang, H.; Xu, D.; Yan, Y.; Torr, P. H. S.; Sebe, N. Local class-specific and global image-level generative adversarial networks for semantic-guided scene generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 7867–7876, 2020.

[31] Tang, H.; Shao, L.; Torr, P. H. S.; Sebe, N. Local and global GANs with semantic-aware upsampling for image generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 45, No. 1, 768–784, 2023.

[32] Wang, Y.; Qi, L.; Chen, Y. C.; Zhang, X.; Jia, J. Image synthesis via semantic composition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 13729–13738, 2021.

[33] Facil, J. M.; Ummenhofer, B.; Zhou, H.; Montesano, L.; Brox, T.; Civera, J. CAM-convs: Camera-aware multi-scale convolutions for single-view depth. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 11818–11827, 2019.

[34] 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 preprint* arXiv:1907.10326, 2019.

[35] Garg, R.; B G, V. K.; Carneiro, G.; Reid, I. Unsupervised CNN for single view depth estimation: Geometry to the rescue. In: *Computer Vision – ECCV 2016. Lecture Notes in Computer Science, Vol. 9912*. Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 740–756, 2016.

[36] Wang, R.; Pizer, S. M.; Frahm, J. M. Recurrent neural network for (un-) supervised learning of monocular video visual odometry and depth. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5550–5559, 2019.

[37] Zhu, S.; Brazil, G.; Liu, X. The edge of depth: Explicit constraints between segmentation and depth. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 13113–13122, 2020.

[38] Aleotti, F.; Tosi, F.; Poggi, M.; Mattoccia, S. Generative adversarial networks for unsupervised monocular depth prediction. In: *Computer Vision – ECCV 2018 Workshops. Lecture Notes in Computer Science, Vol. 11129*. Leal-Taixé, L.; Roth, S. Eds. Springer Cham, 337–354, 2019.

[39] Chakravarty, P.; Narayanan, P.; Roussel, T. GEN-SLAM: Generative modeling for monocular simultaneous localization and mapping. In: Proceedings of the International Conference on Robotics and Automation, 147–153, 2019.

[40] Ranftl, R.; Bochkovskiy, A.; Koltun, V. Vision transformers for dense prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 12159–12168, 2021.

[41] Farooq Bhat, S.; Alhashim, I.; Wonka, P. AdaBins: Depth estimation using adaptive bins. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4008–4017, 2021.

[42] Bhat, S. F.; Alhashim, I.; Wonka, P. LocalBins: Improving depth estimation by learning local distributions. *arXiv preprint* arXiv:2203.15132, 2022.

[43] 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 16x16 words: Transformers for image recognition at scale. *arXiv preprint* arXiv:2010.11929, 2020.

[44] Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. In: Proceedings of the 34th Conference on Neural Information Processing Systems, 2020.

[45] Wu, H.; Xiao, B.; Codella, N.; Liu, M.; Dai, X.; Yuan, L.; Zhang, L. CvT: Introducing convolutions to vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 22–31, 2021.

[46] Lee, K.; Chang, H.; Jiang, L.; Zhang, H.; Tu, Z.; Liu, C. ViTGAN: Training GANs with vision transformers. *arXiv preprint* arXiv:2107.04589, 2021.

[47] Jiang, Y.; Chang, S.; Wang, Z. TransGAN: Two pure transformers can make one strong GAN, and that can scale up. In: Proceedings of the 35th Conference on Neural Information Processing Systems, 2021.

[48] Dong, X.; Bao, J.; Chen, D.; Zhang, W.; Yu, N.; Yuan, L.; Chen, D.; Guo, B. CSWin transformer: A general vision transformer backbone with cross-shaped windows. *arXiv preprint* arXiv:2107.00652, 2021.

[49] Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B.; Zhang, Q.; Yang, Y.; et al. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint* arXiv:2103.14030, 2021.

[50] Vaswani, A.; Ramachandran, P.; Srinivas, A.; Parmar, N.; Hechtman, B.; Shlens, J. Scaling local self-attention for parameter efficient visual backbones. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 12889–12899, 2021.

[51] Zhang, B.; Gu, S.; Zhang, B.; Bao, J.; Chen, D.; Wen, F.; Wang, Y.; Guo, B. StyleSwin: Transformer-based GAN for high-resolution image generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 11294–11304, 2022.

[52] Wang, T. C.; Liu, M. Y.; Zhu, J. Y.; Tao, A.; Kautz, J.; Catanzaro, B. High-resolution image synthesis and semantic manipulation with conditional GANs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8798–8807, 2018.

[53] Zheng, J.; Zhang, J.; Li, J.; Tang, R.; Gao, S.; Zhou, Z. Structured3D: A large photo-realistic dataset for structured 3D modeling. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12354*. Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J. M. Eds. Springer Cham, 519–535, 2020.

[54] Armeni, I.; Sax, S.; Zamir, A. R.; Savarese, S. Joint 2D-3D-semantic data for indoor scene understanding. *arXiv preprint* arXiv:1702.01105, 2017.

[55] Gaidon, A.; Wang, Q.; Cabon, Y.; Vig, E. VirtualWorlds as proxy for multi-object tracking analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4340–4349, 2016.

[56] Cohen, T.; Geiger, M.; Köhler, J.; Welling, M. Convolutional networks for spherical signals. *arXiv preprint* arXiv:1709.04893, 2017.

[57] Tateno, K.; Navab, N.; Tombari, F. Distortion-aware convolutional filters for dense prediction in panoramic images. In: *Computer Vision – ECCV 2018. Lecture Notes in Computer Science, Vol. 11220*. Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 732–750, 2018.

[58] Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. GANs trained by a two time-scale

update rule converge to a local Nash equilibrium. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, 6629–6640, 2017.

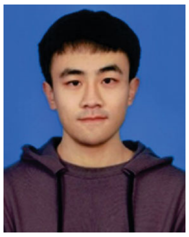
[59] Eigen, D.; Puhrsch, C.; Fergus, R. Depth map prediction from a single image using a multi-scale deep network. In: Proceedings of the 27th International Conference on Neural Information Processing Systems, Vol. 2, 2366–2374, 2014.

[60] Kingma, D. P.; Ba, J. Adam: A method for stochastic optimization. *arXiv preprint* arXiv:1412.6980, 2014.

[61] Wang, Z.; Bovik, A. C.; Sheikh, H. R.; Simoncelli, E. P. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing* Vol. 13, No. 4, 600–612, 2004.

[62] Eigen, D.; Puhrsch, C.; Fergus, R. Depth map prediction from a single image using a multi-scale deep network. In: Proceedings of the 27th International Conference on Neural Information Processing Systems, Vol. 2, 2366–2374, 2014.

[63] Ranftl, R.; Lasinger, K.; Hafner, D.; Schindler, K.; Koltun, V. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 44, No. 3, 1623–1637, 2022.



Yidi Li is a Ph.D. student in the School of Artificial Intelligence, the University of the Chinese Academy of Sciences (UCAS), Beijing, China. He received his bachelor degree in engineering from The University of Science and Technology of China. His research interests include computer vision and generative models.

Jun Xiao is a professor in UCAS. He obtained his Ph.D. degree in communication and information systems from the Graduate University of the Chinese Academy of Sciences, Beijing, in 2008. His research interests include computer graphics, computer vision, image processing, and 3D reconstruction. He is a senior member of the CCF.

Yiqun Wang is an associate professor in the College of Computer Science, Chongqing University. Previously, he was a postdoctoral research fellow in the King Abdullah University of Science and Technology. He received his Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, UCAS. His research interests include computer graphics and 3D vision.

Zhengda Lu received a bachelor degree from Northwestern Polytechnical University, Xi'an, China, in 2016, and his Ph.D. degree from UCAS in 2021. He currently holds a post-doctoral position in the School of Artificial Intelligence, UCAS. His research interests include computer graphics, computer vision, and 3D reconstruction.