SparseGNV: Generating Novel Views of Indoor Scenes with Sparse RGB-D Images

Weihao Cheng, Yan-Pei Cao, Ying Shan

ARC Lab, Tencent PCG
whcheng@tencent.com, caoyanpei@gmail.com, yingsshan@tencent.com

Abstract

We study to generate novel views of indoor scenes given sparse input views. The challenge is to achieve both photorealism and view consistency. We present SparseGNV: a learning framework that incorporates 3D structures and image generative models to generate novel views with three modules. The first module builds a neural point cloud as underlying geometry, providing scene context and guidance for the target novel view. The second module utilizes a transformer-based network to map the scene context and the guidance into a shared latent space and autoregressively decodes the target view in the form of discrete image tokens. The third module reconstructs the tokens back to the image of the target view. SparseGNV is trained across a large-scale indoor scene dataset to learn generalizable priors. Once trained, it can efficiently generate novel views of an unseen indoor scene in a feed-forward manner. We evaluate SparseGNV on real-world indoor scenes and demonstrate that it outperforms state-of-the-art methods based on either neural radiance fields or conditional image generation.

Introduction

Synthesizing high-quality novel views of 3D indoor scenes is a long-standing and challenging task in computer vision (Hedman et al. 2016; Philip et al. 2021; Lei, Tang, and Jia 2022). Typically, this task requires dense scans from various viewpoints as input. However, indoor scenes are often spatially complex, and capturing every region of a scene can be expensive and even intractable. To overcome this challenge, we aim to synthesize novel views with sparse input observations, which reduces the data capture burden. An ideal approach should be capable of generating views by reasonably filling unobserved regions with view consistency.

Sparse-view synthesis methods have gained significant attention recently, particularly those based on neural radiance fields (NeRFs) (Niemeyer et al. 2021; Deng et al. 2022; Roessle et al. 2022), which require inputs with a certain level of view coverage. Due to lack of image generation ability, the above methods are intractable for filling largely unobserved areas. Several methods (Sajjadi et al. 2022; Kulhánek et al. 2022) use transformers (Vaswani et al. 2017) to learn latent scene representations from 2D observations and conditionally generate images given new viewpoints. However, due to lack of explicit 3D representation, it is challenging for these methods to synthesize visual details from unstructured latent space. Another line of work (Gkioxari et al. 2019; Rockwell, Fouhey, and Johnson 2021; Ren et al. 2022) focuses on generating novel views or long-term videos starting from a single image, using generative networks to paint the "outside" of a view autoregressively, but they face limitations in synthesizing consistent views between multiple frames. This leads to the core motivation of our approach: marrying explicit 3D scene structures with image generative models for a joint capability of generating views with limited visual clues and maintaining scene consistency.

Figure 1: The proposed SparseGNV generates novel view images of unseen indoor scenes based on 4 observed views.

We propose SparseGNV: a framework that learns generalizable scene priors to generate novel views conditioned on sparse RGB-D input views. SparseGNV is first trained on a large indoor scene dataset to obtain priors that are generalizable across scenes. Once trained, SparseGNV can efficiently generate novel views through forward passing given observed views of a new scene and target viewpoints, without the need for per-scene optimization. To generate 2D novel views grounded in 3D scene structures, we design SparseGNV with three modules: a neural geometry mod-

The ArXiv version of this paper is available at: https://arxiv.org/abs/2305.07024

ule, a view generator module, and an image converter module. The *neural geometry module* reconstructs a set of input views into a 3D neural point cloud where each point is associated with an embedding vector. The neural point cloud can be rendered to 2D color and mask images from arbitrary viewpoints using volume rendering following PointNeRF (Xu et al. 2022). Although the point cloud can be scattered and incomplete due to input sparsity, the rendered images still provide structural and texture clues for imaging unobserved regions and maintaining consistency. The *view generator module* generates a novel view conditioned on a *scene context* and a *query*. The *scene context* is an overview of the given scene, which consists of the observed images and images rendered by the neural geometry module from multiple sampled viewpoints. It provides a global context that benefits inferring missing regions and maintaining consistency. The *query* specifies the view that is required to generate. It consists of the rendered image from the target viewpoint. The *query* provides guidance to retrieve information from the *scene context* for generating the target novel view. The module uses a joint convolution and transformer-based encoder-decoder network that maps the *scene context* and the *query* to a shared latent space, and then autoregressively generates the novel view in the form of discrete tokens from a codebook (van den Oord, Vinyals, and Kavukcuoglu 2017; Ommer et al. 2020). The *image converter module* is a convolutional decoder network that can reconstruct the discrete tokens back to 2D images in the pixel space. The codebook of tokens provides a compact and expressive representation for images. It enables the use of transformer models which are powerful to align complex scene contexts and queries to target novel views.

We evaluate SparseGNV on a real-world indoor scene dataset (Dai et al. 2017) , and the results outperform recent baselines using either neural radiance fields or conditional image generation. We show example generations of SparseGNV in Figure 1.

Contributions

- We propose SparseGNV: a learning framework to synthesize consistent novel views of indoor scenes with sparse input views. The method combines neural 3D geometry and image generation model to enable photorealistic view synthesis with consistent structure faithful to the observations.
- We design a joint convolution and transformer-based image generation network that effectively incorporates contextual information from 3D scene structures.
- Evaluation results on real-world indoor scenes demonstrate that SparseGNV achieves state-of-the-art performance of synthesizing novel views with only a few observations.

Related Work

Novel View Synthesis Novel view synthesis is a task to produce images of scenes from arbitrary viewpoints given a number of input views. Early work achieves photorealistic synthesis by capturing a dense set of views (Levoy and Hanrahan 1996; Gortler et al. 1996). Recently, neural networks based methods have made significant progress on enabling better synthesis quality, wider ranges of novel viewpoints, and more compact model representation. Neural radiance fields (NeRF) (Mildenhall et al. 2020) is a milestone work that trains a multi-layer perceptron (MLP) to encode radiance and density for producing novel views via volume rendering. Following work based on NeRF extends novel view synthesis on varies of aspects: relaxing image constraints (Martin-Brualla et al. 2020), improving quality (Barron et al. 2021), dynamic view synthesis (Li et al. 2020; Pumarola et al. 2020), pose estimation (Lin et al. 2021; Meng et al. 2021), rendering in real-time (Yu et al. 2021a), and object / scene generation (Poole et al. 2022; Jain et al. 2021).

High-quality synthesis of scene views generally requires iterative per-scene optimizations with large number of observations. As dense inputs is unavailable in many scenarios, the study of few view synthesis is growing rapidly (Sitzmann, Zollhöfer, and Wetzstein 2019; Jain, Tancik, and Abbeel 2021; Niemeyer et al. 2021; Kim, Seo, and Han 2022; Chen et al. 2022), and one direction is to learn priors across scenes and predicts novel views (Yu et al. 2021b; Chen et al. 2021; Wang et al. 2021; Sajjadi et al. 2022; Kulhánek et al. 2022). PixelNeRF (Yu et al. 2021b) is a learning framework that conditions NeRF on one or few input images to predict continuous scene representations. MVSNeRF (Chen et al. 2021) learns a generic deep neural network that combines plane-swept cost volumes with volume rendering for constructing radiance fields. IBRNet (Wang et al. 2021) is a network of MLP and ray transformer that estimates radiance and volume density from multiple source views. Scene Representation Transformer (Sajjadi et al. 2022) combines convolutional networks and transformers to encode input images into latent scene representations and decodes novel views. ViewFormer (Kulhánek et al. 2022) is another transformer based approach with two stages, where images are encoded into tokens via a codebook network in the first stage, and the tokens of novel views are generated autoregressively conditioned on the inputs in the second stage. Depth priors can be helpful for novel view synthesis (Deng et al. 2022; Roessle et al. 2022) which completes a dense depth map first to guide optimization of NeRF. However, these methods can have poor performance given inputs with large sparsity.

Indoor Scene Synthesis from Sparse Views Synthesizing novel view of indoor scenes is a practical task naturally challenged by data sparsity. With incomplete RGB-D scans, SPSG (Dai et al. 2021) generates high-quality colored reconstructions of 3D scenes in the form of TSDF. It uses a self-supervised approach to learn geometry and color inpainting with adversarial and perceptual supervisions on the 2D renderings of the reconstructions. CompNVS (Li et al. 2022) is a framework to synthesis novel views from RGB-D scans with largely incomplete scene coverage. It first encodes scans into neural voxel grids, and then uses a geometry predictor with a texture inpainter to complete the grids with embedding. A neural render decodes the grids into images and refined via adversarial training. These geometry based methods requires strong 3D completion modeling

which are hardly adapted to open-world scenes. PixelSynth (Rockwell, Fouhey, and Johnson 2021) synthesizes novel view of a single image by outpainting unobserved areas projected via 3D reasoning. LookOutsideRoom (Ren et al. 2022) synthesizes long-term video from a single scene image base on an autoregressive transformer modeling consecutive frames. These single image based methods are unable to maintain consistency between observations. Pathdreamer (Koh et al. 2021) targets on generating panorama images at novel positions given one or a few observations. It consists of a structure generator and an image generator. The structure generator projects observations into 3D semantic geometry. The image generator uses SPADE network (Park et al. 2019) to generate photorealistic views from panorama semantic maps. Pathdreamer focuses on panorama images and requires semantic labeling of indoor scene which cannot be applied conventionally.

Methodology

In this section, we first briefly introduce the notations and the problem statement. We then propose SparseGNV with designs of the three modules. Lastly, we introduce the procedures of training and inference.

Notation & Problem

Let $\mathcal{V} = \{(I_i, D_i, \pi_i) \,|\, i = 1, 2, ..., N\}$ be a set of views of indoor scenes, where $I_i \in \mathbb{R}^{W \times H \times 3}$ is the i-th color image, $D_i \in \mathbb{R}^{W \times H}$ is the depth image, and π_i is the camera pose. $\mathcal{V}$ can be divided into an input observed view set $\mathcal{O}$ and a novel view set $\mathcal{X}$. Given an input sparse set of $\mathcal{O}$, our problem is to generate a view image at a target novel viewpoint. As unobserved regions can be large, generating novel views exactly matching ground truth is not easy. We therefore focus on the photorealism of the generations and also the view consistency.

The SparseGNV Framework

We propose SparseGNV: a learning framework incorporating 3D scene structures and image generative models to generate consistent novel views of indoor scenes given only sparse input views. SparseGNV is trained on a large indoor scene dataset to achieve generalization ability. Given sparse input views of an unseen scene, SparseGNV can efficiently generate novel views in a feed-forward manner. SparseGNV is designed with three modules: the neural geometry module, the view generator module, and the image converter module. The neural geometry module takes the input views to build a 3D neural point cloud (Xu et al. 2022) that can provide rendered guidance images from arbitrary viewpoints. The view generator module generates a novel view conditioned on a scene context of global information and a query regarding the target pose. The scene context and the query contain the information provided by the rendered guidance images. They are fed to a convolution encoder and a transformer-based network to generate novel views in the form of discrete image tokens from a codebook (van den Oord, Vinyals, and Kavukcuoglu 2017; Ommer et al. 2020). The image converter module reconstructs the tokens back to the final images through a decoder network. We show an overview of SparseGNV in Figure 2. The detailed description of the three modules is as follows.

Neural Geometry Module. Given an input sparse set of observations $\mathcal{O}$, the neural geometry module builds an underlying 3D neural point cloud, which can be used to produce rendered guidance images from arbitrary poses. Those rendered guidance images provide structural and color clues that can complement scene representation and guide the generation of target novel views.

The module builds a neural point cloud following PointNeRF (Xu et al. 2022) with two steps: 1) reconstructs a 3D point cloud using the input $\mathcal{O}$; 2) assigns each point of the cloud an embedding vector, which is computed by MVSNet (Yao et al. 2018) given the corresponding pixel of the observed image.

With the neural point cloud, the module can produce rendered color images $F_i \in \mathbb{R}^{W \times H \times 3}$ (Xu et al. 2022). In detail, given an arbitrary camera pose π_i, ray marching is performed to query a number of points on each ray. The embedding vectors of all the queried points are mapped to radiance and density via multi-layer perceptrons (MLPs). Through volume rendering, a ray color is obtained and assigned to the corresponding pixel of the image F_i. If a ray hits no neural point, the ray is marked as invalid. All the rays form a validation mask $M_i \in \{0, 1\}^{W \times H}$ indicating which part of F_i is geometrically valid. The module output is formally expressed as:

$$F_i, M_i = \texttt{NeuralGeometry}(\pi_i, \mathcal{O}\,; \theta), \tag{1}$$

where θ is the parameters of module networks including the MVSNet and the MLPs. The mask M_i can be used to filter out the invalid part of F_i for a clear signal.

The module networks are jointly trained to produce a visually reasonable F_i with structure and color information. The objective is regressing F_i to the ground truth color image I_i on the valid rays:

$$\min_{\theta} \sum_{i} ||(F_i - I_i) \odot M_i||_2^2. \tag{2}$$

View Generator Module. The view generator module uses a joint convolution and transformer-based network that takes a scene context and a query as input to generate a target novel view. The scene context is the global information that includes two types of *"previews"*: reference previews and probed previews. The reference previews are from the input observed poses, and the probed previews are from several *sampled* novel poses interpolated between the poses of the observations. The query is a preview from the target novel pose, which specifies the target viewpoint required to generate. Each preview of these three types is composed of four items: 1) an observed image I_i (using an "N/A" image if unavailable); 2) a rendered color image F_i; 3) a rendered mask image M_i; 4) a ray map R_i of origins and directions derived from the camera pose π_i (Sajjadi et al. 2022). For each preview, we concatenate the corresponding I_i, F_i, M_i, and R_i to one multi-channel image, which is then fed into a convolutional network with the output spatially divided into a group of local patches B_i:

$$B_i = \texttt{ConvNet}(I_i \oplus F_i \oplus M_i \oplus R_i). \tag{3}$$

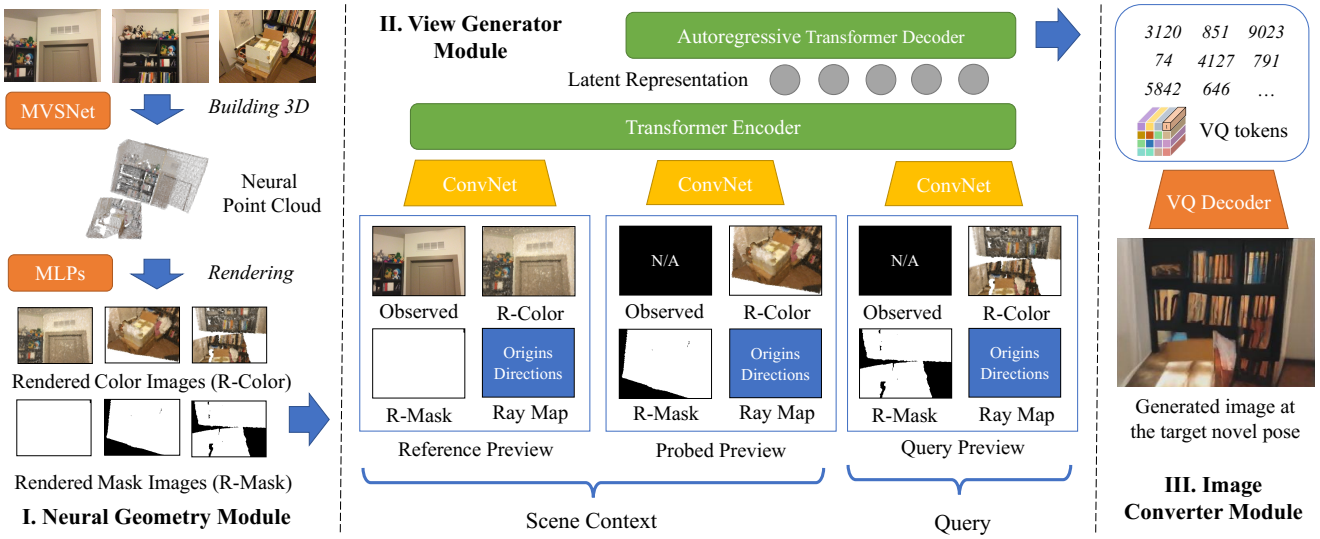

Figure 2: The overview of SparseGNV which consists of three modules: 1) Neural geometry module; 2) View generator module; 3) Image converter module.

Each patch group B_i is additionally labeled by adding a learnable segment embedding (Devlin et al. 2019) regarding one of the three preview categories: reference, probed, and query. This allows the model to distinguish them and utilize information properly. We concatenate all the patches into one sequence, and pass it into a transformer encoder network to obtain a latent representation:

$$h = \texttt{TransformerEncoder}\left(\bigcup_i B_i\right). \tag{4}$$

The latent representation h is a set of hidden vectors that encodes both scene context and query information. The target novel view can then be generated conditioned on h. Due to the recent success of Vector Quantization (VQ) in image synthesis (Ommer et al. 2020; Ramesh et al. 2021; Ren et al. 2022), we present the target image as VQ codebook tokens $\mathcal{S} = \{s_1, s_2, ..., s_T\}$. The distribution of $\mathcal{S}$ is formulated as a probability $p(\mathcal{S}|h)$ which can be factorized as:

$$p(\mathcal{S}|h) = \prod_{t=1}^{T} p(s_t|\mathcal{S}_{<t}, h), \tag{5}$$

where $\mathcal{S}_{<t} = \{s_1, s_2, ..., s_{t-1}\}$, and $p(s_t|\mathcal{S}_{<t}, h)$ is the probability of the t-th image token. We use a transformer decoder to model $p(\mathcal{S}|h)$ by autoregressively estimating $p(s_t|\mathcal{S}_{<t}, h)$. In detail, the last layer of the decoder generates hidden states z, and a linear layer $f(z)$ maps z into a vector with the dimension of the codebook size. The probability $p(s_t|\mathcal{S}_{<t}, h)$ is computed as $\texttt{softmax}(f(z))$. We train the entire network by minimizing the objective of negative log-likelihood loss on the probability estimation:

$$\mathcal{L} = \sum_{s_t \in \mathcal{S}} -\log p(s_t|\mathcal{S}_{<t}, h). \tag{6}$$

Image Converter Module. The image converter module is structurally based on a convolutional autoencoder network that encodes an image into discrete representation and decodes it back to the image. In SparseGNV, the image converter module plays two roles: 1) encoding a ground truth color image I into VQ codebook tokens $\mathcal{S}$ for training the view generation module; 2) decoding a generated $\mathcal{S}$ back to the image at inference. The architecture of the converter network follows VQ-GAN (Ommer et al. 2020).

Training & Inference

The training of SparseGNV requires two stages. In the first stage, the neural geometry module and the image converter module are trained separately. Given scanned views $\mathcal{V} = \{I_i, D_i, \pi_i\}$ of an indoor scene, we randomly sample a set of views as observations to build a neural point cloud, and iterate I_i from $\mathcal{V}$ to supervise the rendered images for training the MVSNet and the MLPs jointly, as shown in Equation (2). The VQ decoder of the image converter module is trained through a self-supervised process of encoding images into VQ tokens and subsequently decoding them back. In the second stage, we use the neural geometry module to produce scene contexts and queries, and supervise the view generator module to generate VQ tokens of novel views obtained by the image converter module, as shown in Equation (6). The inference of SparseGNV is straightforward. Taking a number of observed views, we use the neural geometry module to build a neural point cloud. We then produce the scene context and queries, and pass them to the view generator network. With the output latent representation h, we autoregressively draw out the VQ tokens using multinomial sampling. Lastly, we use the image converter module to reconstruct the VQ tokens back to the final image. Note that there is no optimization process in SparseGNV. With a mod-

	$\|\mathcal{O}\| = 2$			$\|\mathcal{O}\| = 4$			$\|\mathcal{O}\| = 8$		
Method	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Point-NeRF	9.606	0.375	0.689	11.004	0.364	0.680	13.495	0.435	0.617
PixelSynth	11.503	0.412	0.750	12.261	0.443	0.716	12.880	0.459	0.684
IBRNet	11.739	0.400	0.725	12.823	0.450	0.717	14.099	0.524	0.702
ViewFormer	14.365	0.541	0.674	14.927	0.549	0.649	15.420	0.553	0.633
DDP-NeRF	14.281	0.451	0.712	15.799	0.495	0.630	17.491	0.567	0.554
Ours w/o "R-Color"	13.157	0.426	0.699	15.124	0.553	0. 617	16.010	0.537	0.582
Ours w/o "Ray"	15.239	0.559	0.553	15.739	0.559	0.530	16.827	0.569	0.506
Ours w/o "Reference"	14.957	0.560	0.572	15.755	0.562	0.549	17.007	0.573	0.515
Ours w/o "Probed"	**15.825**	0.560	0.539	16.475	**0.569**	0.502	17.402	**0.584**	0.464
Ours	15.713	**0.564**	**0.533**	**16.622**	0.568	**0.500**	**17.931**	0.577	**0.463**

Table 1: Quantitative results on the ScanNet test scenes.

ern GPU, a neural point cloud of a scene can be built within a few seconds (only once), and generating 24 novel views takes about 0.83 second.

Experiments

Experimental Settings

Data Preparation. We use the ScanNet dataset (Dai et al. 2017), following the original train/test split, to study the proposed and baseline methods. For each scan of the dataset, we randomly capture sub-scans of consecutive 256 frames. We then downsample the sub-scans to 32 frames (1/8 ratio) as the samples of view sets. For a training sample, we randomly pick 4 out of 32 frames as observed views and the rest as novel views. For a testing sample, we randomly pick $|\mathcal{O}|$=2, 4, 8 frames as observed views and the rest as novel views for evaluation. Therefore, we have 3 groups of evaluation results. Following the settings of (Roessle et al. 2022), we hold out "scene0708_00", "scene0710_00", "scene0738_00', "scene0758_00", "scene0781_00" as test scenes and randomly select one sample for each scene. The comparing resolution is set to 624 × 468 after scaling and cropping dark borders. In the experiments, we assume accurate camera poses and depths which are provided to all the comparison methods for training and testing.

Evaluation Metrics. We compute the peak signal-to-noise ratio (PSNR), the structural similarity index measure (SSIM) (Wang et al. 2004) and the learned perceptual image patch similarity (LPIPS) (Zhang et al. 2018). We report the averaged metric results.

Method Setting & Baselines. Given a sample of $|\mathcal{O}|$ input views and 32 - $|\mathcal{O}|$ novel views, we use the $|\mathcal{O}|$ input views to build a neural point cloud, which produces the rendered color and mask images for all 32 viewpoints. The scene context is formed by the reference previews produced from the $|\mathcal{O}|$ input views and the probed previews from the 32-$|\mathcal{O}|$-1 novel viewpoints. The query is produced from the rest 1 novel viewpoint. The neural geometry module follows the Point-NeRF settings (Xu et al. 2022). The image converter module follows the settings in (Ren et al. 2022). The network of the view generator module includes an encoder and a decoder. The encoder architecture mainly follows (Sajjadi et al. 2022) with slight modifications. The decoder is a stack of 6 vanilla transformer decoder layers. We train the model with a learning rate 1e-4 and batch size 16 using the Adam optimizer. We set up five baseline methods for comparisons: Point-NeRF (Xu et al. 2022), PixelSynth (Rockwell, Fouhey, and Johnson 2021), IBRNet (Wang et al. 2021), ViewFormer (Kulhánek et al. 2022), and NeRF with dense depth priors (DDP-NeRF) (Roessle et al. 2022). For Point-NeRF, we train its MVSNet and ray marching MLPs across scenes, and test it in a feed-forward manner. For PixelSynth, we take the pre-trained model provided by the authors, and predict novel views by outpainting the re-projection from the nearest observed images. For IBRNet and ViewFormer, we train and test models with the same setting to our method (IBRNet is tested in feed-forward). For DDP-NeRF, we optimize the radiance fields with ground truth depths and camera poses for each scene, and render the novel views.

Primary Results & Analysis

We compare the quantitative results on 3 groups of sparse input views with observation number $|\mathcal{O}|$ = 2, 4, 8, respectively. As the results presented in Table 1, our method outperforms all the baselines on PSNR, SSIM, and LPIPS. We show the generations of novel views with ground truths in Figure 3. The results of Point-NeRF are often corrupted and scattered, which is caused by incomplete underlying point clouds. Without image generation ability, Point-NeRF is unable to fill the missing parts of the novel views. PixelSynth produces distorted views when the novel viewpoints are significantly shifted from the observations. As PixelSynth conditions on a single view, the reasoned 3D surface can only be re-projected correctly within a small area near the input viewpoint. The results of IBRNet are often blurred and show black areas where rays hit no observation due to sparsity. ViewFormer generates basic scene appearances but lacks of details as it only perceives image tokens where visual clues can be missing (Our method perceives full images and decodes tokens). DDP-NeRF performs the best among all the baselines. But due to the sparse inputs, the renderings of DDP-NeRF unavoidably overfit to input views that cause blurs in novel views even with depth in-

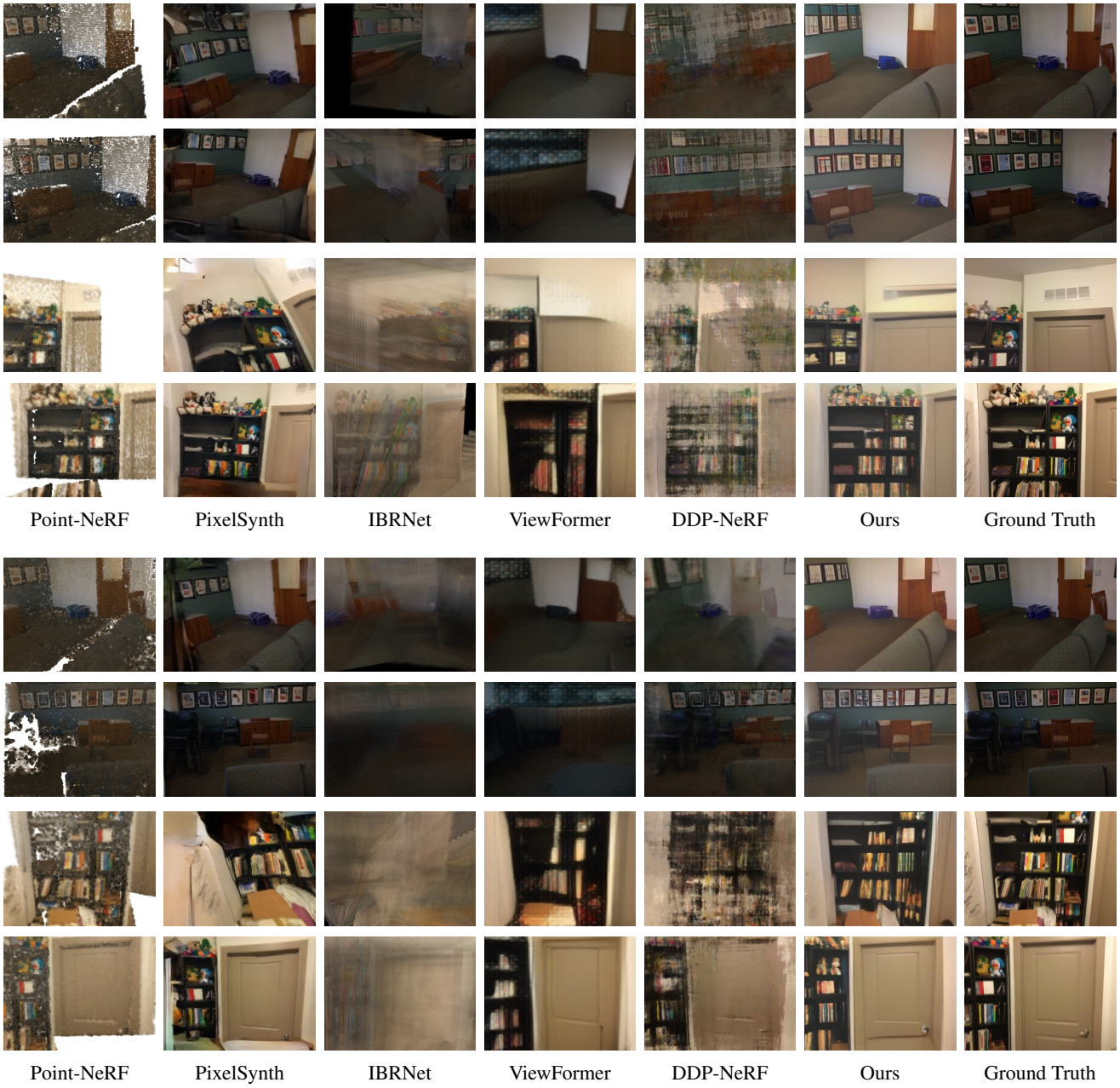

Figure 3: Synthesized novel views given input views with $|\mathcal{O}| = 2$ (Top Half) and $|\mathcal{O}| = 8$ (Bottom Half).

formation. Our method generally outperforms the baselines in terms of fidelity and visual details. With more observed views, our method generates novel views exhibiting better visual quality (shown in metrics). Therefore, the method effectively leverages the input information and demonstrates strong applicability.

Ablation Study

We conduct ablation studies to assess the designs of SparseGNV. The experiments include: 1) w/o "R-Color": the rendered color images from the neural point cloud are excluded from the scene context and the query; 2) w/o "Ray": the ray maps (camera poses) are excluded; 3) w/o "Reference": the reference previews are excluded from the scene context; 4) w/o "Probed": the probed previews are excluded from the scene context. As shown in Table 1, SparseGNV with all components achieves the best results except a few metric numbers. Without "R-Color", the metrics significantly decrease as missing of 3D structural clues from the neural point clouds which are important to high-quality synthesis. Without "Reference", the metrics also decrease a lot due to missing of complete observations, but the metrics

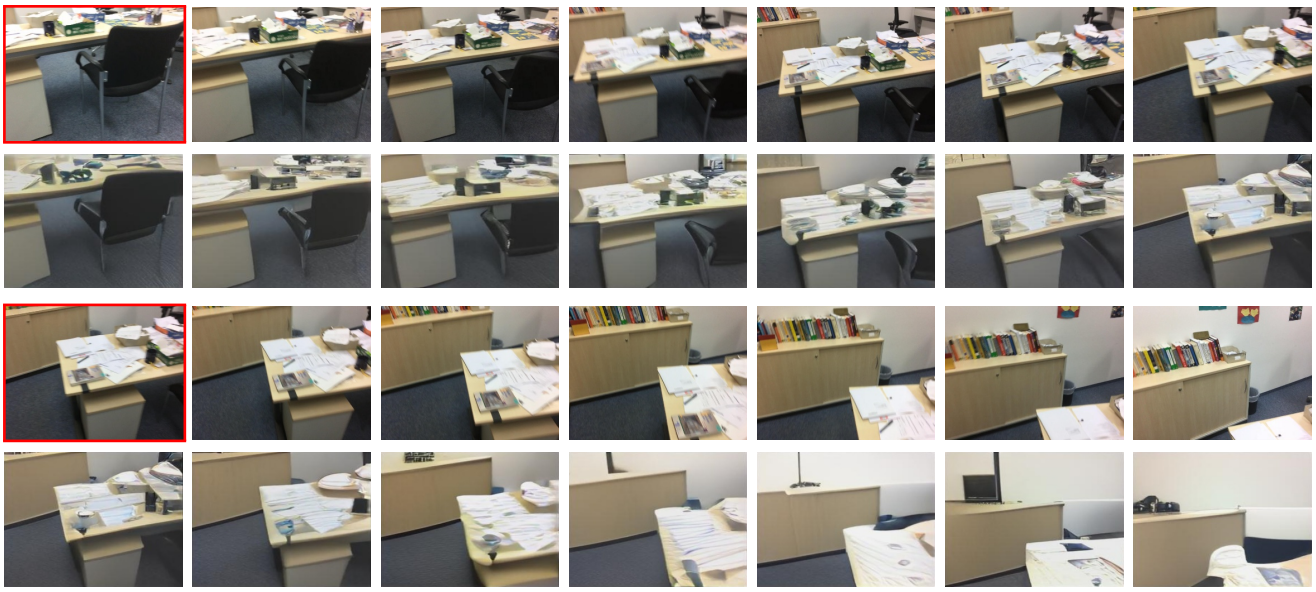

Figure 4: A sequence of continuous generations between only two observations (red box) and moving away. The 1st and 3rd rows are the ground truth. The 2nd and 4th rows are the generated novel views of "in between" and "moving away", respectively.

still outperform the baselines as the probed previews can provide enough scene information. Without "Probed", most of the metrics decline and some are improved. Probed previews could be less important compared to the other components and occasionally mislead the generations. However, they still provide benefits for most cases.

View Consistency

To demonstrate the view consistency of SparseGNV, we show continuous novel view generations between two observations in Figure 4 (first two rows). The quality and consistency are fairly maintained without significant perturbation. The neural geometry module provides a strong scene context of 3D structure, which ensures a stable generation ability by the downstream modules. We further show a sequence of generated novel views that moves away from the two observations in Figure 4 (last two rows). The office desk is fairly maintained until moving away, as there is enough clue of its shape and appearance. Unfortunately, the cabinet appears with only its surface, and the books on top of the cabinet are completely missed of generation. Since there is no clue of their occurrence, the model tends to generate a white wall to maintain consistency.

Time & Memory & Model Size

The experiments are conducted on NVIDIA V100 GPUs. The reconstruction of a neural point cloud takes from a few seconds to less than 1 minutes depending on the number of input views. This is only performed once before rendering the scene. The inference time of image generations is 0.83s per batch of 24 images. The training of the neural geometry module takes about 1 day using 1 GPU (batch size 1, memory $\leq$ 20G depends on scene size). The training of the image generator module takes about 1 week using 2 GPUs (batch size 16, 20.9G). VQ decoder uses the pre-trained checkpoint from (Ren et al. 2022) and is not further fine-tuned. The parameter counts of the three modules are: 0.724M (Point-NeRF), 88M (convolution and transformer network), and 76M (VQ decoder).

Conclusions & Limitations

In this paper, we study the problem of novel view synthesis of indoor scenes given sparse RGB-D input views. To generate both photorealistic and consistent novel views, we propose SparseGNV: a learning framework that marries explicit 3D structures with image generative models. The framework is designed with three network-based modules. The neural geometry module builds a 3D neural point cloud to produce rendered images from arbitrary viewpoints. The view generator module takes the rendered images to form the scene context and queries, which are fed into a convolution and transformer-based network to generate the target novel views represented in VQ codebook tokens. The image converter module finally reconstructs the tokens back to the novel view images. SparseGNV is trained across scenes to learn priors, and infers novel views of unseen scenes in a feed-forward manner. The evaluation results on real-world indoor scenes demonstrate the exceeding performance of the method over recent baselines.

Limitations. SparseGNV synthesizes novel views using an image generation model based on the VQ codebook. The outputs are therefore less 3D consistent compared to the volume rendering-based methods. For example, the object details and lighting could be altered. The framework also requires camera poses which can be unavailable when the input views are extremely sparse.

References

Barron, J. T.; Mildenhall, B.; Tancik, M.; Hedman, P.; Martin-Brualla, R.; and Srinivasan, P. P. 2021. Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields. *Computer Vision and Pattern Recognition*.

Chen, A.; Xu, Z.; Zhao, F.; Zhang, X.; Xiang, F.; Yu, J.; and Su, H. 2021. MVSNeRF: Fast Generalizable Radiance Field Reconstruction From Multi-View Stereo. *International Conference on Computer Vision*.

Chen, D.; Liu, Y.; Huang, L.; Wang, B.; and Pan, P. 2022. GeoAug: Data Augmentation for Few-Shot NeRF with Geometry Constraints. *European Conference on Computer Vision*.

Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Nießner, M. 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. *Computer Vision and Pattern Recognition*.

Dai, A.; Siddiqui, Y.; Thies, J.; Valentin, J.; and Nießner, M. 2021. SPSG: Self-Supervised Photometric Scene Generation from RGB-D Scans. *Computer Vision and Pattern Recognition*.

Deng, K.; Liu, A.; Zhu, J.-Y.; and Ramanan, D. 2022. Depth-supervised nerf: Fewer views and faster training for free.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Gkioxari, G.; Wiles, O.; Szeliski, R.; Johnson, J.; Wiles, O.; Gkioxari, G.; Szeliski, R.; and Johnson, J. 2019. SynSin: End-to-End View Synthesis From a Single Image. *Computer Vision and Pattern Recognition*.

Gortler, S. J.; Grzeszczuk, R.; Szeliski, R.; and Cohen, M. 1996. The lumigraph. *International Conference on Computer Graphics and Interactive Techniques*.

Hedman, P.; Ritschel, T.; Drettakis, G.; and Brostow, G. 2016. Scalable inside-out image-based rendering. *ACM Transactions on Graphics*.

Jain, A.; Mildenhall, B.; Barron, J. T.; Abbeel, P.; and Poole, B. 2021. Zero-Shot Text-Guided Object Generation with Dream Fields. *Computer Vision and Pattern Recognition*.

Jain, A.; Tancik, M.; and Abbeel, P. 2021. Putting NeRF on a Diet: Semantically Consistent Few-Shot View Synthesis. *Computer Vision and Pattern Recognition*.

Kim, M.; Seo, S.; and Han, B. 2022. InfoNeRF: Ray Entropy Minimization for Few-Shot Neural Volume Rendering. *Computer Vision and Pattern Recognition*.

Koh, J. Y.; Lee, H.; Yang, Y.; Baldridge, J.; and Anderson, P. 2021. Pathdreamer: A World Model for Indoor Navigation. *International Conference on Computer Vision*.

Kulhánek, J.; Derner, E.; Sattler, T.; and Babuska, R. 2022. ViewFormer: NeRF-free Neural Rendering from Few Images Using Transformers. *European Conference on Computer Vision*.

Lei, J.; Tang, J.; and Jia, K. 2022. Generative Scene Synthesis via Incremental View Inpainting using RGBD Diffusion Models.

Levoy, M.; and Hanrahan, P. 1996. Light field rendering. *International Conference on Computer Graphics and Interactive Techniques*.

Li, Z.; Fan, T.; Li, Z.; Cui, Z.; Sato, Y.; Pollefeys, M.; and Oswald, M. R. 2022. CompNVS: Novel View Synthesis with Scene Completion.

Li, Z.; Niklaus, S.; Snavely, N.; and Wang, O. 2020. Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Computer Vision and Pattern Recognition*.

Lin, C.-H.; Ma, W.-C.; Torralba, A.; and Lucey, S. 2021. BARF: Bundle-Adjusting Neural Radiance Fields. *International Conference on Computer Vision*.

Martin-Brualla, R.; Radwan, N.; Sajjadi, M. S. M.; Barron, J. T.; Dosovitskiy, A.; and Duckworth, D. 2020. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. *Computer Vision and Pattern Recognition*.

Meng, Q.; Chen, A.; Luo, H.; Wu, M.; Su, H.; Xu, L.; He, X.; and Yu, J. 2021. GNeRF: GAN-based Neural Radiance Field without Posed Camera. *Computer Vision and Pattern Recognition*.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. *European Conference on Computer Vision*.

Niemeyer, M.; Barron, J. T.; Mildenhall, B.; Sajjadi, M. S. M.; Geiger, A.; and Radwan, N. 2021. RegNeRF: Regularizing Neural Radiance Fields for View Synthesis from Sparse Inputs. *Computer Vision and Pattern Recognition*.

Ommer, B.; Esser, P.; Rombach, R.; Esser, P.; Rombach, R.; and Ommer, B. 2020. Taming Transformers for High-Resolution Image Synthesis. *Computer Vision and Pattern Recognition*.

Park, T.; Liu, M.-Y.; Wang, T.-C.; and Zhu, J.-Y. 2019. Semantic Image Synthesis With Spatially-Adaptive Normalization. *Computer Vision and Pattern Recognition*.

Philip, J.; Morgenthaler, S.; Gharbi, M.; and Drettakis, G. 2021. Free-viewpoint indoor neural relighting from multi-view stereo. *ACM Transactions on Graphics*.

Poole, B.; Jain, A.; Barron, J. T.; and Mildenhall, B. 2022. DreamFusion: Text-to-3D using 2D Diffusion. *arXiv*.

Pumarola, A.; Corona, E.; Pons-Moll, G.; and Moreno-Noguer, F. 2020. D-NeRF: Neural Radiance Fields for Dynamic Scenes. *Computer Vision and Pattern Recognition*.

Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; and Sutskever, I. 2021. Zero-Shot Text-to-Image Generation. *International Conference on Machine Learning*.

Ren, X.; Xiaolong, H.; Uc, W.; and Diego, S. 2022. Look Outside the Room: Synthesizing A Consistent Long-Term 3D Scene Video from A Single Image. *Computer Vision and Pattern Recognition*.

Rockwell, C.; Fouhey, D. F.; and Johnson, J. 2021. PixelSynth: Generating a 3D-Consistent Experience From a Single Image. *International Conference on Computer Vision*.

Roessle, B.; Barron, J. T.; Mildenhall, B.; Srinivasan, P. P.; and Niessner, M. 2022. Dense Depth Priors for Neural Radiance Fields from Sparse Input Views. *Computer Vision and Pattern Recognition*.

Sajjadi, M. S. M.; Meyer, H.; Pot, E.; Bergmann, U.; Greff, K.; Radwan, N.; Vora, S.; Lucic, M.; Duckworth, D.; Dosovitskiy, A.; Uszkoreit, J.; Funkhouser, T.; and Tagliasacchi, A. 2022. Scene Representation Transformer: Geometry-Free Novel View Synthesis Through Set-Latent Scene Representations. *Computer Vision and Pattern Recognition*.

Sitzmann, V.; Zollhöfer, M.; and Wetzstein, G. 2019. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*.

van den Oord, A.; Vinyals, O.; and Kavukcuoglu, K. 2017. Neural Discrete Representation Learning. *Advances in Neural Information Processing Systems*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Wang, Q.; Wang, Z.; Genova, K.; Srinivasan, P. P.; Zhou, H.; Barron, J. T.; Martin-Brualla, R.; Snavely, N.; and Funkhouser, T. 2021. IBRNet: Learning Multi-View Image-Based Rendering. *Computer Vision and Pattern Recognition*.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*.

Xu, Q.; Xu, Z.; Philip, J.; Bi, S.; Shu, Z.; Sunkavalli, K.; and Neumann, U. 2022. Point-nerf: Point-based neural radiance fields. *Computer Vision and Pattern Recognition*.

Yao, Y.; Luo, Z.; Li, S.; Fang, T.; and Quan, L. 2018. MVSNet: Depth Inference for Unstructured Multi-view Stereo. *European Conference on Computer Vision*.

Yu, A.; Li, R.; Tancik, M.; Li, H.; Ng, R.; and Kanazawa, A. 2021a. PlenOctrees for Real-time Rendering of Neural Radiance Fields. *International Conference on Computer Vision*.

Yu, A.; Ye, V.; Tancik, M.; and Kanazawa, A. 2021b. pixelNeRF: Neural Radiance Fields from One or Few Images. *Computer Vision and Pattern Recognition*.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. *Computer Vision and Pattern Recognition*.