Scene Representation Transformer:  
Geometry-Free Novel View Synthesis Through Set-Latent Scene Representations

Mehdi S. M. Sajjadi  Henning Meyer  Etienne Pot  Urs Bergmann  Klaus Greff  
Noha Radwan  Suhani Vora  Mario Lučić  Daniel Duckworth  Alexey Dosovitskiy*  
Jakob Uszkoreit*  Thomas Funkhouser  Andrea Tagliasacchi‡

Google Research  ‡ Simon Fraser University

Abstract

A classical problem in computer vision is to infer a 3D scene representation from few images that can be used to render novel views at interactive rates. Previous work focuses on reconstructing pre-defined 3D representations, e.g. textured meshes, or implicit representations, e.g. radiance fields, and often requires input images with precise camera poses and long processing times for each novel scene.

In this work, we propose the Scene Representation Transformer (SRT), a method which processes posed or unposed RGB images of a new area, infers a “set-latent scene representation”, and synthesizes novel views, all in a single feed-forward pass. To calculate the scene representation, we propose a generalization of the Vision Transformer to sets of images, enabling global information integration, and hence 3D reasoning. An efficient decoder transformer parameterizes the light field by attending into the scene representation to render novel views. Learning is supervised end-to-end by minimizing a novel-view reconstruction error.

We show that this method outperforms recent baselines in terms of PSNR and speed on synthetic datasets, including a new dataset created for the paper. Further, we demonstrate that SRT scales to support interactive visualization and semantic segmentation of real-world outdoor environments using Street View imagery.

1. Introduction

The goal of our work is interactive novel view synthesis: given few RGB images of a previously unseen scene, we synthesize novel views of the same scene at interactive rates and without expensive per-scene processing. Such a system can be useful for virtual exploration of urban spaces [26], as well as other mapping, visualization, and AR/VR applications [27]. The main challenge here is to infer a scene representation that encodes enough 3D information to render novel views with correct parallax and occlusions.

Traditional methods build explicit 3D representations, such as colored point clouds [1], meshes [31], voxels [36], octrees [42], and multi-plane images [52]. These representations enable interactive rendering, but they usually require expensive and fragile reconstruction processes.

More recent work has investigated representing scenes with purely implicit representations [39]. Notably, NeRF represents the scene as a 3D volume parameterized by an MLP [25], and has been demonstrated to scale to challenging real-world settings [23]. However, it typically requires hundreds of MLP evaluations for the volumetric rendering of each ray, relies on accurate camera poses, and requires an expensive training procedure to onboard new scenes, as no parameters are shared across different scenes.
Follow-ups that address these shortcomings include re-projection based methods which still rely on accurate camera poses due to their explicit use of geometry, and latent models which are usually geometry-free and able to reason globally, an advantage for the sparse input view setting. However, those methods generally fail to scale to complex real-world datasets. As shown in Tab. 1, interactive novel view synthesis on complex real-world data remains a challenge.

We tackle this challenge by employing an encoder-decoder model built on transformers, learning a scalable implicit representation, and replacing explicit geometric operations with learned attention mechanisms. Different from a number of prior works that learn the latent scene representations through an auto-decoder optimization procedure [38, 16], we specifically rely on an encoder architecture to allow for instant inference on novel scenes. At the same time, we do not rely on explicit locally-conditioned geometry [49, 41] to instead allow the model to reason globally. This not only comes with the advantage of stronger generalization abilities (e.g., for rendering novel cameras further away from the input views, see Fig. 4), but also enables the model to be more efficient, as global information is processed once per scene (e.g., to reconstruct 3D geometry and to resolve occlusions) in our model, rather than once or hundreds of times per rendered pixel, as in previous methods [49, 41, 44].

We evaluate the proposed model on several datasets of increasing complexity against relevant prior art, and see that it strikes a unique balance between scalability to complex scenes (Sec. 4), robustness to noisy camera poses (or no poses at all, Fig. 6), and efficiency in interactive applications (Tab. 3). Further, we show a proof-of-concept for the downstream task of semantic segmentation using the learned representation on a challenging real-world dataset (Fig. 7).

2. Related Work

There is a long history of prior work on novel view synthesis from multiple RGB images based on neural representations (see Tab. 1 and [40]). Recent work has demonstrated that novel views can be synthesized using deep networks trained to compute radiance values for a given 3D position and direction. In particular, NeRF [25] trains an MLP to store radiance and density used in a volume rendering system and produces novel views with remarkably fine details. Several extensions have been proposed to mitigate some its shortcomings, including ones to accelerate rendering [12, 30, 48], handle color variations [23], optimize camera poses [21], and perform well with few images [15]. However, these methods generally still require an expensive optimization of an MLP for each novel scene, and thus are impractical for deployment at large scale.

NeRF with Re-projection. Several NeRF extensions have been proposed that do not always require to be trained individuually for each novel scene. These methods rely on explicit 3D-to-2D projections of samples along the rays to gather features extracted from CNNs, which are then aggregated by a neural network for each point [41, 44, 49], or into a voxel grid [5]. As a result, these networks can be pre-trained, and then applied to novel scenes either without any optimization, or with additional per-scene fine-tuning. However, reliance on explicit camera projections requires precise camera poses, and direct projections cannot take occlusions of objects between samples and input features into account.

Light Fields. Other methods learn to represent a scene with a latent vector from which novel views can be synthesized [7]. More recently, LFN [38] use an auto-decoder to infer a latent code from input images, which then conditions an MLP for computing radiance values for rays of novel views. This method is several times faster than volumetric rendering due to the fact that the network is only queried once per pixel, rather than hundreds of times for volumetric rendering. However, it does not scale beyond simple controlled settings, requires an expensive optimization to produce a latent code for a novel scene, and therefore also relies upon accurate input camera poses. In contrast, we utilize an encoder to infer the scene representation, which executes nearly instantly in comparison (see Tab. 3), and can leverage, but does not require, camera poses (see Sec. 4.4).

Transformers. Originally proposed for seq-to-seq tasks in natural language processing, transformers [43] drop the recurrent memory of prior methods, and instead use a multi-head attention mechanism to gather information from the most relevant portions of the input. The vision community has deeply researched applications thereof in image

Table 1. Comparison of features with prior methods. The rows indicate whether each method: 1) has been demonstrated on real world data, 2) is designed to provide coherence between nearby views, 3) performs inference in real-time, 4) handles images with varying appearance in the same scene, 5) works without known camera poses at test time, 6) works with very sparse input images, 7) generalizes from a training set to novel scenes, 8) captures novel scenes in a feed-forward step, and 9) learns a global latent scene representation.
The text is a description of a network architecture for generating novel views from a set of input images. The architecture consists of a CNN that extracts patch features from the input images, which are then processed by an encoder transformer to generate a set-latent scene representation. This representation is then used by a decoder transformer to generate novel views.

### 3. Method

Our model receives as input an unordered collection of optionally posed images \( \{ I_i \} \in \mathbb{R}^{H \times W \times 3} \), extrinsic camera matrices \( c_i \in SE(3) \), and intrinsic camera matrices \( K_i \in \mathbb{R}^{3 \times 3} \) of the same scene, with extrinsic camera matrix \( c_i = [R_i | t_i] \) and intrinsic camera matrix \( K_i \). The model encodes this collection into a set representation of cardinality \( Z \) via an encoder \( E \) with trainable parameters \( \theta_E \):

\[
\{ \mathbf{z}_z \in \mathbb{R}^d \} = E_{\theta_E} ( CNN_{\theta_{CNN}} ( \{ I_i, c_i, K_i \} ))
\]

In more details, our encoder first processes the images via a shared CNN backbone (Fig. 2, left), and then aggregates features into a set of flat patch embeddings. This set of embeddings is then processed by the encoder transformer that generates the codes \( \{ \mathbf{z}_z \} \) as output. Note that \( \{ \mathbf{z}_z \} \) fully encodes a specific 3D scene as observed by the corresponding set of images, hence it is our set-latent scene representation.

This representation is queried by the decoder transformer \( D \) with trainable parameters \( \theta_D \) to generate the pixel color for a view ray \( r = (o, d) \) with origin \( o \) and direction \( d \):

\[
C(r) = D_{\theta_D}(r | \{ \mathbf{z}_z \})
\]

By convention, we assume \( I_0 \) to be the canonical image, hence queries \( r \) are expressed in the coordinates of \( c_0 \), and, whenever camera information is not available, we assume the canonical camera pose to be identity. If pose information is known, images within a scene are randomly shuffled, hence the reference frame is randomly chosen both at train and test time. Given a collection of images \( \{ I_{s,i} \} \) from different scenes indexed by \( s \), we train all parameters end-to-end via a reconstruction loss on novel-view synthesis:

\[
\arg \min_{\theta = [\theta_{CNN}, \theta_E, \theta_D]} \sum_s \mathbb{E}_{r \sim \mathcal{S}^s} \| C(r) - I_{s,i}(r) \|_2^2
\]

As both inputs and outputs of the model during training are 2D images, there is no explicit 3D supervision. We now describe the internals of our architecture in details, as illustrated in Fig. 2.

**Pose information.** We start by concatenating along the channel dimension the positional encoding \( \gamma \) \([43, 25]\) of camera origin with \( L_o \) octaves and camera direction with...
All features together form (unsorted) sets of scene representations. This network integrates information of the scene at a global scale by attending across patches and input images to infer a 3D scene representation. Note that crucially, the set-latent transformation that alters features in the scene representation to calculate the output color.

The output of the decoder is finally passed to a small 2-layer MLP to compute the pixel color.

3.1. Training and Inference

During training, all components are simply trained end-to-end using an L2 reconstruction loss on novel views, see Eq. (3). We provide further training details in the Appendix A.2. For inference, input images are encoded into a scene representation once, which can then be used to render an arbitrary number of novel views (e.g., for a video). Note that this methodology is designed to maximize inference efficiency, as the 3D reasoning is executed only once in the encoder transformer, independent of the number of novel views to be rendered. For downstream tasks, the encoder for the scene representation can be pre-trained and either frozen, or fine-tuned along with a new decoder for a novel task (e.g., semantic segmentation - Sec. 4.5).

4. Experimental Results

We run a series of experiments to evaluate how well SRT performs novel view synthesis in comparison to previous works and to study how each component of the network architecture contributes to the results. In all experiments, all methods are pre-trained on a dataset of images from one set of scenes and then tested on images from novel scenes. Unless stated otherwise, we train SRT with the same model architecture, hyper parameters, and training protocols for all experiments across all datasets. Synthesized images are evaluated with PSNR, SSIM [46], and LPIPS [50] (VGG).

---

### Table 2. Quantitative Results

<table>
<thead>
<tr>
<th>Model</th>
<th>LFN [38]</th>
<th>PN [49]</th>
<th>SRT (Ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>↑ PSNR</td>
<td>↑ SSIM</td>
<td>↓ LPIPS</td>
</tr>
<tr>
<td>------------------------</td>
<td>----------</td>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td>24.95</td>
<td>0.870</td>
<td>-</td>
</tr>
<tr>
<td>NMR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi-ShapeNet</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 3. Computational Performance

<table>
<thead>
<tr>
<th>Model</th>
<th>Model size (# parameters)</th>
<th>Computation speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>------------------------</td>
<td>----------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>LFN [38]</td>
<td>105 M</td>
<td>~100 s</td>
</tr>
<tr>
<td>PN [49]</td>
<td>28 M</td>
<td>192 fps</td>
</tr>
<tr>
<td>SRT (Ours)</td>
<td>74 M</td>
<td>1.3 fps</td>
</tr>
</tbody>
</table>

- **a)** Scene encoding time
- **b)** Image rendering speed
- **c)** New-scene video rendering
4.1. Datasets

Neural 3D Mesh Renderer Dataset (NMR) [17]. NMR has been used in several previous studies. It consists of ShapeNet [4] objects rendered at 24 fixed views. The dataset is very simple, and likely does not provide good evidence for the applicability in real-world settings.

MultiShapeNet (MSN). We therefore propose this significantly more challenging dataset rendered using photorealistic ray tracing [11]. Each scene has 16–31 ShapeNet objects dropped at random locations within an invisible bounding box. We further sample from 382 complex HDR backgrounds and environment maps. Viewpoints are selected by uniformly sampling within a half-sphere shell.

This dataset is significantly more difficult than NMR for several reasons: 1) images are rendered photorealistically, 2) scenes contain many different types of objects in complex, tightly-packed configurations, 3) the nontrivial background maps hamper the model’s abilities to segment the objects (just like in the real world), and 4) the randomly sampled viewpoints prevent models from overfitting, unlike NMR where the same cameras and canonically oriented objects are used. The dataset is available at srt-paper.github.io.

Street View. This is a new dataset created from real-world Street View data [10]. The training dataset contains 5.5 M scenes from San Francisco, each with 10 viewpoints. This dataset is very challenging, since the input viewpoints are typically several meters from the reference viewpoints, the cameras are often arranged in straight lines (along the car trajectory, meaning there is not a lot of diversity in viewing angles), the scenes contain challenging geometry (e.g., trees and thin poles) and dynamic elements (e.g., moving vehicles and pedestrians), the cameras have fisheye distortion and rolling shutter (making reprojection-based methods such as [49, 41, 44] inapplicable), and finally, different images often contain different exposure and white-balance settings.

4.2. Comparisons to Baselines

In the first set of experiments, we compare the results of SRT to previous methods for novel view synthesis. For these experiments, we restrict our tests to the NMR and MSN datasets, since they provide an opportunity for direct comparisons to recently published methods in a synthetic setting with known ground truth. We run SRT with the same parameters on both datasets.

Baselines. We provide comparisons to two baselines that share aspects of our approach. The first baseline is PixelNeRF [49], which also pretrains a network to produce features from images. The second baseline is Light Field Networks (LFN) [38], which like our method uses a light field formulation rather than volumetric rendering. These methods have demonstrated favorable quantitative and qualitative results in comparison to other previous work [41, 37, 28, 22].

Quantitative Results – Tabs. 2 and 3. SRT provides the best image quality on NMR, outperforming LFN and PixelNeRF. In terms of computational performance (Tab. 3), both PixelNeRF and SRT are fast to encode new scenes, while LFN requires slow optimization of the scene embedding. Once a scene is encoded, both LFN and SRT render new frames at interactive rates, while PixelNeRF is significantly slower due to the volumetric rendering. SRT is the only method fast at both scene encoding and novel-view generation, making it suitable for the practical application of rendering videos of novel scenes at interactive rates. See Appendix A.2 further details. We note that SRT is comparably small for a transformer: its 74 M parameters consist of 23 M for CNN, 47 M for Encoder, and 4 M for Decoder. For comparison, ViT [6] has 86 - 632 M parameters.

On the MSN dataset, SRT outperforms LFN and PixelNeRF in terms of PSNR and SSIM. In terms of LPIPS, PixelNeRF outperforms SRT, which we attribute to the fact that SRT tends to blur regions of uncertainty. In a separate experiment, we tried increasing the latent dimension of LFN (from 256 to 1024) to account for more complex scenes, but this did not increase PSNR on novel scenes.

Qualitative Results – Figs. 3 and 4. The results show that our method outperforms both baselines on the NMR dataset. While all methods are able to synthesize high quality images for nearby views (Fig. 3, center), PixelNeRF renders degrade particularly quickly as the camera moves away from the input views (Fig. 3, bottom). LFN render quality degrades similarly, and it is also blurry for more complex objects. In contrast, SRT provides similar performance for a wide range of views. The MSN dataset (Fig. 4) with its complex scenes clearly demonstrates the limits of LFN’s single-latent scene representation. While PixelNeRF results in detailed reconstructions for target views close to the inputs (Fig. 4 middle), SRT is the only method that allows for good reconstructions of far-away target views (Fig. 4 bottom).

Model Inspection – Fig. 5. Further insight can be gleaned...
by investigating the attention weights of the encoder and decoder transformers. We can see from Fig. 5 (top) that the patch with the small chair (marked with a green box) mostly attends into the same chair in the other input images, even when the chair is facing different directions. We further noticed that all patches additionally attend into a subset of patches positioned along the edges of the images (bottom corners of the first input image). We believe that the encoder learns to store global information in those specific patches.

A glance at the decoder supports this conjecture. When rendering a ray of the chair from a novel direction, the first attention layer only attends into the bottom edge of all input views, while the second layer attends into similar patches as the encoder, and into the bottom corners of the first input image. This learned 2-stage inference in the decoder appears to be crucial, as we noticed in prior experiments that a 1-layer decoder is significantly worse, while more than two layers do not lead to significant gains. The model appears to have learned a hybrid global/local conditioning pattern through back propagation, without explicit geometric projections.

### 4.3. Ablation Studies

We now perform a series of ablation studies on the MSN dataset, where we remove or substitute the main components of SRT and measure the change in performance to evaluate individual contributions of different design decisions.

**No Encoder Transformer.** Retraining the system without an encoder transformer (i.e., the decoder attends directly into the output of the CNN) provides a significant drop from 23.41 to 21.64 PSNR. This suggests that the encoder transformer adds crucial capacity for inference, and that the choice to move compute from the decoder (which should be as small and fast as possible) to the encoder is a feasible design decision. Note that its computational overhead is negligible in most practical settings, as it is only a single feed-forward step, and only run once per novel scene, independent of the number of rendered frames.

**Flat Latent Scene Representation.** One of the main novelties of SRT is the set-latent scene representation. We investigate the more commonly used flat latent by feeding the mean patch embedding of the encoder transformer into a large 8-layer MLP and dropping the decoder transformer. This architecture leads to a significant drop from 23.41 down to 20.88 in PSNR, showing the strength of a large set scene representation along with an attention-decoder.

**Volumetric Rendering.** As an alternative to the light field formulation, we also investigate a volumetric parametrization. To this end, we simply query the decoder not with rays, but rather with 3D points, followed by the volumetric rendering [25]. For simplicity, we do not inject viewing directions, and only use a single coarse network. We call this variation V-SRT. While this variation leads to an explicit 3D volume with easy-to-visualize depth maps, and results are visually comparable to SRT, it is to be mentioned that the V-SRT’s decoding is slower by a theoretical factor of 192×, the number of samples per ray, making inference time of this variation more similar to existing volumetric methods [49]. It is notable that our model architecture can be trained in both setups without further changes to architecture or hyper-parameters. See details in Appendix A.3.

### 4.4. Robustness Study

While perfect pose information is available in synthetic setups, real-world applications often depend on estimated camera poses (e.g., [35]), which is slow and often contains errors [21]. We therefore study SRT and baselines in a...
which contains images of real-world outdoor environments. To compensate for variations in appearance and exposure information about the scene, we augment the network with an appearance encoder, see Appendix A.2 for details.

**View Interpolation.** The goal in this application is to interpolate images captured in Street View panoramas to provide smooth video transitions between them. This application requires not only accurate view synthesis, but also temporal coherence between nearby views. We show a representative sample of interpolated views in Fig. 7. Although the renders are blurrier than the input views, our results demonstrate that SRT scales to complex real-world scenes with nontrivial camera pose distributions. It also shows that the model learns enough 3D scene information to render novel views far away from the inputs (last column). We provide videos and further results showing temporal coherence in render videos in Appendix A.3.

**Semantic Segmentation.** Our goal in this application is to predict dense semantics for novel views of outdoor scenes. Rather than synthesizing the color images for novel views and then employing a 2D image segmentation network on them, we show that SRT’s scene representation trained on RGB can be leveraged more directly in new domains. Once SRT has been trained for the RGB reconstruction task, we freeze the encoder, and train a new decoder transformer to synthesize semantic segmentation images from the frozen scene representation directly. The semantic decoder has the same network architecture as the color decoder, except for the final output layer that is changed from 3 RGB channels to 46 semantic classes. We train the semantic decoder using the standard multi-class cross-entropy loss in a semi-supervised setup, see Appendix A.2.

Example semantic segmentation results are shown in Fig. 7. These results suggest that the scene representation learned through a color reconstruction loss contains enough information about the scene to allow semantic reasoning.
Figure 7. **Qualitative results on Street View** – SRT performs reasonably on highly challenging real-world data with small and large changes in camera perspective. Furthermore, the scene representation contains enough information for 3D semantic scene inference. A comparison with NeRF-based optimization methods is provided in Appendix A.3.

Though this is just one example of many possible downstream tasks, it is a significant finding that SRT has learned a scene representation useful for a non-trivial application.

5. Limitations

This project investigates whether transformers can learn scalable scene representations from only images without explicit geometry processing. We here identify a few limitations, which we believe to be tackled by follow up works.

First, our results show some blurriness in images on the complex datasets, even for views close to the input cameras. This is expected, as the model needs to learn pixel-accurate light ray transformations, which leads to uncertainty about exact positions, known to result in blurriness under an L2 loss [33].

Second, SRT is a geometry-free learning-based method, and hence will not work as well on very small datasets compared to methods with explicit geometric inductive biases. In practice, we find that our model converges at rates similar to others, but even more training usually results in better performance. In particular, when trained on the standard datasets reported in this paper, SRT outperforms LFN and PixelNeRF. Further investigation is required to better understand the best training protocols.

Finally, in comparisons to prior art, our model is best when input views are sparse or when camera poses are noisy or missing. When novel views are reliably close to input views, and perfect poses are known, explicit geometric methods such as [49] often provide better results, albeit with much longer inference times. While SRT has been specifically designed for the sparse input scenario, future work could investigate how to better leverage nearby input views at inference time when they are available.

6. Conclusion

We propose Scene Representation Transformer, a model for learning scalable neural scene representations using only self-supervision from color images. The novel encoder-decoder transformer architecture learns to render novel views without explicit geometric reasoning, and optionally without 3D image poses, yet surpassing the quality of prior art on standard benchmarks and a novel, more challenging dataset that we propose. At the same time, SRT is orders of magnitudes faster than prior methods for the realistic settings such as novel-scene video generation. This unique combination of flexibility, generality, and efficiency is well-suited for real world applications with very large datasets, and we believe that this work will inspire the community towards similar more implicit, yet scalable methods.

7. Acknowledgments

We thank Adam Kosiorek, Alex Yu, Alexander A. Kolesnikov, Aravindh Mahendran, Luke Barrington, Konstantinos Rematas, Sebastian Ebert, Srinadh Bhojanapalli Vickie Ye, and Vincent Sitzmann for their help and fruitful discussions.
References


