This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

ERUPT: Efficient Rendering with Unposed Patch Transformer

Maxim V. Shugaev* Vincent Chen* Maxim Karrenbach†
Kyle Ashley* Bridget Kennedy* Naresh P. Cuntoor*
*BlueHalo †Carnegie Mellon University
maxim.shugaev@bluehalo.com

Abstract

This work addresses the problem of novel view synthesis in diverse scenes from small collections of RGB images. We propose ERUPT (Efficient Rendering with Unposed Patch Transformer) a state-of-the-art scene reconstruction model capable of efficient scene rendering using unposed imagery. We introduce patch-based querying, in contrast to existing pixel-based queries, to reduce the compute required to render a target view. This makes our model highly efficient both during training and at inference, capable of rendering at 600 fps on commercial hardware. Notably, our model is designed to use a learned latent camera pose which allows for training using unposed targets in datasets with sparse or inaccurate ground truth camera pose. We show that our approach can generalize on large real-world data and introduce a new benchmark dataset (MSVS-1M) for latent view synthesis using street-view imagery collected from Mapillary. In contrast to NeRF and Gaussian Splatting, which require dense imagery and precise metadata, ERUPT can render novel views of arbitrary scenes with as few as five unposed input images. ERUPT achieves better rendered image quality than current state-of-the-art methods for unposed image synthesis tasks, reduces labeled data requirements by 95% and decreases computational requirements by an order of magnitude, providing efficient novel view synthesis for diverse real-world scenes.

1. Introduction

Recovering 3D scene representation from a few images is a fundamental challenge in computer vision with wide-reaching applications in navigation, robotics, and augmented reality. Recent advances in learning implicit (*e.g.* NeRF [21]) and explicit (*e.g.* Gaussian Splatting [13]) scene representation have shown impressive capability in rendering high-fidelity, novel views of complex environments. These models typically rely on dense imagery with precise camera pose to reconstruct highly accurate models which can render novel views of the scene. However, one problem, which is central to both the NeRF and Gaussian Splatting frameworks, is that a new network or a set of Gaussians must be trained for each scene. This requires significant training time and limits the model's ability to incorporate prior knowledge in cases such as limited input views or large camera shifts. Several methods have been introduced to reduce this scene dependency, by including depth [8], normal [32], semantic [5] and other priors to use information from external data as a basis for learning a particular scene. Yet the problem remains that these frameworks are generally unable to perform novel view synthesis without training on a sufficiently large number of input views and associated camera information for each new scene.

More recent work has explored techniques for generalized 3D reconstruction with limited imagery using a latent scene representations transformer (SRT) [28] and learnable pose estimator (RUST) [27] to reduce reliance on precise camera poses. By training on large and diverse datasets these models learn to reconstruct arbitrary scenes using fewer images and limited camera pose information.

While SRT allows images to be queried directly using an arbitrary camera pose, it does not provide a mechanism for learning from unposed target views. This makes it difficult to apply in datasets where ground truth camera pose is noisy or unknown. RUST addresses this limitation using a learnable pose estimator but requires a portion of the target image to query the model, which limits control of the camera at inference. Both methods require significant compute during training and inference using individual queries to the decoder for each pixel in the rendered images. Furthermore, existing benchmark datasets of sufficient scale are either composed of synthetic data (MSN) or are proprietary datasets not available to the public, making verification and real-world application of these techniques difficult.

To address these limitations we propose ERUPT, a computationally effective approach for novel view synthesis trained on posed and unposed target images, providing direct camera control at inference. Our model includes a learnable latent camera pose which is sampled in conjunction with sparse ground truth pose to address cases of sparse

or inaccurate target camera pose. We introduce a novel decoding strategy using patch-based rays instead of pixel-based rays to improve computation efficiency of rendering by an order of magnitude over existing models. To accelerate convergence, during training we generate multiple targets reusing scene representation and apply a pre-trained feature extractor for transfer learning. We compare our model with existing methods using the MSN benchmark dataset, achieving state-of-the-art performance even compared to methods which rely on 100% posed target images. Additionally, we introduce a new real-world view synthesis dataset called MSVS-1M (Mapillary Street-view Synthesis 1-million) to allow future studies to conduct direct comparisons on real-world data.

In summary our contributions include:

1. An efficient network architecture and training regime for latent-based novel view synthesis trained on posed and unposed target images which provides direct camera control at inference. We show that training on data with only 5% posed targets results in minimal performance degradation compared to the baseline with target poses provided.
2. Introduction of patch-based rays in the decoder instead of pixel-based rays results in an order of magnitude reduction in rendering and training time as well as VRAM requirements. Specifically, the synthesis of 224×224 images can be performed at a rate of over 600 fps.
3. An optimized model architecture for image synthesis with unknown input pose. The produced models establish a new SOTA for image synthesis on MSN dataset with unknown input pose in terms of PSNR and SSIM.
4. Using ERUPT as a foundation, we investigate the use of adversarial learning and diffusion-based rendering for latent novel view synthesis. Both approaches drastically improve the perceptual quality of the output and outperform previous perception based SOTA on MSN dataset.
5. A new multi-scene view synthesis dataset, MSVS-1M, curated from Mapillary imagery to facilitate large-scale experiments in generalized real-world view synthesis.

2. Related Work

2.1. NeRF and Gaussian Splatting

In the realm of novel view synthesis, two main frameworks have stood out at the forefront. Neural Radiance Fields (NeRF) [21] has approached the problem by predicting the color and density of points in a 3D representation space through neural networks as an implicit representation. By modifying the neural network and training scheme, this framework can serve as a base for further works, like object editing/removal [14], appearance or lighting editing [20], or dynamic scene synthesis [2]. More recently, 3D Gaussian Splatting [13] has been proposed and shown to reduce the high rendering time cost of NeRFs at the cost of requiring most detailed hyperparameters. The 3DGS framework uses an explicit representation of the 3D scene based on 3D Gaussians with spherical harmonics defining the color and opacity. The low rendering time is a draw to this type of framework for inference-heavy applications, and the granularity of the 3D Gaussians allow for techniques such as per-Gaussian editing [3], Gaussian grouping [40], or appearance conditioning [6].

Both frameworks have been built upon by numerous other works to reduce limitations including reduced data dependency, improved rendering quality and optimizations for rendering speed. For example, frameworks like Sparse-GS [39] and SparseNeRF [36] have been developed to address performance degradation due to limited number of training views by using alternate strategies like depth information. To address the problems of large training times, both frameworks have implemented techniques, like the NeRF-based multi-resolution hash encodings in Instant-NGP [22] or 3DGS-based vector quantization [23] in Compact3D. These frameworks have even been combined to get the best of both techniques via hybrid models [18] called Viewing Direction Gaussian Splatting, which uses the Gaussian distributions and includes a neural network which interprets the Gaussian parameters and viewing directions.

However, one problem which is central to both the NeRF and 3DGS frameworks is that a new network or a new set of Gaussians must be trained and fitted for each scene, requiring training time. Works have indeed been done to reduce this scene dependency, by including depth [8], normal [32], semantic [5] and other priors to use information from external data as a basis for learning a particular scene. Yet the problem remains in that these frameworks are generally unable to perform novel view synthesis without training on numerous input views with precise metadata provided, which results in a necessary time to train on a novel scene.

2.2. Posed and unposed SRT

Latent scene representation approaches mitigate limitations of NeRF and Gaussian Splatting on transferring prior knowledge by considering a generalized encoder-decoder architecture. This model, after training on a large corpus of scenes, creates a scene representation based on the provided input images and allows rendering novel views without the need to retrain the model for the particular scene. One of the early works that has demonstrated the scalability of the latent scene representation approach to complex scenes is the Scene Representation Transformer (SRT) [28]. The proposed model consists of a light CNN applied to input images and the corresponding input poses and creates a set of tokens. These tokens are then passed to a scene representation transformer. The transformer performs mixing of all input views and creates a latent scene representation.

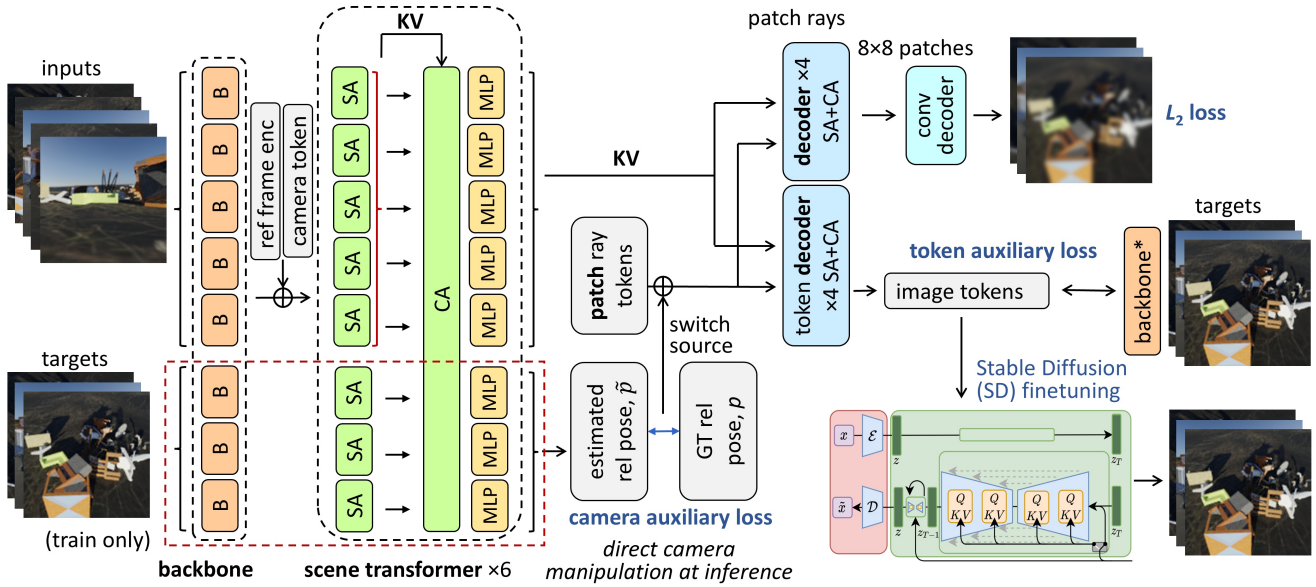


Figure 1. Schematic illustration of ERUPT model. B refers to backbone (feature extractor), while SA and CA refer to self- and cross-attention.

Finally, this representation is queried in a set of decoder blocks based on rays corresponding to each target pixel. In [12] the architecture was extended to decoder only. In several studies [37, 41] the NeRF framework is extended with learnable mechanism to enable novel view synthesis based on a few or even single input image. However, they have limitations or under-perform SRT [28]. The SRT model requires knowledge of the camera parameters for all input and target images at training and inference. This limitation is partially addressed in UpSRT that uses known pose only for the first input image and the target. However, as illustrated in [27], training on perturbed target poses (which corresponds to the realistic setup of using estimated target poses) results in significant performance degradation.

2.3. RUST

To address the limitation of SRT work on training on scenes with unknown target poses, Latent Neural Scene Representations from Unposed Imagery (RUST) [27] model was proposed. In this model, the UpSRT setup is supplemented with an additional pose estimation module that takes half of the target image and the latent scene representation to compute a latent pose (8-dimensional vector), which is used to query the scene representation in the decoder (instead of the real target pose) and generate the remaining half of the target image. This setup enables robust training on completely unposed imagery sets. Generation of the particular target view at inference still requires half of the target image, which limits the views that can be generated. Although, [27] demonstrates the capability of RUST to perform shifts and rotations of the camera by manipulating principal components of the latent pose vector, precise camera control is challenging and may involve manual tuning steps.

2.4. DORSal

Diffusion for Object-Centric Representation of Scenes (DORSal) [11] is a recent work that considers a diffusion decoder to perform scene rendering based on the scene representation created with a frozen Object Scene Representation Transformer (OSRT) [26]. Use of the diffusion model drastically improves the quality of the generated scenes improving FID by 5-10×.

3. Methodology

3.1. Model

Fig. 1 schematically illustrates the ERUPT model developed in our work. The input scene consists of an unordered set of (unposed) views, which are processed with a transformer-based feature extractor. A reference frame encoding and positional encoding are added to the collection of image features, and the resulting set of tokens is expanded with a learnable camera token. Next, the token sequence is processed with a scene transformer designed to generate compact scene and camera representations. This transformer alternates token mixing within each image and between images within the entire scene, producing the scene representation and camera pose estimation for each image (which are computed with respect to the reference frame). The resulting scene representation is queried for each patch-ray in the image decoder and token decoder,

which produce the image output and a sequence of tokens matching the target image. Using patch rays instead of pixel rays drastically improves the computational efficiency and enables the use of the generated sequence of tokens to condition a fine-tuned Stable Diffusion (SD) model [25].

At the training stage, a set of target images is also passed through the feature extractor and scene transformer to estimate the target camera parameters. Then the query camera parameters are selected from the estimated and the ground truth poses to maintain the RUST advantage of training on unposed images or images with inaccurate camera poses and at the same time to have the ability of direct camera manipulation at inference. Below we discuss each part of the model in details.

Scene transformer: The objective of the scene transformer is to extract a compact representation of the scene from a small set of images which is flexible enough to function on diverse scenes. In the SRT setup proposed in [28], camera information is attached to each image which enables effective discrimination of patches from different input views and scene understanding. However, this model architecture cannot be easily adapted to the case of unknown camera parameters of input images. The approach proposed in RUST [27] applies a similar transformer setup to SRT except, instead of encoding the camera parameters, a simple camera encoding is used to distinguish the reference frame from the remaining input images, *i.e.* all non-reference images are not distinguishable in the model, and scene understanding is limited. To address this fundamental problem we have redesigned the architecture of the scene transformer. Specifically, we alternate mixing of tokens within images (self-attention between image tokens) with mixing of tokens between images for the entire scene (cross-attention of image tokens and all scene tokens), *i.e.* the model while exchanging the information between input views has a capability to distinguish tokens from different images and build the camera representation for each input even if the camera parameters are unknown. This design alleviates the limitations of simple token mixing used by existing approaches to build a robust scene representation for unposed inputs. The scene transformer consists of 6 transformer blocks in all experiments.

Pose estimation: Our scene transformer network enables direct camera parameter estimation for each image without including a separate model branch (as used in RUST). We add a learnable camera token to each image to aggregate the camera information and the relative camera pose of each image to the reference. An auxiliary loss is applied on the learnable camera pose that guides the camera estimation and may improve the scene understanding. In our setup we use L_2 loss applied to the relative camera position and negative cosine loss applied to 3D vectors describing the relative camera view. To avoid information leakage from the target images, the cross-attention part of the scene transformer uses the combination of tokens from input images only as the key-value (KV) input.

Target camera switching: A major contribution of our method is the ability to query images from the model in a fully-controllable manner using arbitrary target poses. In contrast, RUST uses pose estimation on half of the target image followed by an estimated latent pose to render the remaining part. While this enables training and inference on input images with unknown camera pose, it requires a portion of the target view to be available at inference time which limits the model to only views which have been partially observed. This issue is discussed in the RUST paper in detail, and the use of several principal components of the latent pose is suggested to describe a particular motion of the camera. However, precise camera manipulation is challenging and may involve manual tuning steps for each model.

We introduce the ability to manipulate the target camera at inference time while maintaining the advantage of training on unposed images. To accomplish this, during training, we concatenate the estimated latent pose with sine encoded scaled camera parameters [28, 33] and randomly sample between three modes (1) providing only the estimated latent camera pose, (2) providing only the ground truth camera pose and (3) providing both estimated and ground truth camera pose. These modes are sampled evenly with 1/3 probability during training, unless otherwise specified. At inference we use only the encoded target image pose, which enables direct camera control. Meanwhile, use of the estimated latent camera pose pathway may be helpful for training on images without known pose or with inaccurate camera pose estimation: when the target view contradicts the provided pose, the model may use the latent pose pathway to generate the expected output and enable robust training. We conducted an ablation study to assess the robustness of the approach when highly limited ground truth is available and found that using just 5% ground truth poses resulted in minimal degradation (Tab. 2).

Patch based decoder: Existing latent scene representation models require significant computation for both training and inference. SRT and RUST use pixel-based decoding, *i.e.* querying the scene representation for each pixel, which introduces an enormous overhead both in terms of the speed and VRAM consumption, as discussed in the section on the computational performance. To improve the computational performance we use patch base rays with patch size of 8×8 for the image decoder and patch size matching the backbone for the token decoder. Using this method ERUPT recovers 64 pixels per query instead of 1 pixel as done in pixel-based decoding models.

The patch ray queries are learned during training and initialized with sine positional encoding [33]. The target

camera representation is concatenated to each query token. Each decoder is composed of 4 standard transformer decoder blocks [33] alternating self-attention within ray queries and cross-attention with the scene representation. The image decoder is followed with 3 convolutional pixel unshuffle [29] up-sampling blocks producing the final image. L_2 loss is applied to the pixel output. In addition to the image decoder, we use token decoder that is trained to match the embedding produced by the pretrained backbone on the target image, *i.e.* enforce the semantics of the generated output. ArcGeo auxiliary loss [30] is applied to each token of the produced embedding within the target image.

Feature extractor: Both SRT and RUST use a simple convolutional encoder trained together with the rest of the model to extract image tokens, while in [34] ImageNet initialized ResNet18 is applied. We follow the idea of using a pretrained model to extract reliable features and accelerate training. In our setup we upscale input images to 224 size and apply DINOv2 [24] as a feature extractor. We use DINOv2 with additional registers, which is demonstrated to have better performance and feature map interpretability [7]. This model is suggested to have a robust single image 3D understanding as well as multi-view consistency [4], which may be helpful for 3d scene synthesis. The produced feature map has 16×16 dimensions (similar to RUST) plus 1+4 extra classification and register tokens. Learnable positional encoding is added to the produced sequence of tokens before feeding to the scene transformer together with the reference frame encoding. We freeze the backbone in training and perform an additional set of experiments with enabling LORA [10] fine-tuning of the backbone after the initial model training. The computational cost of the forward pass for DINOv2 B model is comparable to the forward and backward pass cost for the extractor used in SRT and RUST, and use of the pretrained model introduces a minimum overhead during training.

GAN setup: In several experiments we performed fine-tuning the models to improve the perceptual quality of the output. Generative adversarial networks (GANs) has been successfully leveraged in related applications of image generation [15, 16] to improve realism and align distributions of synthesized imagery. We explored their impact on latent view synthesis in ERUPT, and follow existing adversarial formulations [16]. We replaced L_2 image output loss with a combination of L_1, perception, and GAN losses. We use the SwinIR [16] configuration from *KAIR* repository [42] as a starting point.

Stable Diffusion based rendering: ERUPT's patch-based decoder not only improves computational efficiency but also enables the use of pretrained generative models, *e.g.* Stable Diffusion (SD) [25], as a rendering tool. Specifically, in a series of experiments we applied SD 2.1 from the *Diffusers* repository [35] to perform the final step of image generation instead of image decoder. In this setup we freeze the ERUPT model and use the token decoder output with an additional positional encoding as a prompt to the diffusion model. Then we perform a full fine-tuning of SD U-Net for 20 epochs on the considered dataset with rescaling the target images to 512×512. SD model is quite flexible to prompting adjustment, and relatively short fine-tuning is sufficient to enable generation even based on non-CLIP tokens.

Model Size: We consider two ERUPT models sizes. Our ERUPT-B model uses embedding dimension of 768 and contains a comparable number of trainable parameters to the 74M SRT model [28]. Specifically, the scene transformer and image decoder contain 57.4M and 15M parameters, respectively (72.4M in total). In addition, ERUPT-B contains a 85.7M parameter frozen DINOv2-B feature extractor and a 38.9M parameter token decoder, which may be omitted at the inference with image decoder. ERUPT-L uses embedding dimension of 1024 and contains 102M and 23.9M scene transformer and image decoder (125.9M in total). In addition, this model uses 194.9M parameter frozen DINOv2-L extractor and 68.9M parameter token decoder.

3.2. Data

MSN dataset (MSN) is a novel view synthesis dataset introduced in [28] containing 1M train and 100k test scenes. Each scene consists of 10 128×128 images representing viewpoints selected by uniformly sampling within a half-sphere shell. This dataset is created with photorealistic ray tracing, and each scene has 16-31 ShapeNet objects dropped at random locations within an invisible bounding box. The dataset includes precise camera information. In the standard training/inference setup the model receives 5 randomly selected input views per scene, and the target is sampled from the remaining views.

MSVS-1M dataset [1] is a collection of real-world panoramic images captured by a vehicle along its trajectory [38]. This dataset aims to tackle real-world challenges not captured by synthetic MSN data such as noisy camera pose, lighting/semantic variability, and diverse environments. The data is similar to the private StreetView dataset used in [27, 28], but is collected through publicly available sources and released to the community under Mapillary's CC-BY-SA license. To create the dataset, we randomly selected 10 locations in the world and sampled street view panoramas within 60 kilometers of each location. We selected only images with corrected camera parameters obtained through Mapillary's structure-from-motion process [19] and performed filtering to reduce the fraction of blurry images and images with rain droplets. We chose image sequences with a distance of 0.5-10 m between image acquisition location and a total length of at least 6. The number of filtered images is approximately 1M, corresponding to 32k sequences. At training/evaluation we perform dynamic

scene generation with details provided in the Supplementary Materials.

3.3. Training Setup

We train our models with AdamW [17] optimizer for 160 epochs with the maximum learning rate of 5×10^{-4} and cosine schedule (10k warmup steps). The batch size is 128, and weight decay is set to 0.05. We use 5 target images per scene (if other values are not specified) to reuse the scene representation and improve the efficiency of training, even if the time per epoch is approximately doubled compared to the single target setup. Our models exhibit competitive results while training is more than 5 times shorter than 1000 epochs used in SRT setup [28].

In LORA fine-tuning [10] we added r=32 adapters to qkv and output projection in attention and to MLP layers of the frozen DINOv2 backbone, enabling adjustment of the model to the task of 3D scene reconstruction and to synthetic MSN input. LORA added 5M parameters to B and 13M parameters to L model. As the starting point we used a model trained with a frozen feature extractor and continued training for 32 additional epochs with LORA and the maximum learning rate of 2×10^{-5}.

In GAN experiments we merged LORA with the frozen backbone and trained ERUPT for an additional 64 epochs (with a single target image), adding perception and GAN losses while using a maximum learning rate of 10^{-5}. In SD experiments we performed full fine-tuning of SD U-Net (while ERUPT is frozen) for 20 epochs with the maximum learning rate of 10^{-5}.

4. Experiments

Method		PSNR↑	SSIM↑	LPIPS↓	FID↓
SRT	p_x p_y	23.41	0.697	0.369	-
SRT*	p_x p_y	**25.93**	-	0.237	67.29
UpSRT	~~p_x~~ p_y	23.03	0.683	0.362	-
RUST	~~p_x~~ ~~p_y~~	23.49	0.703	0.351	-
DORSal	p_x p_y	18.99	-	0.265	9.00
ERUPT (Ours):					
B+LORA	~~p_x~~ p_y	24.69	0.749	0.360	99.7
L+LORA	~~p_x~~ p_y	**25.26**	**0.769**	0.340	91.1
B+GAN	~~p_x~~ p_y	23.38	0.713	**0.204**	7.45
B+SD	~~p_x~~ p_y	21.06	0.637	0.234	**6.89**

Table 1. Comparison of ERUPT to latent scene representation methods on MSN dataset. p_x,p_y refer to input and target poses. Colors: green - available pose, red - not available pose, orange (ERUPT) - sparse camera pose at training and available pose at inference. SRT metrics are reported in [28]; UpSRT and RUST in [27]; and SRT* and DORSal in [11].

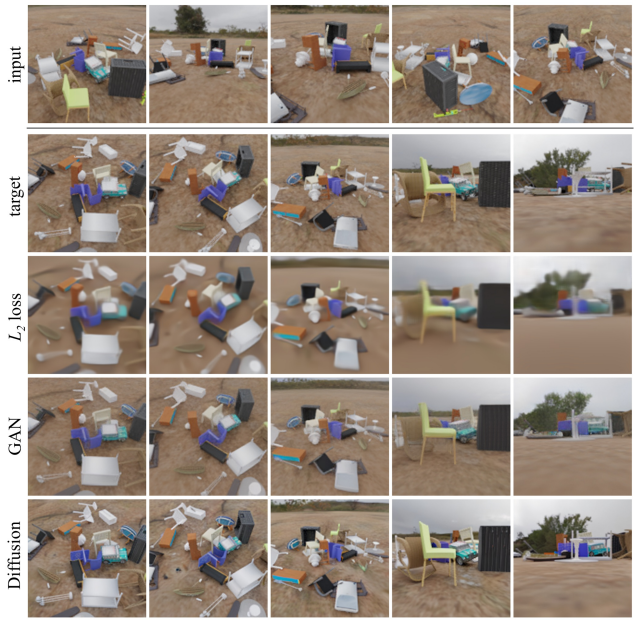


Figure 2. Qualitative results on MSN: input scene (5 images), target views (5 images), ERUPT B + LORA, ERUPT B + GAN, ERUPT B + Diffusion.

4.1. Novel view synthesis

We compare ERUPT with several existing methods. SRT provides a reference for synthesizing novel views under ideal conditions with known ground truth camera parameters both for input and target images and uses a transformer-based encoder decoder model. The original SRT architecture is improved in the later publications [26, 27]. UpSRT and RUST are included in our evaluation to demonstrate results for cases with unknown input image pose and known/unknown target pose, while DORSal showcases diffusion-based model performance. Quantitative results are summarized in Tab. 1.

ERUPT B and L with LORA finetuned feature extractor strongly outperforms UpSRT/RUST in terms of PSNR/SSIM and reaches the performance close to the optimized SRT despite of unknown input poses and mixture of known and unknown target poses used in training. The perceptual quality of the output is comparable to unoptimized SRT, UpSRT, and RUST, and the larger gap between PSNR and LPIPS may be attributed to use of patch-ray instead of pixel-ray based decoder. While our model trained with simple L_2 loss yields blurry images due to high uncertainty of the scene content, it can be significantly improved after a short fine-tuning in a GAN based setup, outperforming the perceptual quality reported in literature on MSN dataset. Both, GAN and diffusion setups outperform DORSal (the current SOTA on MSN) in terms of FID. Synthesized images generated using various ERUPT configurations are il-

lustrated in Fig. 2. Despite the superior visual quality of diffusion based output, it has only slightly better FID than the GAN setup, which may miss fine details. Finally, in contrast to DORSal, each output frame is synthesized independently on each other, which may result in inconsistency. Despite this limitation, the model is able to produce coherent scene output even if the particular item is presented in 1-2 input views, as a car in Fig. 2. Further details on the model output are provided in the supplementary materials.

Method	PSNR↑	SSIM↑	LPIPS↓	FID↓
baseline	23.85	0.718	0.382	105
Single target	23.43	0.700	0.401	115
Patch RUST	23.20	0.690	0.411	119
ERUPT B+LORA	24.69	0.749	0.360	99.7
ERUPT L	24.29	0.735	0.367	96.4
ERUPT L+LORA	25.26	0.769	0.349	91.1
5% known poses	23.55	0.706	0.392	108

Table 2. Ablation results on MSN dataset. *baseline* refers to ERUPT B.

4.2. Ablation Results

We now show results from a series of ablation studies on the MSN dataset in Tab. 2. As a baseline ERUPT-B setup is used. Reduction of the number of target images from 5 to 1 in training reduces PSNR from 23.85 to 23.43 that suggests reusing the scene representation to render multiple targets may noticeably improve the training efficiency. Next, we compare the baseline model architecture and an architecture using a separate pose estimation branch and simple token mixing (similar to RUST), which is modified to use a DINOv2 feature extractor and patch-based decoders. This configuration performs comparably to the original RUST [27] while having significantly better computational speed. LORA based fine-tuning of the feature extractor provides a significant performance boost, suggesting that even a robust foundational model as DINOv2 may need adjustment to extract features for 3D scene synthesis task. Increase of the model size from B to L significantly improves the performance suggesting that scaling up ERUPT model may further improve the performance.

Finally, we examine the case where only 5% of target images retain ground truth poses during training, where learning is primarily driven by the estimated pose. At inference, meanwhile, we evaluate the performance on synthesis conditioned by the ground truth pose rather than by the estimated pose. We observe only a marginal decrease in performance compared to the baseline, while maintaining the ability to manipulate the camera at inference. UpSRT [28], which similar to ERUPT provides camera control at inference and uses unposed input, exhibits a significant performance degradation to 18.64 PSNR when trained on noisy poses [27]. RUST [27] trained on unposed images, meanwhile, also reports the ability to manipulate the camera using latent pose instead of ground truth target image, but the corresponding performance is not evaluated.

Method	res	F *(ms)*		$F + B$ (ms)
		s	d	$s + d$
ERUPT B	224	6.0 (2.9)	1.5	15
ERUPT L	224	14 (9.4)	2.3	27
SRT	128	3.0 (1.0)	4.7	27(2.6)
RUST & late SRT	128	5.3 (0.8)	14	73(2.3)
SRT	224	11	49	168
RUST & late SRT	224	30	118	OOM

Table 3. The time of forward and backward pass per scene. F, B, s, d stands for forward, backward, scene, and decode, respectively, while res refers to the image size. The time spent by the feature extractor is listed in parentheses.

4.3. Computational Performance

The computational performance of ERUPT is summarized in Tab. 3. We analyze the time needed to perform forward and backward passes for scene representation generation ('scene') and single image generation ('decode') on the Ada A6000 GPU. We exclude SD based rendering, which provides high quality output but at about 1 fps for 512x512 resolution. The batch size is selected to maximize VRAM usage and the time measurements are normalized to provide per-scene values. Mixed precision is enabled in all experiments. We compare our method with the original SRT [28] and late SRT/RUST [27] architectures. Since the corresponding papers do not provide the code, we use the implementation from [31]. The late SRT/RUST setup has a small patch size of 8×8 and 5 scene transformer blocks instead of 10 in the original SRT.

The time spent for scene representation generation is comparable for both models, and the discrepancy is associated with a small patch size used by RUST. Meanwhile, the speed of the ERUPT decoder is an order of magnitude faster than that of RUST, depending on the output resolution. This is a result of computational efficiency gained using our patch-based decoder in comparison to pixel-based decoders used in RUST. The time of the forward and backward pass (corresponding to training speed) of ERUPT-B model at 224-pixel resolution is 5 times shorter than one for RUST at 128-pixel resolution. Finally, the use of frozen DINO-v2 encoder does not introduce a significant overhead during training compared to a simple convolutional extractor in RUST/SRT: 2.9 vs. 2.6 and 2.3 respectively.

The RUST model cannot fit on a single Ada A6000 GPU with 48 GB when using 224-pixel resolution even when

using a batch size of 1 because of insufficient VRAM (if all pixels in the image are generated). This is denoted as 'OOM' in Tab. 3. The VRAM required by ERUPT-B is approximately 16 times smaller using a 128-pixel configuration and 64 times smaller using a 224-pixel compared to corresponding RUST model requirements mainly due to ERUPT's memory friendly patch-based decoder.

4.4. Real scenes (Mapillary data from MSVS-1M)

In contrast to MSN, MSVS-1M [1] is a challenging real-world dataset which captures many of the typical challenges associated with "in-the-wild" data including high object variability, incomplete and discontinuous scenes, transient objects, and camera pose inaccuracy. These present a challenge to view synthesis methods designed strictly for pristine conditions, and better reflect the expected performance of such models in real operating cases. We present this dataset to the community with the hope that it will assist in development of robust view synthesis methods which are suitable for real image datasets. Quantitative results on street-view data from MSVS-1M are provided in Tab. 4, and an example of the model output is shown in Fig. 3.

Method	PSNR↑	SSIM↑	LPIPS↓	FID↓
ERUPT B+LORA	20.64	0.657	0.703	442
ERUPT B+GAN	21.07	0.639	0.727	465
ERUPT B+SD	17.10	0.506	0.371	12.6

Table 4. Quantitative results on MSVS-1M dataset.

As expected, our experiments with ERUPT on this data show that the overall model performance is lower than that reported for MSN dataset. Perceptual quality metrics are especially impacted in the LORA and GAN configurations which may be attributed to blurry images (in case of L_2 loss) and GAN artifacts. Interestingly, the ERUPT setup using Stable Diffusion (SD) based rendering, shows improved perceptual quality and good alignment to the ground truth.

4.5. Limitations

One limitation of our approach is its reliance on an L_2 objective during training. In the case of the MSN dataset, when the inputs typically fully define the scene, and the diversity of objects in the scene is limited, the L_2 objective enables effective training and provides competitive results. However, if generation of the target image involves out-painting areas not present in the input images or cases where input images have low overlap, the L_2 objective produces blurry output and may inhibit training from focusing on realistic image generation. Adversarial loss in our GAN-based setup partially addresses this issue, but still suffers from visual artifacts for highly diverse scenes like those in MSVS-1M. Use of diffusion models conditioned on ERUPT for rendering novel views drastically improves the quality of the output especially in the case of MSVS-1M data. However, under conditions of limited input, the diffusion model may produce inconsistent visual features between views. This limitation may be addressed with multi-view diffusion applied in DORSal [11]. Finally, under conditions of limited view our approach with L_2 objective is likely to under-perform setups trained fully on diffusion objective, *e.g.* recent CAT3d [9].



Figure 3. Qualitative results on MSVS-1M: input scene (5 images), target views (5 images), ERUPT B + LORA, ERUPT B + GAN, ERUPT B + Diffusion.

5. Conclusions

We present ERUPT, a latent view synthesis model designed for efficient unposed scene representation learning in highly unconstrained environments. Our novel pose-estimation strategy and patch-based encoder-decoder transformer architecture learns to efficiently render novel views, training on both posed and unposed images. We demonstrate an effective method for fully-controllable view synthesis, where target images can be directly queried by specifying camera pose relative to the input images. Our method achieves an order of magnitude improvement of computational efficiency while providing state-of-the-art accuracy in rendered novel views on benchmark datasets. Additionally, we introduce a large-scale, real-world scene reconstruction dataset containing 1M images from 32k sequences. We anticipate that our innovative and computationally efficient approach and open benchmark dataset will enable significant advancements in large-scale latent scene reconstruction.

6. Acknowledgments

Supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/ Interior Business Center (DOI/IBC) contract number 140D0423C0075. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/IBC, or the U.S. Government.

References

[1] Msvs-1m. https://huggingface.co/datasets/erupt/MSVS-1M. 5, 8

[2] Benjamin Attal, Eliot Laidlaw, Aaron Gokaslan, Changil Kim, Christian Richardt, James Tompkin, and Matthew O'Toole. Törf: Time-of-flight radiance fields for dynamic scene view synthesis. In *Advances in Neural Information Processing Systems*, volume 31, 2021. 2

[3] Jeongmin Bae, Seoha Kim, Youngsik Yun, Hahyun Lee, Gun Bang, and Youngjung Uh. Per-gaussian embedding-based deformation for deformable 3d gaussian splatting, 2024. 2

[4] Mohamed El Banani, Amit Raj, Kevis-Kokitsi Maninis, Abhishek Kar, Yuanzhen Li, Michael Rubinstein, Deqing Sun, Leonidas Guibas, Justin Johnson, and Varun Jampani. Probing the 3d awareness of visual foundation models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21795–21806, 2024. 5

[5] Chong Bao, Yinda Zhang, Bangbang Yang, Tianxing Fan, Zesong Yang, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Sine: Semantic-driven image-based nerf editing with prior-guided editing field. 2023. 1, 2

[6] Hiba Dahmani, Moussab Bennehar, Nathan Piasco, Luis Roldao, and Dzmitry Tsishkou. Swag: Splatting in the wild images with appearance-conditioned gaussians, 2024. 2

[7] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers, 2024. arXiv:2309.16588. 5

[8] Kangle Deng, Andrew Liu, Jun Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2022-June, 2022. 1, 2

[9] Ruiqi Gao*, Aleksander Holynski*, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul P. Srinivasan, Jonathan T. Barron, and Ben Poole*. Cat3d: Create anything in 3d with multi-view diffusion models. *Advances in Neural Information Processing Systems*, 2024. 8

[10] Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR 2022 - 10th International Conference on Learning Representations*, 2022. 5, 6

[11] Allan Jabri, Sjoerd van Steenkiste, Emiel Hoogeboom, Mehdi S. M. Sajjadi, and Thomas Kipf. Dorsal: Diffusion for object-centric representations of scenes et al, 2024. 3, 6, 8

[12] Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. Lvsm: A large view synthesis model with minimal 3d inductive bias, 2024. 3

[13] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42, 2023. 1, 2

[14] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2022-June, 2022. 2

[15] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network, 2017. 5

[16] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2021-October, 2021. 5

[17] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019*, 2019. 6

[18] Dawid Malarz, Weronika Smolak, Jacek Tabor, Sławomir Tadeja, and Przemysław Spurek. Gaussian splatting with nerf-based color and opacity, 2024. 2

[19] Mapillary. Opensfm. www.opensfm.org. 5

[20] Ricardo Martin-Brualla, Noha Radwan, Mehdi S.M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2021. 2

[21] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65(1):99–106, dec 2021. 1, 2

[22] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41, 2022. 2

[23] KL Navaneet, Kossar Pourahmadi Meibodi, Soroush Abbasi Koohpayegani, and Hamed Pirsiavash. Compact3d: Smaller and faster gaussian splatting with vector quantization, 2024. 2

[24] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. arXiv:2304.07193. 5

[25] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2022-June, 2022. 4, 5

[26] Mehdi S. M. Sajjadi, Daniel Duckworth, Aravindh Mahendran, Sjoerd van Steenkiste, Filip Pavetić, Mario Lučić, Leonidas J. Guibas, Klaus Greff, and Thomas Kipf. Object scene representation transformer, 2022. 3, 6

[27] Mehdi S.M. Sajjadi, Aravindh Mahendran, Thomas Kipf, Etienne Pot, Daniel Duckworth, Mario Lučić, and Klaus Greff. Rust: Latent neural scene representations from unposed imagery. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2023-June, 2023. 1, 3, 4, 5, 6, 7

[28] Mehdi S.M. Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lucic, Daniel Duckworth, Alexey Dosovitskiy, Jakob Uszkoreit, Thomas Funkhouser, and Andrea Tagliasacchi. Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2022-June, 2022. 1, 2, 3, 4, 5, 6, 7

[29] Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2016-December, 2016. 5

[30] M. Shugaev, I. Semenov, K. Ashley, M. Klaczynski, N. Cuntoor, M. Lee, and N. Jacobs. Arcgeo: Localizing limited field-of-view images using cross-view matching. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 208–217, Los Alamitos, CA, USA, jan 2024. IEEE Computer Society. 5

[31] stelzner. Srt: Scene representation transformer. https://github.com/stelzner/srt. 7

[32] Matias Turkulainen, Xuqian Ren, Iaroslav Melekhov, Otto Seiskari, Esa Rahtu, and Juho Kannala. Dn-splatter: Depth and normal priors for gaussian splatting and meshing, 2024. 1, 2

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 2017-December, 2017. 4, 5

[34] Naveen Venkat, Mayank Agarwal, Maneesh Singh, and Shubham Tulsiani. Geometry-biased transformers for novel view synthesis, 2023. arXiv:2301.04650. 5

[35] Patrick von Platen, Suraj Patil, and Anton Lozhkov. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 5

[36] Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*, 2023. 2

[37] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering, 2021. 3

[38] Frederik Warburg, Soren Hauberg, Manuel Lopez-Antequera, Pau Gargallo, Yubin Kuang, and Javier Civera. Mapillary street-level sequences: A dataset for lifelong place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 5

[39] Haolin Xiong, Sairisheek Muttukuru, Rishi Upadhyay, Pradyumna Chari, and Achuta Kadambi. Sparsegs: Real-time 360° sparse view synthesis using gaussian splatting. *Arxiv*, 2023. 2

[40] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes, 2024. 2

[41] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images, 2021. 3

[42] Kai Zhang. KAIR. https://github.com/cszn/KAIR/tree/master, 2022. 5