1. Supplements

1.1. 360-rotation illustration

Figure 1 provides examples of the model output, showcasing a sequence of images generated by 360-rotation around the scene. To give a complete view of ERUPT's scene understanding and synthesis capabilities, we consider two scenarios: one where all items are sufficiently represented in the input views (left), and a more challenging case where input views are far from target views and portions of the scene are not in view of any input imagery (right). When input views are sufficiently dense, ERUPT is able to accurately and consistently synthesize imagery for arbitrary target views after training using a simple L_2 objective. In the more challenging case where portions of the scene are missing, the L_2 objective is insufficient resulting in blurry output. Introduction of adversarial loss in our GAN configuration drastically improves the quality of the output eliminating uncertain blurry regions. However, missing information still presents a challenge to reconstruction, resulting in small inconsistencies and GAN artifacts. After finetuning using SD for rendering, challenging target views are reproduced with a high level of detail and fidelity. However, when using fine-tuned Stable Diffusion (SD) for rendering, each novel view is generated independently and uncertain regions may result in inconsistent output between different views.

1.2. Diffusion Output Consistency

The output of Diffusion-based models may widely change depending on the provided input noise, which is unfavorable for novel view synthesis. The conditioning provided by ERUPT embedding, however, is sufficient to guide image generation toward semantical and visual coherency. As depicted in Fig. 3 (main text), our approach enables consistent generation of high quality novel views, and in Fig. 2 we provide further illustration of the generation consistency for various input noise (defined by setting random seeds). The images are nearly identical with an insignificant difference in fine details. It should be pointed out, however, that if the input scene has blind spots, the SD output generated for these blind spots may vary.

1.3. Dynamic Sampling and Scene Creation with Mapillary MSVS-1M Data

To create a real dataset of sufficient scale to train our model, we collected a series of panoramic imagery for 10 real-world locations through the Mapillary API (e.g. https://tiles.mapillary.com). In this section we describe the data collection procedures and present results using ERUPT for novel view rendering on the proposed MSVS-1M dataset. The total number of filtered images in the dataset is approximately 1M, which are grouped into 32k continuous sequences (30k train, 944 eval, and 875 test). For training and evaluation we perform dynamic sampling of the original trajectories/panoramas to generate scenes. First, we select a subsequence of the length of 5, and randomly assign it to the input and target locations with repetitions. Next, we select a random reference point at the distance 5-15 of the scene size and sample view directions from the normal distribution with the mean pointing to the reference point and the standard deviation of 0.35 FoV. This strategy is used to make sure that all views cover approximately the same area of the scene but have sufficient diversity. The pitch is selected in the range 0-10 degrees with an additional yaw dependent offset to minimize appearance of the car collecting the data. Finally, we perform Gnomonic projection [20] to generate the corresponding images with FoV of 60 degrees from the panoramic input. The camera parameters are evaluated according to the selected view. At evaluation we draw 48 random scenes from each trajectory. The resulting dataset, MSVS-1M will be released to the public under the CC-BY-SA license. Fig. 3 shows several example scenes sampled from the MSVS-1M dataset using the previously described sampling procedure. The MSVS-1M dataset is designed to provide challenging, real-world cases for benchmarking novel view synthesis methods. For example, despite selecting 5 image acquisition locations and pointing the views to the reference point, some target views may include parts of the scene never provided in the input. Additionally, the dataset contains a large variety of locations, and items in contrast to the MSN dataset which includes only ShapeNet items. Another challenge present in the dataset is poor or no overlap between parts of the scene, especially in case of the reference image, which is illustrated at the right bottom part of Fig. 3. Under conditions of poor overlap, scene understanding by the model is challenging, and overall, we observe lower performance of the ERUPT model on Mapillary data compared to MSN, however we believe the data serves as a helpful step toward view synthesis on large-scale real world data.

1.4. Loss Functions

The loss function used for training ERUPT is described by the following:

$$L_{tot} = L_{img} + w_c L_c + w_t L_t \tag{1}$$

where the first term, L_{img}, is image L_2 loss applied between the model output x^{img} and target y^{img} images as shown in Equation (2). This term is included to minimize the average error between the target and reconstructed image, and maintain overall scene structure.

$$L_{img} = \left\| x^{img} - y^{img} \right\|^2 \tag{2}$$

L_c is the camera auxiliary loss applied to both the estimated input and target camera parameters, given by (3). This loss

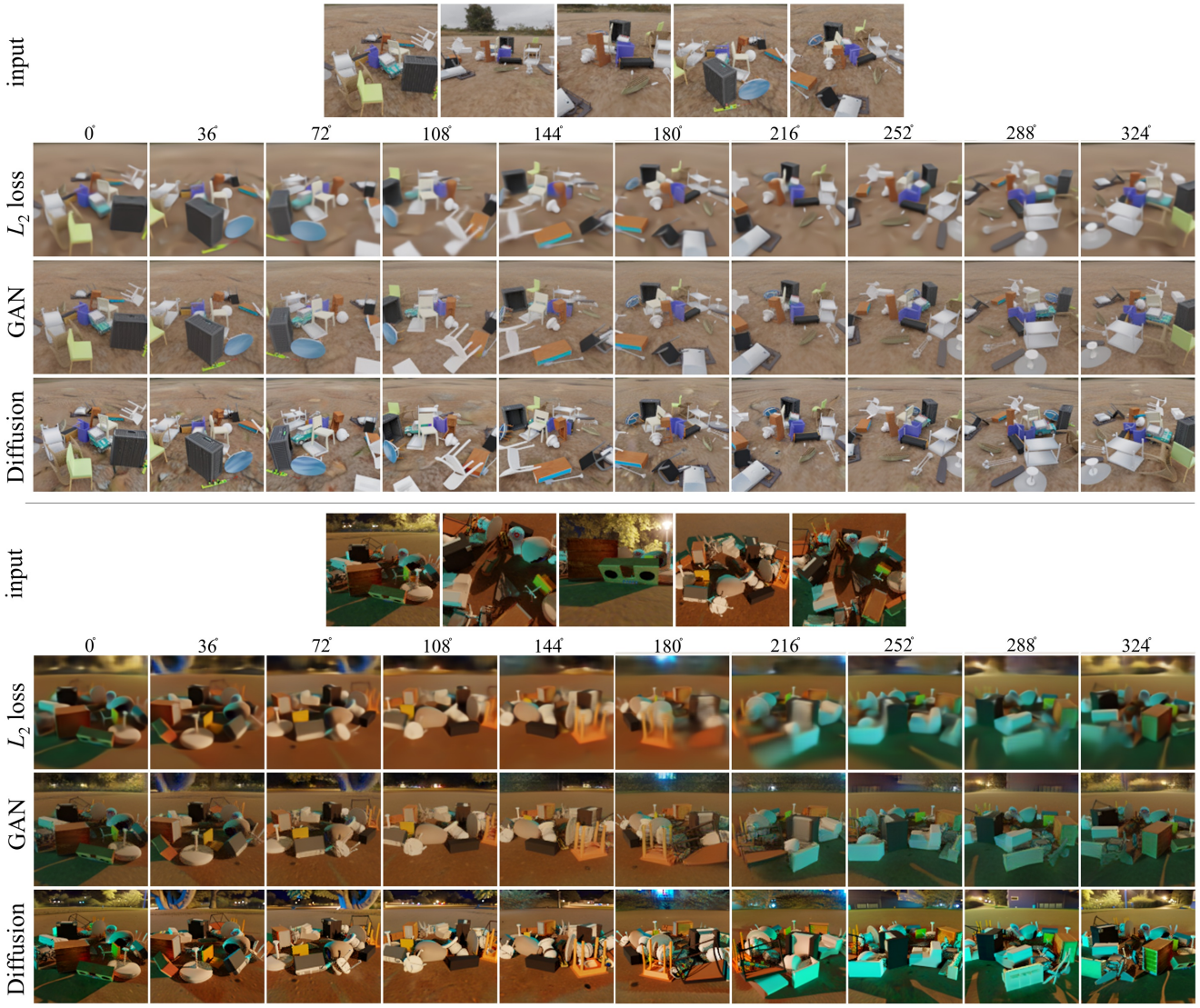

Figure 1. 360-rotation around a scene for cases with all items sufficiently represented in the input images (top) and having missing parts (bottom), which result in blur uncertain output in L_2 setup.

Figure 2. Scene consistency: effect of seed on SD output.

is included for all experiments except the experiment with 5% labeled target images, where this loss is omitted. w_c is a weighting parameter which controls the contribution of the camera auxiliary loss equal to 1/20. x_k^{view},y_k^{view} are the k-th component of the predicted and ground truth camera basis vectors (xyz), respectively, and the corresponding term represents negative cosine loss; while x^{pos}, y^{pos} are predicted and ground truth camera positions.

$$L_c = \frac{1}{3}\Sigma_{k=1}^{3}(1 - \frac{x_k^{view} \cdot y_k^{view}}{\left\|x_k^{view}\right\| \left\|y_k^{view}\right\|}) + \frac{1}{20}\|x^{pos} - y^{pos}\|^2 \tag{3}$$

L_t is the pair-wise contrastive token loss (see Section 3.1 in the main text) computed for N tokens within each image, as shown in Equation (4) where s=20 and m=0.5 are scale and margin, θ_{ij} is the angle between i-th and j-th tokens of the considered image, and L_t is equal to 1/5. This loss is applied to token decoder output and is included to guide learning the semantics of the scene.

Figure 3. Example scenes created using dynamic sampling of MSVS-1M imagery. Five input views and five target views are shown for each scene.

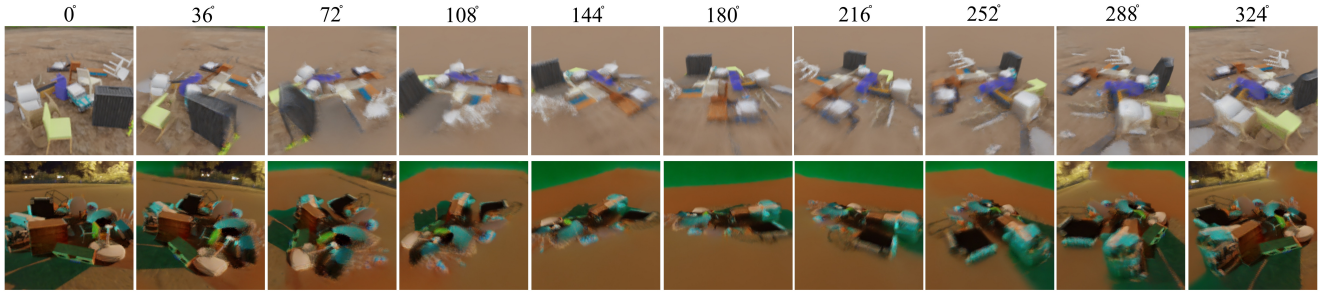

Figure 4. Example of output for scenes from Fig. 1 generated with ZeroNVS. Only reference view is used as the input.

$$L_t = \frac{1}{N}\Sigma_N log(\frac{e^{s\cdot cos(\theta_{ii+m})}}{e^{s\cdot cos(\theta_{ii+m})} + \Sigma_N e^{s\cdot cos(\theta_{ij})}}) \quad (4)$$

In the case of our GAN setup, we follow a standard loss formulation with several terms including L_1, perceptual and adversarial components written as:

$$L_{tot}^{GAN} = L_{img}^1 + w_c L_c + w_t L_t + L_p + L_g \quad (5)$$

where L_{img}^1 is L_1 loss applied between input and target images, L_p is VGG based perception loss, L_g is the GAN loss.

In case of SD finetuning, the ERUPT model is frozen, and we add positional encoding to the output of the token decoder followed by learnable projection to the SD prompt dimension. SD U-net and the projection layers are finetuned with standard epsilon L_2 diffusion objective.

1.5. Comparison to ZeroNVS

A series of recent studies consider the task of 3d scene generation based on a single input view using a combination of diffusion models and NeRF. Fig. 4 provides a qualitative comparison of the ZeroNVS output [Sargent *et al.*, ZeroNVS: Zero-Shot 360-Degree View Synthesis from a Single Image] for scenes considered previously with ERUPT, Fig. 1. Only the reference view is used as input to the model. Even if the model is able to reconstruct a few objects in several views, the overall output quality is noticeably lower compared to ERUPT coupled with finetuned SD, which suggests that the use of multiple input views may be crucial for accurate scene reconstruction. In addition, the ZeroNVS runtime is several hours per scene compared to several seconds to ERUPT if SD rendering is used.