DORSal: Diffusion for Object-centric Representations of Scenes *et al.*

Allan Jabri[†,*] UC Berkeley
Sjoerd van Steenkiste[*] Google Research
Emiel Hoogeboom Google DeepMind
Mehdi S. M. Sajjadi Google DeepMind
Thomas Kipf Google DeepMind

Abstract

Recent progress in 3D scene understanding enables scalable learning of representations across large datasets of diverse scenes. As a consequence, generalization to unseen scenes and objects, rendering novel views from just a single or a handful of input images, and controllable scene generation that supports editing, is now possible. However, training jointly on a large number of scenes typically compromises rendering quality when compared to single-scene optimized models such as NeRFs. In this paper, we leverage recent progress in diffusion models to equip 3D scene representation learning models with the ability to render high-fidelity novel views, while retaining benefits such as object-level scene editing to a large degree. In particular, we propose DORSal, which adapts a video diffusion architecture for 3D scene generation conditioned on object-centric slot-based representations of scenes. On both complex synthetic multi-object scenes and on the real-world large-scale Street View dataset, we show that DORSal enables scalable neural rendering of 3D scenes with object-level editing and improves upon existing approaches.

1 Introduction

Recent works on 3D scene understanding have shown how geometry-free neural networks trained on a large number of scenes can learn scene representations from which novel-views can be synthesized [44, 40]. Unlike Neural Radiance Fields (NeRFs) [25], they are trained to generalize to novel scenes and require only few observations per scene. They also benefit from the ability of learning more *structured* scene representations, e.g. object representations that capture shared statistical structure (e.g. cars) observed throughout many different scenes [48, 55, 38]. However, these models are trained with only a few observations per scene, and without a means to account for the uncertainty about scene content that remains unobserved they typically fall short at synthesizing precise novel views and produce blurry renderings (see Figure 4 for representative examples).

Equally recently, diffusion models [45] have led to breakthrough performance in image synthesis, including super resolution [37], image-to-image translation [35] and in particular text-to-image generation [36]. Part of the appeal of diffusion models lies in their simplicity, scalability, and steerability via conditioning. For example, text-to-image models can be used to edit scenes via prompting because of the compositional scene structure induced by training with language [11]. While diffusion models have recently been applied to novel-view synthesis, scaling to complex visual scenes while maintaining 3d consistency remains a challenge [51].

[†]Work done while interning at Google, [*]equal contribution.
Correspondence: svansteenkiste@google.com, tkipf@google.com

Preprint. Under review.

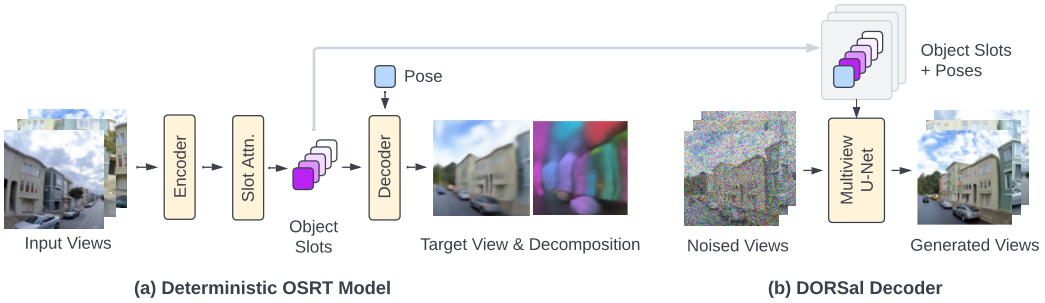


Figure 1: **Model overview.** (a) OSRT is trained to predict novel views through an Encoder-Decoder architecture with an *Object Slot* latent representation of the scene. Since the model is trained with the L2 loss and the task contains significant amounts of ambiguity, the predictions are commonly blurry. (b) After training the OSRT model, we take the Object Slots and combine it with the target Poses to be used as conditioning. Our Multiview U-Net is trained in a diffusion process to denoise novel views while cross-attending into the conditioning features (see Figure 2 for details). This results in sharp renders at test time, which can still be decomposed into the objects in the scene to support edits.

In this work, we combine techniques from both of these subfields to further neural 3D scene rendering. We leverage object-centric scene representations to condition probabilistic diffusion decoders capable of synthesizing novel views while also handling uncertainty about the scene. In particular, we use Object Scene Representation Transformer (OSRT) [38] to compute a set of *Object Slots* for a visual scene from only few observations, and condition a video diffusion architecture [17] with these slots to generate sets of 3D consistent novel views of the same scene. We show that conditioning on *object-level* representations allows for scaling more gracefully to complex scenes, large sets of target views, and allows basic object-level scene editing by removing slots. In summary, our contributions are as follows:

- We introduce *Diffusion for Object-centric Representations of Scenes et al.* (DORSal), an approach to 3D novel-view synthesis combining object representations with diffusion decoders.
- Compared to prior methods from the 3D scene understanding literature [38, 40], DORSal renders novel views that are significantly more precise (e.g. 5x-10x improvement in FID) while staying true to the content of the scene. Compared to prior work on 3D Diffusion Models [51], DORSal scales to more complex scenes, performing significantly better on real-world Street View data.
- Finally, we demonstrate how, by conditioning on a structured, object-based scene representation, DORSal learns to compose scenes out of individual objects, enabling basic object-level scene editing capabilities at inference time.

2 Preliminaries

DORSal is a diffusion generative model conditioned on a simple object-centric scene representation.

Object-centric Scene Representations. Recent breakthroughs in neural rendering have inspired multiple works for 3D-centric object representations, including uORF [55] and ObSuRF [48]. However, these methods do not scale beyond simple datasets due to the high memory and compute requirements of volumetric rendering. More recently, the Object Scene Representation Transformer (OSRT) [38] has been proposed as a powerful method that scales to much more complex datasets with wider camera pose distributions such as MultiShapeNet [40]. Building upon SRT [40], it uses light-field rendering and to obtain speed-ups by a factor of $\mathcal{O}(100)$ at inference time.

An overview of OSRT's model architecture is shown in Figure 1(a). A small set of *input views* is encoded through a CNN followed by a self-attention Transformer [50] (*Encoder*). The resulting set-latent scene representation (SLSR) is fed to Slot Attention [24], which cross-attends from a set of slots into the SLSR. This leads to the Object Slots, an object-centric description of the scene. The number of slots is chosen by the user and sets an upper bound on the number of objects that can be modeled for each individual scene during training.

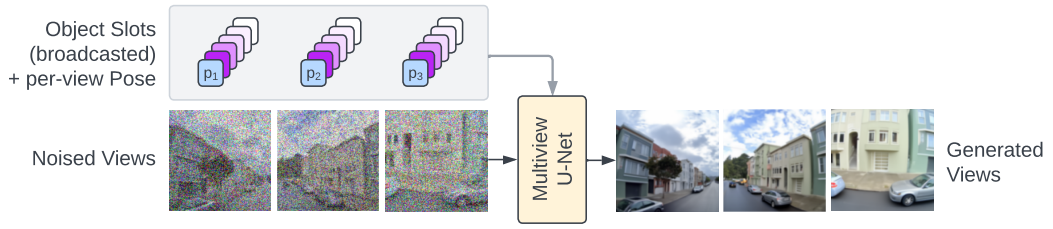


Figure 2: **DORSal slot and pose conditioning.** DORSal is conditioned via cross-attention and FiLM-modulation [28] on a set of Object Slots (shared across views) and a per-view Pose vector.

Once the input views are encoded into the Object Slots, arbitrary novel views can be rendered by passing the target ray origin and direction (the *Pose*) into the *Decoder*. To encourage an object-centric decomposition in the Object Slots, Spatial Broadcast Decoders [52] are commonly used in the literature: Each slot is decoded independently into a pair of RGB and alpha using the same decoder, after which a Softmax over the slots decides on the final output color. Since OSRT is trained end-to-end with the L2 loss, any uncertainty about novel views necessarily leads to blur in the final renders.

Generative Modeling with Conditional DDPMs. Denoising Diffusion Probabilistic Models (DDPMs) learn to generate data $\boldsymbol{x}$ by learning the reverse of a simple destruction process [45]. Such a diffusion process is convenient to express in its marginal form:

$$q(\boldsymbol{z}_t|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{z}_t|\alpha_t\boldsymbol{x}, \sigma_t^2\mathbf{I}), \tag{1}$$

where α_t is a decreasing function and σ_t is an increasing function over diffusion time $t \in [0, 1]$. A neural network is then used to approximate $\boldsymbol{\epsilon}_t$, the reparametrization noise to sample $\boldsymbol{z}_t$:

$$L = \mathbb{E}_{t\sim\mathcal{U}(0,1),\boldsymbol{\epsilon}_t\sim\mathcal{N}(0,\mathbf{I})}\Big[w(t)||\boldsymbol{\epsilon}_t - f(\boldsymbol{z}_t, t)||^2\Big], \tag{2}$$

where f is a neural network and $\boldsymbol{z}_t = \alpha_t\boldsymbol{x} + \sigma_t\boldsymbol{\epsilon}_t$. There exists a particular weighting $w(t)$ for this objective to be a variational negative lowerbound on $\log p(\boldsymbol{x})$, although in practice the constant weighting $w(t) = 1$ has been found to be superior for sample quality [14, 21]. Because diffusion models learn to correlate the pixels in their generations, they are able to generate images with crisp details even if the exact location of such details is not entirely known.

A diffusion model can be made to generate *conditionally* by adding conditioning information $\boldsymbol{s}$ into the neural network function $f(\boldsymbol{z}_t, t, \boldsymbol{s})$, e.g. implemented using a cross-attention in a U-Net [34]. When $\boldsymbol{s}$ sufficiently describes $\boldsymbol{x}$, using a diffusion model does not necessarily yield improvements compared to directly predicting $\boldsymbol{x}$ given $\boldsymbol{s}$. However, if $\boldsymbol{s}$ contains only partial information about $\boldsymbol{x}$ (such as a language description or a high-level scene description), then a conditional diffusion model will be able to utilize that information in addition to correlating pixels. Compared to directly predicting $\boldsymbol{x}$ given $\boldsymbol{s}$ in this scenario, a diffusion model will be able to generate high frequency details even if precise knowledge (such as about location, or shape) is not entirely given by $\boldsymbol{s}$. In contrast, directly predicting $\boldsymbol{x}$ from $\boldsymbol{s}$ (for instance after training with mean-squared error) would result in blurry predictions due to incomplete information.

3 DORSal

DORSal consist of two main components, illustrated in Figure 1. First, we encode a few context views into Object Slots using the encoder of a pre-trained Object Scene Representation Transformer (OSRT) [38]. Second, we train a video diffusion architecture [17] conditioned on these Object Slots to synthesize a set of 3D consistent renderings of novel views of that same scene.

3.1 Decoder Architecture & Conditioning

Architecture details. The DORSal decoder uses a convolutional U-Net architecture as is conventional in the diffusion literature [14]. To attain consistency between L views generated in parallel, following Video Diffusion [17], each frame has feature maps which are enriched with 2d convolutions to process information within each frame and axial (self-)attention to propagate information between

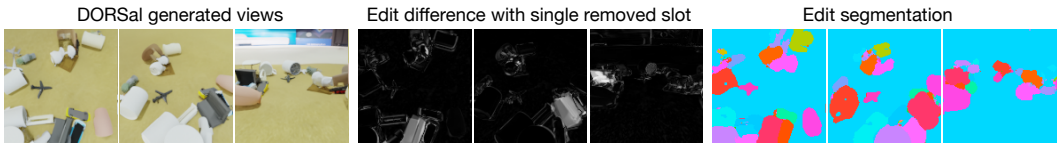


Figure 3: **DORSal scene editing and evaluation.** To obtain instance segmentations of objects in a scene, we perform scene edits by dropping out individual slots, rendering the resulting views, and computing a pixel-wise difference (*middle*) compared to the unedited rendered views (*left*). These differences are smoothed and thresholded to arrive at a segmentation image (*right*).

frames. We refer to this as a Multiview U-Net in our setting as each frame corresponds to a separate view of a scene. This is different from the X-UNet used in 3DiM [51], which relies on cross-attention between conditioning views and generated views, and therefore scales less favourably when increasing the number of output views L. DORSal relies on Object Slots for context about the scene, which avoids the cost of attending directly to large sets of conditioning features that are often redundant.

Conditioning. The generator is conditioned with embeddings of the slots, target pose, and diffusion noise level. To compute these embeddings, given a set of K Object Slots $[\mathbf{s}_1, \ldots, \mathbf{s}_K]$ that describe a single scene, we project the individual Object Slots and broadcast them across views. We append the target camera pose $\mathbf{p}_i$ to the Object Slots for each view $i = 1, \ldots, L$, after applying a learnable linear projection. Thus, each view i is conditioned on the following set of $K + 1$ tokens: $[f(\mathbf{s}_1), \ldots, f(\mathbf{s}_K), g(\mathbf{p}_i)]$, where $f(...)$ and $g(...)$ are learnable linear projections to the same dimensionality D. This process is depicted in Figure 2.

We apply this conditioning in the same way that text is treated in recent work on text-to-image models Saharia et al. [37], i.e. integrated into the U-Net in two ways: 1) we attention-pool [30] conditioning embeddings into a single embedding for modulating U-Net feature maps via FiLM [28], and 2) we use cross-attention [50] to attend on conditioning embeddings (keys) from the feature map (queries).

Independent slot conditioning. Ideally, slot representations that summarize the scene should be conditionally independent given an image $\boldsymbol{z}_t$, $p(\boldsymbol{s}_{1:K}|\boldsymbol{z}_t) = \prod_{k=1,K} p(\boldsymbol{s}_i|\boldsymbol{z}_t)$, i.e. to be able to manipulate the presence of objects independently for editing purposes. In reality, the slot representations may respect this assumption to varying degrees, with an OSRT model trained with instance-level supervision (Sup-OSRT) being more likely to achieve this. However, even if slots would exclusively bind to particular regions of the encoded input views that correspond to individual objects, slots may still share information as the input view encoder has a global receptive field. To mitigate this issue, we consider dropping slots from the conditioning set independently following a Bernoulli rate set as a hyper-parameter λ_{sd}. The model thus sees slot subsets at training time (such that edits are now effectively in-distribution). As we will see, this allows for qualitatively better edits.

3.2 Editing & Sampling

Scene editing. At inference time, we explore a simple form of scene editing: by removing individual slots, we can—if the slot succinctly describes an individual object in the scene—remove that object from the scene. We remove slots by masking out the value of the slot, including any attention weights derived from it. Sampling with this edited conditioning yields K edited scene renderings, where K is the number of object slots in each model. We can then derive the effect of each edit by comparing it to unedited samples generated by keeping all slots for conditioning. To measure success, and to compare between methods, we propose to segment pixels based on whether they were affected by removing a particular slot, and compare to ground-truth instance segments using standard segmentation metrics.

To obtain instance segments from edits with DORSal, we propose the following procedure:

1. **Edit pixel difference:** We take the pixel-wise difference between unedited novel views and their edited counter-parts, averaged across color channels (see Figure 3 middle). This difference is sensitive to object removal if the revealed pixels differ in appearance from the removed object.

2. **Smoothing:** We apply a per-pixel softmax across all K difference images to suppress the contribution of minor side effects of edits (e.g. pixels unrelated to an edited object that slightly change

after an edit) and provide a consistent normalization across each of the K edits. Furthermore, we apply a median filter with a filter size of approx. 5% of the image size (e.g. width).

3. **Assignment:** Finally, we take the per-pixel argmax across K edits to arrive at instance segmentation masks, from which we can compute segmentation metrics for evaluation.

View-consistent sampling Repeatedly generating blocks of L frames is fast, but there is no guarantee on the consistency between the different blocks. This is because sampling from a conditional generative model inherently adds bits of information to the conditioning signal to produce a one-to-many mapping. This is useful in cases of partial observability, such as occlusion, wherein there exist multiple possibilities for unobserved scene content. However, this means that achieving consistency across views involves synchronizing the manner in which bits of information are added, which is challenging as the number of output views grows beyond the amount used during training (eg. $L = 3$, 5 or 10). One approach is to alter the diffusion process to include a selection of already generated previous frames to generate the next set. This strategy is *auto-regressive*, and thus can be slower (only one additional frame is generated each time), and lead to exploding errors for longer auto-regressive chains.

Instead, we leverage the iterative nature of the generative denoising process to create *smooth transitions* as well as *global consistency* between frames. Our technique is inspired by Hoogeboom et al. [18], where high resolution images are generated with overlapping patches by dividing the typical denoising process with T steps is divided into multiple stages. To apply this technique for 3D camera-path rendering of up to 190 views in Section 5.3, we propose to interleave 3 types of frame shuffling for subsequent stages: 1) no shuffle (identity), to allow the model to make blocks of the context length consistent; 2) shift the frames in time by about half of the context length, which puts frames together with new neighbours in their context, allowing the model to create smooth transitions; 3) shuffle all frames with a random permutation, to allow the model to resolve inconsistencies globally.

4 Related Work

Novel View Synthesis (NVS) and 3D Scene Representations. Motivated by NeRF [25], significant advances have recently been achieved in neural rendering [49]. From many observations, NeRF optimizes an MLP through volumetric rendering, thereby allowing high-quality NVS. While several works extend this method to generalizing from few observations per scene [54, 2], they do not provide accessible latent representations. Several *latent* 3D representation methods exist [43, 4, 26], however they do not scale beyond simple synthetic datasets. The recently proposed Scene Representation Transformer (SRT [40]) and extensions (RUST [39]) use large set-latent scene representations to scale to complex real-world datasets with or without pose information. However, SRT often produces blurry images due the L2-loss and high uncertainty in unobserved regions. While approaches like [33] consider generative models for NVS, attaining 3d consistency is challenging with auto-regressive models.

Diffusion. Modern score-based diffusion models [45, 47, 14] have been very successful in multiple domains. They learn to approximate a small step of a denoising process, the reverse of the pre-defined diffusion process. This setup has proven to be very successful and easy to use compared to other generative approaches such as variational autoencoders [21], normalizing flows [32] and adversarial networks [5]. Examples where diffusion models have had success are generation of images [16, 3], audio [22], and video [15]. Moreover, the extent to which they can be steered to be consistent with conditioning signals [13, 27] has allowed for much more controllable image generation. More recently, pose-conditional image-to-image diffusion models have been applied to 3D NVS [51, 23, 10], focusing mainly on 3D synthesis of individual objects as opposed to complex visual scenes. DORSal leverages video diffusion models [17] and object-slot conditioning to synthesize novel views that are more consistent, especially in real-world settings, and support object-level edits.

Object representations. Object-centric methods for representation learning [8] aim at structuring the representation of, e.g., an image, a video, or of a 3D scene, in terms of its object content. This provides a direct interface for scene editing as well as object-centric downstream tasks, such as detection, tracking or instance segmentation. Most successful methods adopt a slot-based approach [7, 24, 42, 41], i.e. they learn a set of object slots to describe the content of a scene. In the context of 3D scene representation learning, ObSuRF [48], uORF [55], and OSRT [38] are recent examples of combining slot-based models with methods for 3D representation learning. We base DORSal on object slots obtained from OSRT as this method scales to large and diverse datasets. As explored in

Figure 4: **Novel View Synthesis**. Comparison of DORSal with the following baselines: 3DiM [51], SRT [40], and OSRT [38] on the MultiShapeNet (only 2/5 views shown) and Street View datasets.

Slot-TTA [29], OSRT can further benefit from instance mask supervision, if available. In concurrent work, slot-based methods have also been explored in combination with diffusion-based decoders: LSD [19] and SlotDiffusion [53] combine Slot Attention with a diffusion decoder in latent space for image and (for the latter) video object segmentation. Neither approach, however, considers 3D scenes or NVS, but solely focus on auto-encoding objectives.

5 Experiments

We evaluate DORSal on challenging synthetic and real-world scenes in three settings: 1) we compare the ability to synthesize novel views of a scene with related approaches, 2) we analyze the capability for simple scene edits (object removal) by dropping individual object slots, and 3) we investigate the ability of DORSal to render smooth, view-consistent camera paths. Complete experimental details are available in Appendix C and additional results in Appendix D.

Datasets. *MultiShapeNet (MSN)* [40] consists of scene with 16–32 ShapeNet [1] objects each. The complex object arrangement, realistic rendering [9], HDR backgrounds, random camera poses, and the use of fully novel objects in the test set make this dataset highly challenging. We use the version from Sajjadi et al. [38] (*MSN-Hard*). The *Street View (SV)* dataset contains photographs of real-world city scenes. The highly inconsistent camera pose distribution, moving objects, and changes in exposure and white balance make this dataset a good test bed for generative modeling. Street View imagery and permission for publication have been obtained from the authors [6].

Baselines. For comparison, we focus on SRT and OSRT from the 3D scene understanding literature [38, 40], and 3DiM from the diffusion literature [51]. Because OSRT (Figure 1(a)) and DORSal (Figure 1(b)) leverage the same object-centric scene representation, we can compare them in terms of the quality of generated novel-views as well as the ability to perform object-level scene edits. SRT, which was previously applied to Street View and mainly differs to OSRT in terms of its architecture, does not include Object Slots as a bottleneck. We use Sup-OSRT to compute object-slots for DORSal on MultiShapeNet and plain OSRT on Street View (where ground-truth masks are unavailable).

3DiM is a pose-conditional image-to-image diffusion model for generating novel views of the same scene [51]. During training, 3DiM takes as input a pair of views of a static scene where one of the views is corrupted with noise for training purposes. During inference, 3DiM makes use of *stochastic conditioning* to generate 3D-consistent views of a scene: a new view for a given target camera pose is generated by conditioning on a randomly selected view from a conditioning set at each denoising step. Each time a new view is generated, it is added to the conditioning set.

Table 1: **Novel-view synthesis**. Evaluation on 1 k test scenes from MultiShapeNet and Street View.

	MultiShapeNet				Street View			
Model	**PSNR↑**	**SSIM↑**	**LPIPS↓**	**FID↓**	**PSNR↑**	**SSIM↑**	**LPIPS↓**	**FID↓**
SRT	25.93	0.813	0.237	67.29	23.60	0.739	0.282	87.91
OSRT	23.35	0.719	0.330	100.7	21.19	0.614	0.410	165.1
Sup-OSRT	22.64	0.680	0.358	112.1	—	—	—	—
3DiM	18.20	0.559	0.287	10.94	12.68	0.283	0.477	15.58
DORSal	18.76	0.557	0.266	11.01	16.05	0.421	0.361	16.24

5.1 Novel-view Synthesis

Set-up. We separately train DORSal, OSRT, SRT, and 3DiM on MultiShapeNet and Street View, where DORSal and (Sup-)OSRT leverage the same set of Object Slots. We quantitatively evaluate performance at novel-view synthesis on a test set of 1000 scenes. We measure PSNR and SSIM, which capture how well each novel view matches the corresponding ground truth, though are easily exploited by blurry predictions. To address this we also measure FID [12], which compares generated novel views to ground-truth at a distributional level, and LPIPS (VGG) [56], which measures frame-wise similarities using deep feature embeddings.

Results. Quantitative results can be seen in Table 1 and qualitative results in Figure 4. On MultiShapeNet and Street View it can be seen how DORSal obtains slightly lower PSNR and SSIM compared to SRT and (Sup-)OSRT, but greatly outperforms these methods in terms of FID, as expected. This effect can easily be observed qualitatively in Figure 4, where SRT and OSRT render novel views that are blurry (because they average out uncertainty about the scene), while DORSal synthesizes novel-views much more precisely by 'imagining' some of the details, while staying close to the actual content in the scene. Compared to 3DiM, which also leverages a diffusion probabilistic model, it can be seen how DORSal yields a more favorable trade-off between FID and PSNR. Especially on Street View, where there exist large gaps between different views, 3DiM struggles as it only receives a single conditioning view during training, and primarily generates variations on its input view. We provide an additional comparison to 3DiM having access to additional GT input views at inference time, and further a comparison between DDIM and DDPM for DORSal in Appendix D.

5.2 Evaluation of Object-level Edits

Setup. We evaluate the scene editing capabilities of DORSal on both MultiShapeNet and Street View and compare to (Sup-)OSRT. To remove objects from the scene and compute scene edit segmentation masks we follow the protocol described in Section 3.2. We compare the edit segmentation masks obtained in this way to the ground-truth instance segmentation mask for these scenes using ARI [31] and mIoU, which are standard metrics from the segmentation literature. As is common practice, we compute these metrics solely for foreground objects (indicated as FG-). Because ground-truth instance segmentations are unavailable for Street View we only report qualitative results.

Results. Quantitative results on MultiShapeNet can be seen in Table 2 and qualitative results in Figure 5. Compared to OSRT, it can be observed how DORSal is similarly capable at object-level scene editing and attains a high degree of compositionality with regards to the Object Slots it is conditioned on. Interestingly, we find that a smaller version of DORSal training on lower-resolution views achieves significantly better scene editing performance, closely approaching the performance of the supervised OSRT baseline (which here acts as an upper limit). This suggests a cascaded model training strategy as an optimal strategy for object-level scene editing for future work, where first a 64x64 model is trained, followed by one or

Table 2: **Scene editing**. Evaluation on 1 k test scenes of MultiShapeNet (FG-mIoU in %).

Model	**FG-mIoU**	**FG-ARI**
OSRT [38]	43.1	79.6
Sup-OSRT [29]	50.0	75.5
DORSal (64x64)	45.8	70.0
DORSal (128x128)	35.0	64.6

Figure 5: **MultiShapeNet scene edits.** We remove one slot at a time in the conditioning of DORSal and render the resulting scene while keeping the initial image noise fixed. In the leftmost panel, the slot corresponding to the background is removed while all objects are present. The other panels show examples of deleted objects (highlighted in red circles) when their corresponding slot is removed.

Figure 6: **Street View scene edits.** Removing one slot at a time, we show the most compelling examples where objects are erased from the scene. Notably, the encircled tree in the second row is generated upon *removal* of a slot to fill up the now-unobserved facade previously explained by the removed slot. The ground-truth scene does not contain a tree in this position.

multiple (conditional) upsampling models to increase the visual quality while retaining the structure of the scene content [37, 17].

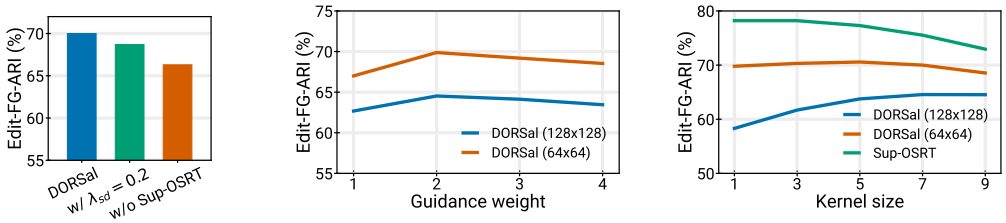
On the real-world Street View dataset, the notion of an object is much more ambiguous and, unlike for MultiShapeNet, the Object Slots provided by the OSRT encoder capture individual objects less frequently. Nonetheless, we qualitatively observe how removal of individual Object Slots in DORSal can often still result in meaningful scene edits. We show a selection of successful scene edits in Figure 6, where dropping a specific Object Slot results in the removal of, for example, a car, a street sign, a trash can, or in the alteration of a building. Not all edits, however, are meaningful and many slots have little to no effect when removed, likely because the OSRT base model often assigns multiple slots to a single object such as a car. This qualitative result, however, remains quite remarkable as the model received no instance supervision whatsoever. We provide exhaustive editing examples (incl. failure cases) in Appendix D.

5.3 Camera-path Rendering

Setup. We qualitatively compare two different training strategies: the first is our default setup on MultiShapeNet where we train on randomly sampled views of the scene. Further, we generate a dataset which has a mix of both nearby views (w.r.t. previously generated views) and uniformly sampled views (at random) from the full view distribution. At inference time, we generate a full circular camera path for each scene using our sampling strategy described in Section 3.2.

Results. We show qualitative results in Figure 7 and in video format in the supplementary material. We find that DORSal is able to render certain objects which are well-represented in the scene representation (e.g. clearly visible in the input views) consistent and smoothly across a camera path, but several regions and objects "flicker" between views as the model fills in slightly different details depending on the view point to account for missing information. We find that this can be largely resolved by training DORSal on the mixed-views dataset (both nearby and random views) as described above, which results in qualitatively smooth videos. This is also reflected in our quantitative results (computed on 40 held-out scenes having 190 target views each) using PSNR as an approximate measure of scene consistency, where we obtain 16.50db PSNR for DORSal, 17.47db for 3DiM and 18.06db for DORSal trained on mixed views.

Figure 7: **Camera path rendering.** *Top*: Example of a circular camera path rendered for DORSal (64x64) trained on MultiShapeNet. While the rendered views are mostly consistent, there can be small inconsistencies in regions of high uncertainty which result in flickering artifacts (see object highlighted in red circle). *Bottom*: When trained on a dataset with a mix of close-by and fully-random camera views, DORSal achieves improved consistency resulting in qualitatively smooth videos.



(a) Slot dropout & base model. (b) Guidance weight. (c) Median filter kernel size.

Figure 8: **Hyperparameter choices and ablations**. In (a), we compare DORSal (64x64) without slot dropout ($\lambda_{sd} = 0$) with two variants, $\lambda_{sd} = 0.2$ and using an unsupervised OSRT model (w/o Sup-OSRT) as base. In (b), we analyse the effect of the guidance weight parameter during inference, and in (c) we show the effect of kernel size on the median filter used during scene edit evaluation.

5.4 Ablations

We investigate the effect of 1) slot dropout, 2) instance segmentation supervision in the base OSRT model (for MultiShapeNet), 3) the guidance weight during inference, and 4) the median filter kernel size for scene edit evaluation. Our results are summarized in Figure 8.

We find that adding slot dropout can have a negative effect on scene editing metrics in MultiShapeNet for which we use Sup-OSRT as the base model (Figure 8a). This is interesting, since for Street View, where supervision is not available, we generally report results using a model with $\lambda_{sd} = 0.2$, as the model without slot dropout did not produce meaningful scene edits. Removing instance supervision in in MultiShapeNet in the OSRT base model expectedly reduces scene editing performance (Figure 8a). Further, we find that choosing a guidance weight larger than 1 generally as a positive effect on prediction quality, with an optimal value of 2 (Figure 8b).

An important hyperparameter for scene editing evaluation is the median filter kernel size, which sets an upper bound on achievable segmentation performance (as fine-grained details are lost), yet is important for removing sensitivity to high-frequency details which can often vary between multiple samples in a generative model. We find that DORSal at 128x128 resolution benefits from smoothing up to a kernel size of 7 (our chosen default), which slightly lowers the achievable segmentation score of the base model (Sup-OSRT), but removes most noise artifacts in our edit evaluation protocol (Figure 8c).

6 Conclusion

We have introduced DORSal, a generative model capable of rendering precise novel views of diverse 3D scenes. By conditioning on an object-centric scene representation, DORSal further supports scene editing: the presence of an object can be controlled by its respective object slot in the scene representation. DORSal adapts an existing text-to-video generative model architecture [17] to controllable 3D scene generation by conditioning on camera poses and object-centric scene representations, and by training on large-scale 3D scene datasets. As we base our model on a state-of-the-art text-to-video model, this likely enables the transfer of future improvements in this model class to the task of compositional 3D scene generation, and opens the door for joint training on large-scale video and 3D scene data.

Summary of Limitations. While DORSal makes significant progress, there are several limitations and open problems worth highlighting, relating to 1) lack of end-to-end training, 2) worse editing performance and consistency for high-resolution training, 3) configuration of the MultiView U-Net architecture for 3D, and 4) non-local editing effects. We discuss these in detail in Appendix B.

Acknowledgments

We would like to thank Alexey Dosovitskiy for general advice and detailed feedback on an early version of this paper. We are grateful to Daniel Watson for making the 3DiM codebase readily available for comparison, and help with debugging and onboarding new datasets.

References

[1] Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al. ShapeNet: An Information-Rich 3D Model Repository. *arXiv preprint arXiv:1512.03012*, 2015.

[2] Chen, A. and Xu, Z. MVSNeRF: Fast Generalizable Radiance Field Reconstruction From Multi-View Stereo. In *ICCV*, 2021.

[3] Dhariwal, P. and Nichol, A. Q. Diffusion models beat gans on image synthesis. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS*, 2021.

[4] Eslami, S. A., Rezende, D. J., Besse, F., Viola, F., Morcos, A. S., Garnelo, M., Ruderman, A., Rusu, A. A., Danihelka, I., Gregor, K., et al. Neural scene representation and rendering. *Science*, 2018.

[5] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems*, pp. 2672–2680, 2014.

[6] Google. Street view, 2007. URL www.google.com/streetview/.

[7] Greff, K., Kaufman, R. L., Kabra, R., Watters, N., Burgess, C., Zoran, D., Matthey, L., Botvinick, M., and Lerchner, A. Multi-object representation learning with iterative variational inference. In *ICML*, 2019.

[8] Greff, K., Van Steenkiste, S., and Schmidhuber, J. On the binding problem in artificial neural networks. *arXiv preprint arXiv:2012.05208*, 2020.

[9] Greff, K., Belletti, F., Beyer, L., Doersch, C., Du, Y., Duckworth, D., Fleet, D. J., Gnanapragasam, D., Golemo, F., Herrmann, C., et al. Kubric: A Scalable Dataset Generator. In *CVPR*, 2022.

[10] Gu, J., Trevithick, A., Lin, K.-E., Susskind, J., Theobalt, C., Liu, L., and Ramamoorthi, R. Nerfdiff: Single-image view synthesis with nerf-guided distillation from 3d-aware diffusion. *arXiv preprint arXiv:2302.10109*, 2023.

[11] Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., and Cohen-or, D. Prompt-to-prompt image editing with cross-attention control. In *ICLR*, 2023.

[12] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 30, 2017.

[13] Ho, J. and Salimans, T. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[14] Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS*, 2020.

[15] Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A. A., Kingma, D. P., Poole, B., Norouzi, M., Fleet, D. J., and Salimans, T. Imagen video: High definition video generation with diffusion models. *CoRR*, abs/2210.02303, 2022.

[16] Ho, J., Saharia, C., Chan, W., Fleet, D. J., Norouzi, M., and Salimans, T. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 23:47:1–47:33, 2022.

[17] Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., and Fleet, D. J. Video diffusion models, 2022.

[18] Hoogeboom, E., Agustsson, E., Mentzer, F., Versari, L., Toderici, G., and Theis, L. High-fidelity image compression with score-based generative models. *arXiv preprint arXiv:2305.18231*, 2023.

[19] Jiang, J., Deng, F., Singh, G., and Ahn, S. Object-centric slot diffusion. *arXiv preprint arXiv:2303.10834*, 2023.

[20] Jouppi, N. P., Kurian, G., Li, S., Ma, P., Nagarajan, R., Nai, L., Patil, N., Subramanian, S., Swing, A., Towles, B., et al. Tpu v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings. *arXiv preprint arXiv:2304.01433*, 2023.

[21] Kingma, D. P., Salimans, T., Poole, B., and Ho, J. Variational diffusion models, 2023.

[22] Kong, Z., Ping, W., Huang, J., Zhao, K., and Catanzaro, B. Diffwave: A versatile diffusion model for audio synthesis. In *9th International Conference on Learning Representations, ICLR*. OpenReview.net, 2021.

[23] Liu, R., Wu, R., Van Hoorick, B., Tokmakov, P., Zakharov, S., and Vondrick, C. Zero-1-to-3: Zero-shot one image to 3d object. *arXiv preprint arXiv:2303.11328*, 2023.

[24] Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., and Kipf, T. Object-centric learning with slot attention. In *NeurIPS*, 2020.

[25] Mildenhall, B., Srinivasan, P., Tancik, M., Barron, J., Ramamoorthi, R., and Ng, R. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*, 2020.

[26] Moreno, P., Kosiorek, A. R., Strathmann, H., Zoran, D., Schneider, R. G., Winckler, B., Markeeva, L., Weber, T., and Rezende, D. J. Laser: Latent set representations for 3d generative modeling. *arXiv preprint arXiv:2301.05747*, 2023.

[27] Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. *CoRR*, abs/2112.10741, 2021. URL https://arxiv.org/abs/2112.10741.

[28] Perez, E., Strub, F., De Vries, H., Dumoulin, V., and Courville, A. Film: Visual reasoning with a general conditioning layer. In *AAAI*, volume 32, 2018.

[29] Prabhudesai, M., Goyal, A., Paul, S., van Steenkiste, S., Sajjadi, M. S., Aggarwal, G., Kipf, T., Pathak, D., and Fragkiadaki, K. Test-time adaptation with slot-centric models. In *ICML*, 2023.

[30] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021.

[31] Rand, W. M. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 1971.

[32] Rezende, D. J. and Mohamed, S. Variational inference with normalizing flows. In Bach, F. R. and Blei, D. M. (eds.), *Proceedings of the 32nd International Conference on Machine Learning, ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 1530–1538. JMLR.org, 2015.

[33] Rombach, R., Esser, P., and Ommer, B. Geometry-free view synthesis: Transformers and no 3d priors. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14336–14346, 2021.

[34] Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241. Springer, 2015.

[35] Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., and Norouzi, M. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pp. 1–10, 2022.

[36] Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[37] Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D. J., and Norouzi, M. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[38] Sajjadi, M. S. M., Duckworth, D., Mahendran, A., van Steenkiste, S., Pavetic, F., Lucic, M., Guibas, L. J., Greff, K., and Kipf, T. Object Scene Representation Transformer. In *NeurIPS*, 2022.

[39] Sajjadi, M. S. M., Mahendran, A., Kipf, T., Pot, E., Duckworth, D., Lučić, M., and Greff, K. RUST: Latent Neural Scene Representations from Unposed Imagery. *CoRR*, abs/2211.14306, 2022.

[40] Sajjadi, M. S. M., Meyer, H., Pot, E., Bergmann, U., Greff, K., Radwan, N., Vora, S., Lucic, M., Duckworth, D., Dosovitskiy, A., et al. Scene Representation Transformer: Geometry-Free Novel View Synthesis Through Set-Latent Scene Representations. In *CVPR*, 2022.

[41] Seitzer, M., Horn, M., Zadaianchuk, A., Zietlow, D., Xiao, T., Simon-Gabriel, C.-J., He, T., Zhang, Z., Schölkopf, B., Brox, T., et al. Bridging the gap to real-world object-centric learning. In *ICLR*, 2023.

[42] Singh, G., Deng, F., and Ahn, S. Illiterate dall-e learns to compose. In *ICLR*, 2022.

[43] Sitzmann, V., Zollhöfer, M., and Wetzstein, G. Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations. In *NeurIPS*, 2019.

[44] Sitzmann, V., Rezchikov, S., Freeman, W. T., Tenenbaum, J. B., and Durand, F. Light field networks: Neural scene representations with single-evaluation rendering. In *NeurIPS*, 2021.

[45] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

[46] Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[47] Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS*, 2019.

[48] Stelzner, K., Kersting, K., and Kosiorek, A. R. Decomposing 3d scenes into objects via unsupervised volume segmentation. *arXiv preprint arXiv:2104.01148*, 2021.

[49] Tewari, A., Thies, J., Mildenhall, B., Srinivasan, P., Tretschk, E., Wang, Y., Lassner, C., Sitzmann, V., Martin-Brualla, R., Lombardi, S., et al. Advances in neural rendering. *Computer Graphics Forum*, 41(2), 2022.

[50] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *NeurIPS*, 2017.

[51] Watson, D., Chan, W., Brualla, R. M., Ho, J., Tagliasacchi, A., and Norouzi, M. Novel view synthesis with diffusion models. In *ICLR*, 2023.

[52] Watters, N., Matthey, L., Burgess, C. P., and Lerchner, A. Spatial broadcast decoder: A simple architecture for learning disentangled representations in vaes. *arXiv preprint arXiv:1901.07017*, 2019.

[53] Wu, Z., Hu, J., Lu, W., Gilitschenski, I., and Garg, A. Slotdiffusion: Unsupervised object-centric learning with diffusion models. *ICLR NeSy-GeMs workshop*, 2023.

[54] Yu, A., Ye, V., Tancik, M., and Kanazawa, A. pixelNeRF: Neural Radiance Fields from One or Few Images. In *CVPR*, 2021.

[55] Yu, H.-X., Guibas, L. J., and Wu, J. Unsupervised discovery of object radiance fields. In *ICLR*, 2022.

[56] Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pp. 586–595, 2018.

A Broader Impacts

DORSal enables precise 3D rendering of novel views conditioned on Object Slots, as well as basic object-level editing. Though we present initial results on Street View, the practical usefulness of DORSal is still limited and thus we foresee no immediate impact on society more broadly. In the longer term, we expect that slot conditioning may facilitate greater interpretability and controllability of diffusion models. However, though we do not rely on web-scraped image-text pairs for conditioning, our approach remains susceptible to dataset selection bias (and related biases). Better understanding the extent to which these biases affect model performance (and interpretability) will be important for mitigating future negative societal impacts that could arise from this line of work.

B Limitations

While DORSal makes significant progress on the challenging problems of novel view synthesis and scene editing, there are several limitations and open problems worth highlighting. As we follow the design of Video Diffusion Models [17] for simplicity, DORSal is not end-to-end trained and is ultimately limited by the quality of the scene representation (Object Slots) provided by the separately trained upstream model (OSRT). End-to-end training comes with additional challenges (e.g. higher memory requirements), but is worth exploring in future work.

As highlighted in our experiments, training at 128x128 resolution with our model design results in decreased editing performance compared to a 64x64 model. We also observed qualitatively worse cross-view consistency in the higher-resolution model. To overcome this limitation, one would likely have to scale the model further in terms of size (at the expense of increased compute and memory requirements) or train a cascade of models to initially predict at 64x64 resolution, followed by one or more conditional upsampling stages, as done in Video Diffusion Models [17].

As the U-Net architecture of DORSal is based on Video Diffusion Models [17], it can be sensitive to ordering of frames in the dataset. While frames in MultiShapeNet are generated from random view points, frames in Street View are ordered by time. DORSal is able to capture this information, which—in turn—makes rendering views from arbitrarily chosen camera paths at test time challenging, as the model has learned a prior for the movement of the camera in the dataset.

For scene editing, we find that removing individual object slots can have non-local side effects, e.g. another object or the background changing its appearance, in some cases. Furthermore, edits of individual are typically not perfect, even when trained with a supervised OSRT base model: objects are sometimes only partially removed, or removal of a slot might have no effect at all.

C Experimental Details

C.1 Evaluation

Novel View Synthesis. We follow the experimentation protocol outlined in Sajjadi et al. [38, 40] and evaluate DORSal and baselines using 5 and 3 novel target views for MultiShapeNet and Street View respectively. Similarly, the OSRT base model, is trained with 5 input views on these datasets. To accommodate the U-Net architecture used in DORSal and 3DiM, we crop Street View frames to 128x128 resolution.

Evaluation of Object-level Edits. We use a median kernel size of 7 for all edit evaluations (incl. the baselines). We evaluate models on the first 1k scenes of the MultiShapeNet dataset. For DORSal, we use an identical initial noise variable (image) for each edit to ensure consistency.

Camera-path Rendering. For camera-path rendering of many views (beyond what DORSal was trained with) we deploy the sampling procedure outlined in Section 5.3. Camera trajectories follow a circular, center-facing path starting from the first input view. Further, we generate a dataset which has a mix of both nearby views (w.r.t. previously generated views) and uniformly sampled views (at random) from the full view distribution as in MultiShapeNet. Here we train DORSal using 10 input views and 10 target views (down-sampled to 64x64 resolution) to keep a similar amount of diversity

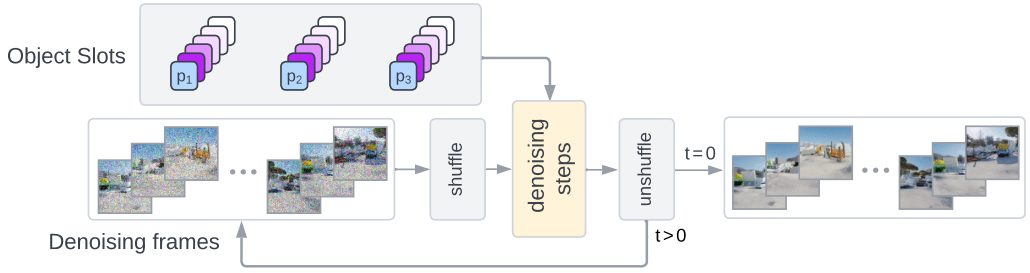


Figure 9: **View consistent rendering for a large number of frames.** Frames are denoised in blocks of length L (e.g. $L = 3, 5$ or 10). To ensure consistency between blocks, the shuffle component acts in one of the following three ways: 1) identity (do nothing) 2) shift the frames by about half the context length, for smoothness between neighbouring blocks, and 3) a random permutation for global consistency.

Table 3: DORSal U-Net architecture details.

Model	Channels per level	Blocks per level	Attention resolution	Patching
U-Net 64	192, 384, 576	3	16×16	No
U-Net 128	256, 512, 1024	3	16×16	2×2

when sampling far-away as well as close-by views. 3DiM is trained similarly as for the novel-view synthesis experiments.

C.2 Model Details

C.2.1 DORSal

Conditioning. We obtain Object Slots from a separately trained OSRT model. In the case of MultiShapeNet, we train OSRT with instance segmentation mask supervision following the approach by Prabhudesai et al. [29]: we take the alpha masks produced by the broadcast decoder to obtain soft segmentation masks, which we match using Hungarian matching with ground-truth instance masks (under an L2 objective) and finally train the model using a cross-entropy loss using the alpha mask logits on the matched target masks. For Street View, we use the default unsupervised OSRT model with a broadcast decoder, as instance masks are not available. All OSRT models use 32 Object Slots.

Network Architecture. For DORSal we follow the architecture of Ho et al. [15], which is a U-Net that has axial attention over time. Differences are that DORSal does not require text conditioning cross attention layers, and it uses slot embeddings augmented with camera poses. The architecture sizes are small compared to Ho et al. [15] as can be seen in Table 3. The U-Net on resolutions of 128×128 uses patching to avoid memory expensive feature maps. For the 16×16 resolution, the ResBlocks use per-view self-attention and between-views cross-attention.

Training. We adopt a similar training set-up to Ho et al. [17], using a cosine-shaped noise schedule, a learning rate with a peak value of 0.00003 using linear warm-up for 5000 steps, optimization using Adam with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and EMA decay for the model parameters. We train with a global batch size of 8 and classifier-free guidance with a conditioning dropout probability of 0.1 (and an inference guidance weight of 2). We report results after training for 1 000 000 steps. For MultiShapeNet, we use Object Slots from Sup-OSRT (i.e. supervised) and do not use slot dropout during training. For Street View, we use Object Slots from OSRT (i.e. unsupervised) and use a slot dropout probability of 0.2, which we found to improve editing quality on this dataset (compared to no slot dropout).

Camera-Path Sampling. We leverage the iterative nature of the generative denoising process to create *smooth transitions* as well as *global consistency* between frames. Our technique (Figure 9) is inspired by Hoogeboom et al. [18], where high resolution images are generated with overlapping

patches by dividing the typical denoising process with T steps is divided into multiple stages. Instead of generating a patch entirely for all denoising timesteps from one to zero, first a noisy version of each patch is generated by generating from one to $3/4$, in the case of 4 stages. Only after *all* noisy patches at time $3/4$ are generated, do they proceed to the next stage from $3/4$ to $2/4$.

To apply this technique for 3D camera-path rendering of up to 190 views in Section 5.3, we propose to interleave 3 types of frame shuffling for subsequent stages: 1) no shuffle (identity), to allow the model to make blocks of the context length consistent; 2) shift the frames in time by about half of the context length, which puts frames together with new neighbours in their context, allowing the model to create smooth transitions; 3) shuffle all frames with a random permutation, to allow the model to resolve inconsistencies globally.

C.2.2 3D Diffusion Model (3DiM)

We compare to 3DiM, which is a pose-conditional image-to-image diffusion model for generating novel views of the same scene [51]. During training, 3DiM takes as input a pair of views of a static scene (including their poses), where one of the views (designated as the "target view") is corrupted with noise. The training objective is to predict the Gaussian noise that was used to corrupt the target view. During inference, 3DiM makes use of *stochastic conditioning* to generate 3D-consistent views of a scene. In particular, given a small set of k conditioning views and their camera poses (typically $k = 1$), a new view for a given target camera pose is generated by conditioning on a randomly selected view from the conditioning set at each denoising step. Each time a new view is generated, it is added to the conditioning set.

Network Architecture. In our experiments we use the default $\sim$471M parameter version of their X-UNet, which amounts to a base channel dimension of *ch* $= 256$, four stages for down- and up-sampling using *ch_mult* $= (1, 2, 2, 4)$, and 3 ResBlocks per stage using per-view self-attention and between-views cross-attention at resolutions $(8, 16, 32)$. Note how this configuration uses many more parameters per view, compared to DORSal. In line with DORSal, we use absolute positional encodings for the camera rays in our experiments on MultiShapeNet and StreetView (scaling down the ray origins by a factor of 30).

Training. We adopt the same training set-up as in the 3DiM paper, which consist of a cosine-shaped noise schedule, a learning rate with peak value of 0.0001 using linear warm-up for 10M samples, optimization using Adam with $\beta_1 = 0.9$ and $\beta_2 = 0.99$, and EMA decay for the model parameters. We train with a global batch size of 128 and classifier-free guidance 10% with a weight of 3, as was done for the experiment on SRN cars in their paper. We report results after training for 320 000 steps.

Sampling. We generate samples in the same way as in the 3DiM paper, using 256 DDPM denoising steps and clip to [-1, 1] after each step.

For additional details, including code, we refer to Sections 6 & 7 in Watson et al. [51].

C.2.3 SRT & OSRT

SRT was originally proposed by Sajjadi et al. [40] with Set-Latent Scene Representations (SLSR) and subsequently adapted to Object Slots for OSRT [38]. At the same time, a few tweaks were made to the model, e.g. by using a smaller patch size and a larger render MLP [38]. For all our experiments (SRT and OSRT), we use the improved architecture from the OSRT paper, see Appendix A.4 in [38] for the model and training details. Following Sajjadi et al. [40], we train all models for $\sim$4M steps on both datasets.

C.3 Compute and Data Licenses

We train DORSal on 8 TPU v4 [20] chips using a batch size of 8 for approx. one week to reach 1M steps. The MultiShapeNet dataset was introduced by Sajjadi et al. [40] and was generated using Kubric [9], which is available under an Apache 2.0 license. Street View imagery and permission for publication have been obtained from the authors [6].

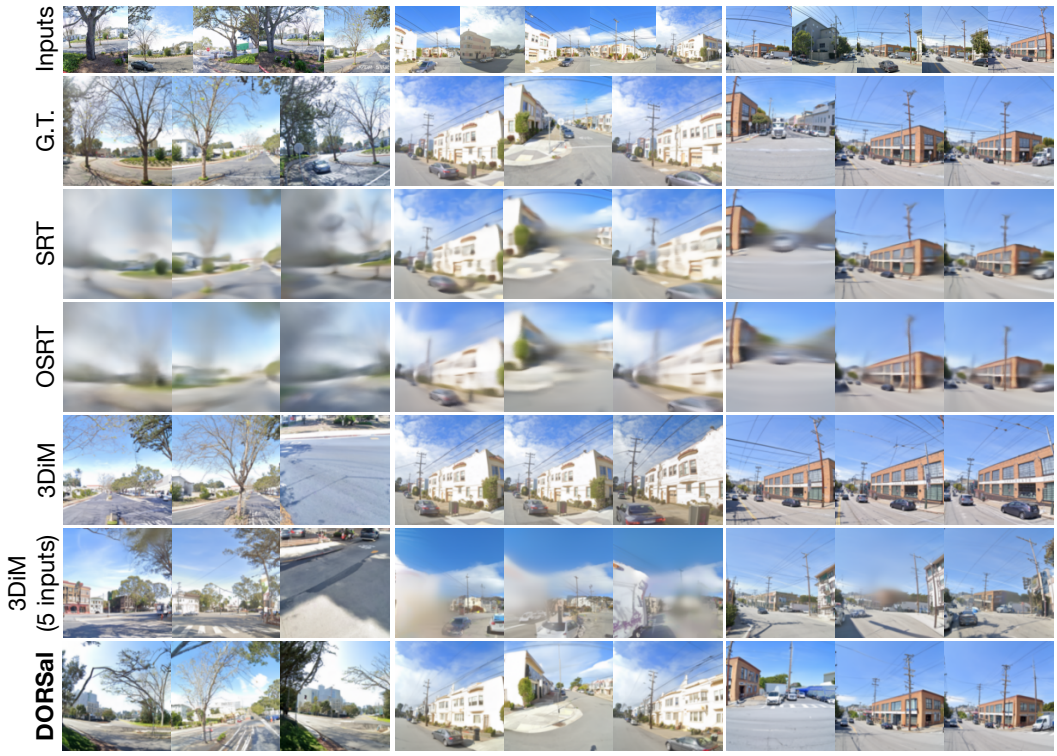


Figure 10: **Novel View Synthesis (Street View)**. Qualitative results incl. input views (top tow) for additional Street View scenes. We further include a version of 3DiM that is conditioned on 5 ground-truth input views.

D Additional Results

D.1 Qualitative Results

Novel View Synthesis We provide additional qualitative results for novel view synthesis in Figure 10 (Street View) and in Figure 11 (MultiShapeNet). For Street View, it is evident that even when modifying 3DiM to use 5 ground-truth input views during inference, it is unable to synthesize accurate views from novel directions, while DORSal renders realistic views that adhere to the content of the scene.

Scene Editing In Figure 12 we provide exhaustive scene editing results for several Street View scenes: each image shows one generation of DORSal with exactly one slot removed. These results further highlight that several meaningful edits can be made per scene. Typical failure modes can also be observed: 1) some objects are unaffected by slot removal, 2) some edits have side effects (e.g. another object disappearing or changing its appearance), and 3) multiple different edits have the same (or a very similar) effect. These failure modes likely originate in part from the unsupervised nature of the OSRT base model, which sometimes assigns multiple slots to a single object, or does not decompose the scene well. Fully "imagined" objects (i.e. objects which are not visible in the input views and therefore not encoded in the Object Slots) further generally cannot be directly edited in this way. Some of these issues can likely be overcome in future work by incorporating object supervision (as done for MultiShapeNet), and by devising a mechanism by which "imagined" objects not visible in input views are similarly encoded in Object Slots.

D.2 Sampling from DORSal using DDIM vs. DDPM

In Table 5 we additionally report results when sampling from DORSal using 256 steps of DDPM. The latter lends itself better when using longer sampling chains as is reflected in an improvement in

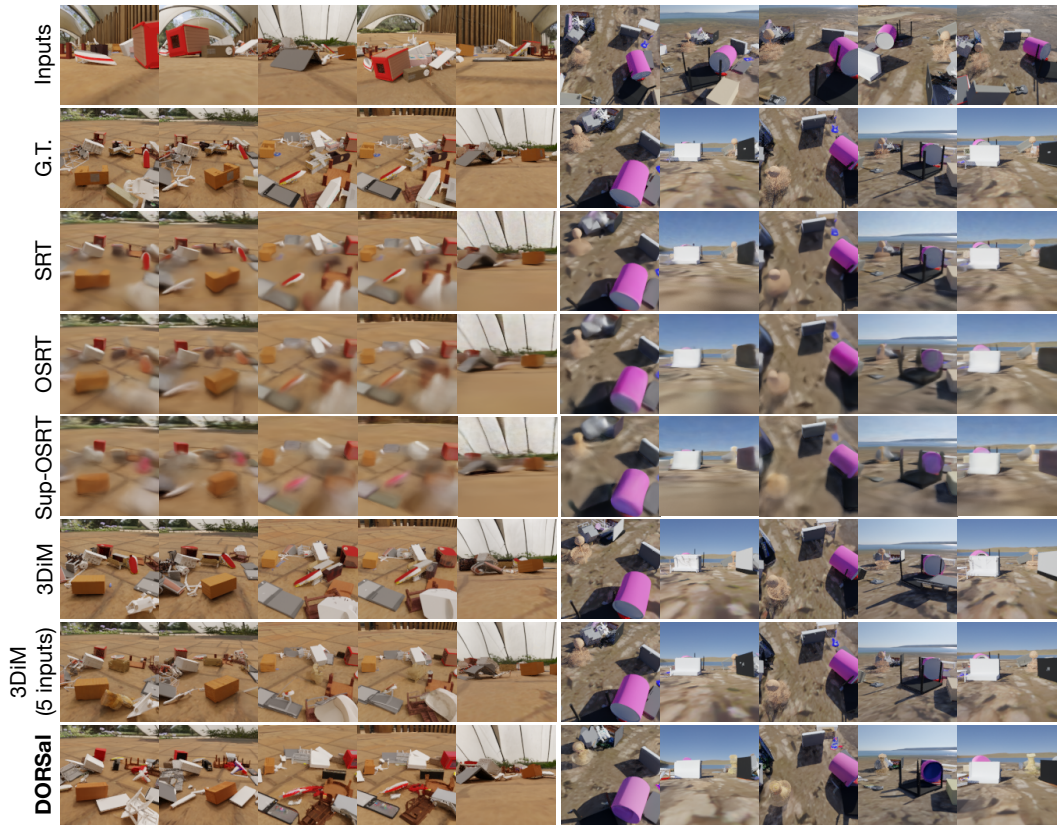


Figure 11: **Novel View Synthesis (MultiShapeNet)**. Qualitative results incl. input views (top tow) for additional MultiShapeNet scenes. We further include Sup-OSRT, which is trained using segmentation supervision (and provides the Object Slots for DORSal on MultiShapeNet), and a version of 3DiM that is conditioned on 5 ground-truth input views.

Table 4: **Novel-View Synthesis**. Evaluation on 1 k test scenes from MultiShapeNet and Street View.

	MultiShapeNet				StreetView			
Model	**PSNR↑**	**SSIM↑**	**LPIPS↓**	**FID↓**	**PSNR↑**	**SSIM↑**	**LPIPS↓**	**FID↓**
3DiM	18.20	0.559	0.287	10.94	12.68	0.283	0.477	15.58
3DiM (5 input)	21.46	0.707	0.182	8.20	12.25	0.271	0.557	34.47
DORSal	18.76	0.557	0.266	11.01	16.05	0.421	0.361	16.24

FID and PSNR [46]. Notice how in this case, when using the same procedure for sampling, DORSal consistently outperforms 3DiM.

D.3 Comparison to 3DiM using Additional Input Views

The stochastic conditioning procedure used during sampling from 3DiM can be initialized with an arbitrary number of ground-truth input views. In the main paper, we follow the implementation details from Watson et al. [51] and use a single ground-truth input view. However, because DORSal conditions on Object Slots computed from five input views, it would be informative to increase the number of input views to initialize 3DiM sampling accordingly. The results for this experiment are reported in Table 4, where it can be seen how 3DiM performs markedly better on MultiShapeNet in this case. In contrast, on Street View the opposite effect can be seen, where 3DiM performs markedly worse in this case.

Table 5: **Novel-View Synthesis (DDIM vs. DDPM)**. Evaluation on 1 k test scenes from MultiShapeNet and Street View.

	MultiShapeNet				Street View			
Model	**PSNR↑**	**SSIM↑**	**LPIPS↓**	**FID↓**	**PSNR↑**	**SSIM↑**	**LPIPS↓**	**FID↓**
3DiM	18.20	0.559	0.287	10.94	12.68	0.283	0.477	15.58
DORSal	18.76	0.557	0.266	11.01	16.05	0.421	0.361	16.24
DORSal (DDPM)	18.99	0.557	0.265	9.00	16.36	0.425	0.356	14.62

Table 6: **Scene Editing**. Evaluation on 1k test scenes of MultiShapeNet. All metrics reported in %.

Model	**Edit-mIoU**	**Edit-FG-mIoU**	**Edit-ARI**	**Edit-FG-ARI**
OSRT	43.4	43.1	38.2	79.6
Sup-OSRT	52.0	50.0	79.4	75.5
DORSal (64x64)	47.4	45.8	67.9	70.0
DORSal (128x128)	36.6	35.0	53.5	64.6

We hypothesize that this difference is due to how well 3DiM performs after training on these datasets. On MultiShapeNet, 3DiM achieves a better training loss and renders novels views that are close to the ground truth. Hence, initializing stochastic conditioning with additional views, will help provide more information about the actual content of the scene and thus help produce better samples. In contrast, 3DiM struggles to learn a good solution during training on Street View due to large gaps between cameras (and the increased complexity of the scene) and resorts to generating target views close to its input view. Hence, increasing the diversity of the ground-truth input views, will cause the model to generate views that lie in between these, which hurts its overall performance.

D.4 Full Edit Metrics (incl. Background Slots)

Table 6 presents quantitative results for scene editing when additionally including the background slots during evaluation. For Edit-mIoU and Edit-ARI effectively the same conclusions hold true: DORSal performs well compared to OSRT and attains performance close to that of Sup-OSRT. Further, it can be observed how the smaller DORSal model is more capable of scene editing compared to the large model trained on higher resolution data. A noticeable difference to the results reported in Table 2 can be observed for OSRT, where we see a large drop in performance when reporting Edit-ARI (compared to Edit-FG-ARI). This difference suggests that the object segments OSRT learns only loosely correspond to the precise object shape, but rather spans a part of the background as well. As a consequence, modifying individual slots will also significantly alter the background pixels in an undesirable way.

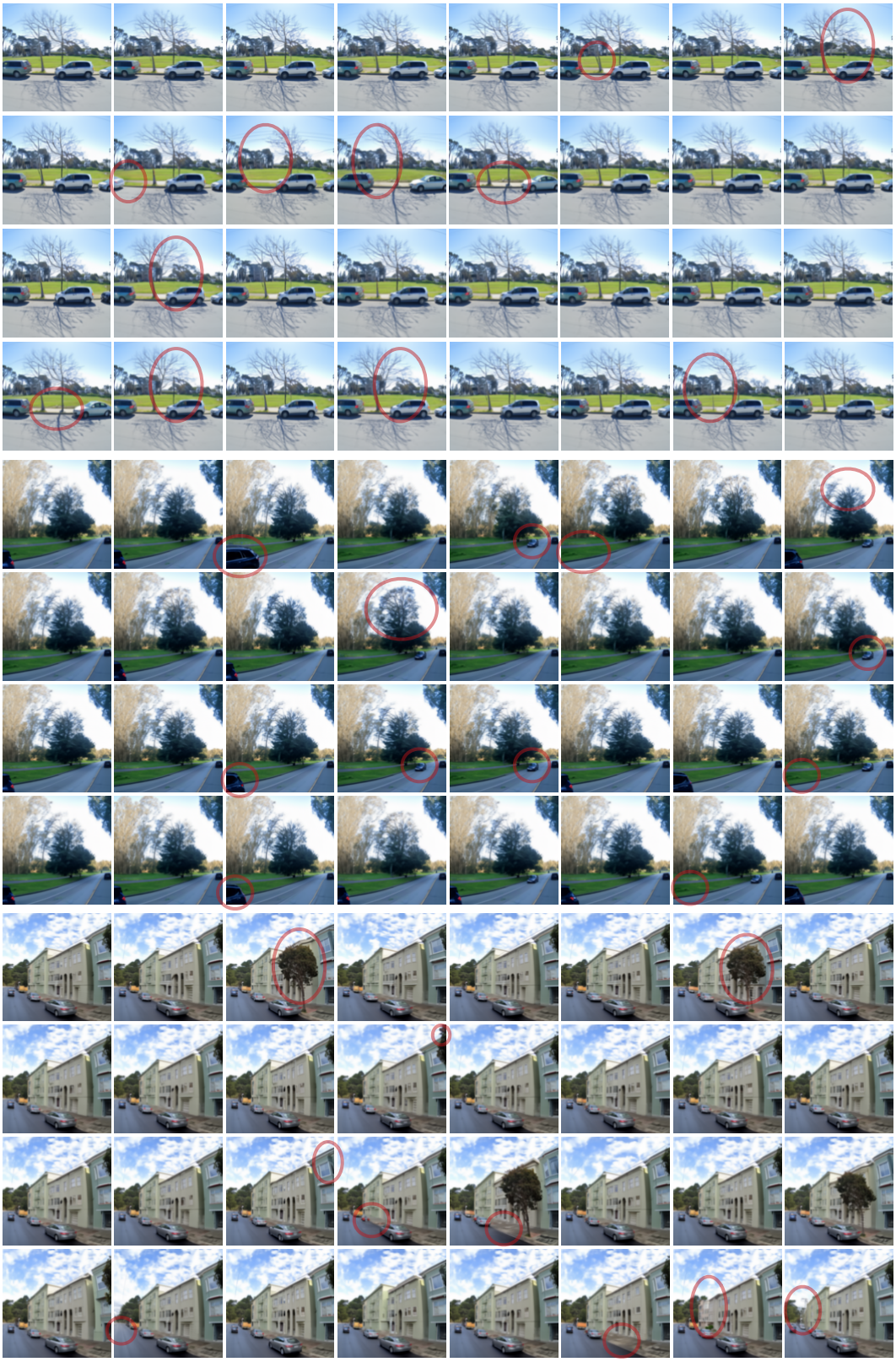


Figure 12: **Scene Editing (Street View)**. Exhaustive DORSal scene editing results for three Street View scenes, with one Object Slot removed at a time. Several examples where scene content differs are highlighted.