A Solution for Scale Ambiguity in Generative Novel View Synthesis

Fereshteh Forghani

A THESIS SUBMITTED TO
THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

GRADUATE PROGRAM IN COMPUTER SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO

JULY, 2024

Abstract

Generative Novel View Synthesis (GNVS) involves generating plausible unseen views of a scene given an initial view and the relative camera motion between the input and target views using generative models. A key limitation of current generative methods lies in their susceptibility to scale ambiguity, an inherent challenge in multi-view datasets caused by the use of monocular techniques to estimate camera positions from uncalibrated video frames. In this work, we present a novel approach to tackle this scale ambiguity in multi-view GNVS by optimizing the scales as parameters in an end-to-end fashion. We also introduce Sample Flow Consistency (SFC), a novel metric designed to assess scale consistency across samples with the same camera motion. Through various experiments, we demonstrate our approach yields improvements in terms of SFC, providing more consistent and reliable novel view synthesis.

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisor, Marcus Brubaker, and my committee member, Kosta Derpanis, for their support and guidance throughout my Master's studies. They taught me that research is less about having the best results and bold numbers and more about posing insightful questions and being principled in how we approach them. Their passion and excitement have reassured me that I am on the right path. I am truly grateful for their influence.

I would also like to sincerely thank my collaborators and friends, Jason Yu and Tristan Aumentado-Armstrong. They have taught me a lot and provided invaluable help and guidance.

Additionally, I extend my heartfelt thanks to my fellow members of the CVIL lab. You have not only been fantastic role models and collaborators but, most importantly, wonderful new friends. I have learned so much from each of you, and our shared experiences have enriched my journey tremendously.

I would also like to thank Borna Barahimi, whose support and encouragement have been unwavering. His advice and positivity have been a steady source of strength, especially during challenging times.

Last but not least, I want to thank my parents, Hossein and Parisa, and my uncle, Amir, for their endless love and belief in me. I am grateful for their sacrifices to help me pursue my dreams. To my entire family and friends, I am incredibly fortunate to have you by my side on this path. Your support has been invaluable.

Thank you all for everything. I couldn't ask for more.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Novel View Synthesis (NVS) is a well-known challenge in computer vision. In NVS, the aim is to generate new views of the scene, given one view and the corresponding camera poses. The camera motion between the generated and conditioning views can be small or large.

Inferring unseen content or occluded regions of the scene only from information in one image of the scene makes NVS a highly uncertain task. Hence, using generative models as a solution has become common. Generative Novel View Synthesis (GNVS) methods [14, 21, 22, 37, 40, 41] use generative priors to generate randomly plausible samples from the conditional distribution of generated views conditioned on given view and camera poses.

Scale ambiguity, on the other hand, is an intrinsic limitation in monocular visual odometry (VO). Visual odometry is the process of obtaining camera motion using single or multiple images [26]. Visual odometry can be carried out using both monocular and stereo

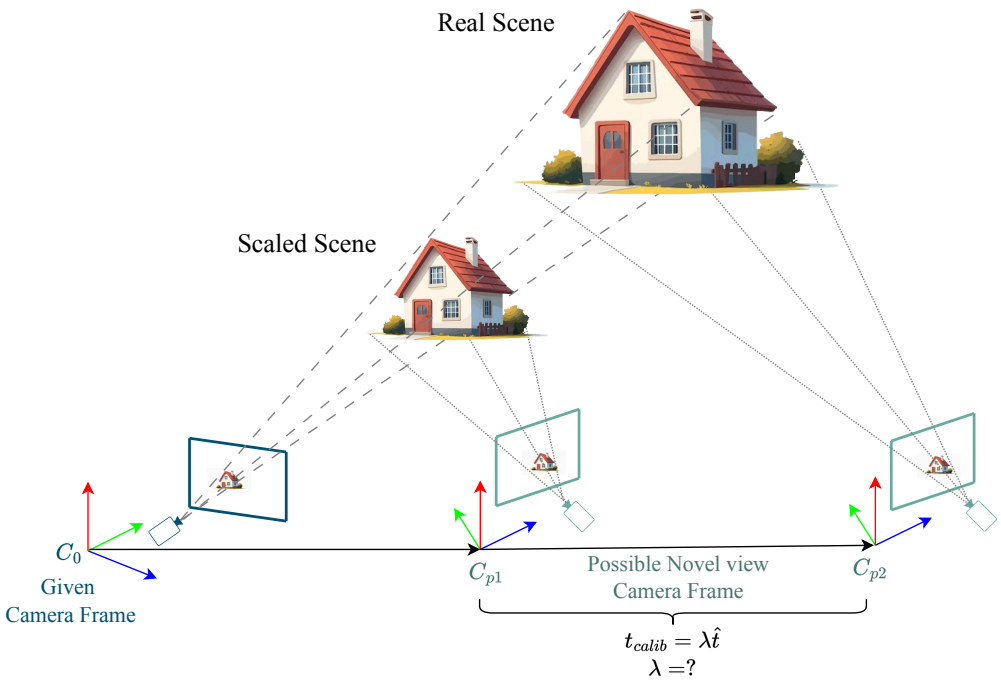


Figure 1.1: Using two images of a scene, relative rotation and translation between the conditioned and target frames can be deduced; however, the absolute scale of the scene still remains unknown.

cameras. Monocular VO is popular due to its low cost and the fact that most datasets can only satisfy monocular viewing conditions. However, monocular VO suffers a drawback: it cannot recover the absolute scale information based on the monocular VO techniques alone. In other words, the monocular VO methods can only recover the camera motion and scene geometry information up to an arbitrary absolute scale.

As shown in Figure 1.1, having frames C_0 and $C_{pi}, i = \{1, 2, 3, \cdots\}$, we can not determine the distance between C_0 and the target frame since images captured at $C_{pi}, i = \{1, 2, 3, \cdots\}$ are the same [38]. We get the same image by scaling both scene and camera

translation by the factor of λ. This λ factor is the reason for scale ambiguity in monocular VO.

Algebraically, we can formulate the geometry between two views of a scene with epipolar constraint, given as $x_2^T E x_1$, where x_1 and x_2 are corresponding points in the first and second frames, respectively. E, known as the essential matrix, can be derived using either the 8-point [15] or the 5-point [18] algorithm. Then, by decomposing E as $E = \hat{T}R$, where R is the relative camera rotation, and $\hat{T}$ the camera translation up to a scale, $\hat{T}$. Due to the homogeneous property of the decomposition equation, the translation component can only be recovered up to an unknown scale factor. This unknown scale factor is the root cause of scale ambiguity in camera pose estimation techniques that use point correspondences to recover relative motion.

1.1 Scale Ambiguity in GNVS

GNVS methods require images and corresponding camera poses for training. Datasets that fulfill such requirements are multi-view datasets. Available multi-view datasets generally use images that are not calibrated with camera poses and, hence, try to estimate the poses for each frame with pose estimation methods, such as SLAM-based methods [4], which all suffer from scale ambiguity. As a result, an "unknown" scale is introduced to the camera parameters and scene geometry.

Several attempts have been made to reduce the effect of scale ambiguity in these datasets. For example, in RealEstate10K [42], poses are scaled such that the 10th percentile of the depth set is 1.25m. These approaches, however useful, do not solve the

global scale ambiguity. Generative models, e.g., diffusion models [11], used in GNVS methods, learn the input data distribution and sample from it during inference. Therefore, GNVS methods inherit this scale ambiguity due to their dependence on uncalibrated data in multi-view datasets, which we demonstrate next.

We perform a simple experiment to show the effect of scale ambiguity in GNVS methods. We use PolyOculus [41] and generate multiple samples conditioned on the same view of the scene and a fixed lateral camera motion. Ideally, we would expect all generated images to translate the scene content the same amount relative to the conditioned image. However, in Figure 1.2, we can see that the generated samples vary significantly with the same camera motion.

Now, the question arises: what if we use multiple views for conditioning? In this case, the model is expected to get more spatial information, reducing the ambiguity in scale. Theoretically, having more than one conditioning view should resolve the scale ambiguity. This means the model can more reliably estimate the true positions and movements of scene contents, as it is less likely to be misled by scale ambiguities present in a single view.

We empirically show that the variance among samples is reduced by using more than one conditioning view. As an extension to the previous experiment, we generate samples conditioned on two views of the scene instead of one. Then, we measure the edge (bolded with lines in Figure 1.2) movement from the conditioning image to generated images by pixels in both settings (conditioned on one and two views). By comparing the variance among edge movements of the two groups, we observed that the edge movement has much less variance when conditioning on two views, which can be seen in Figure 1.3. The

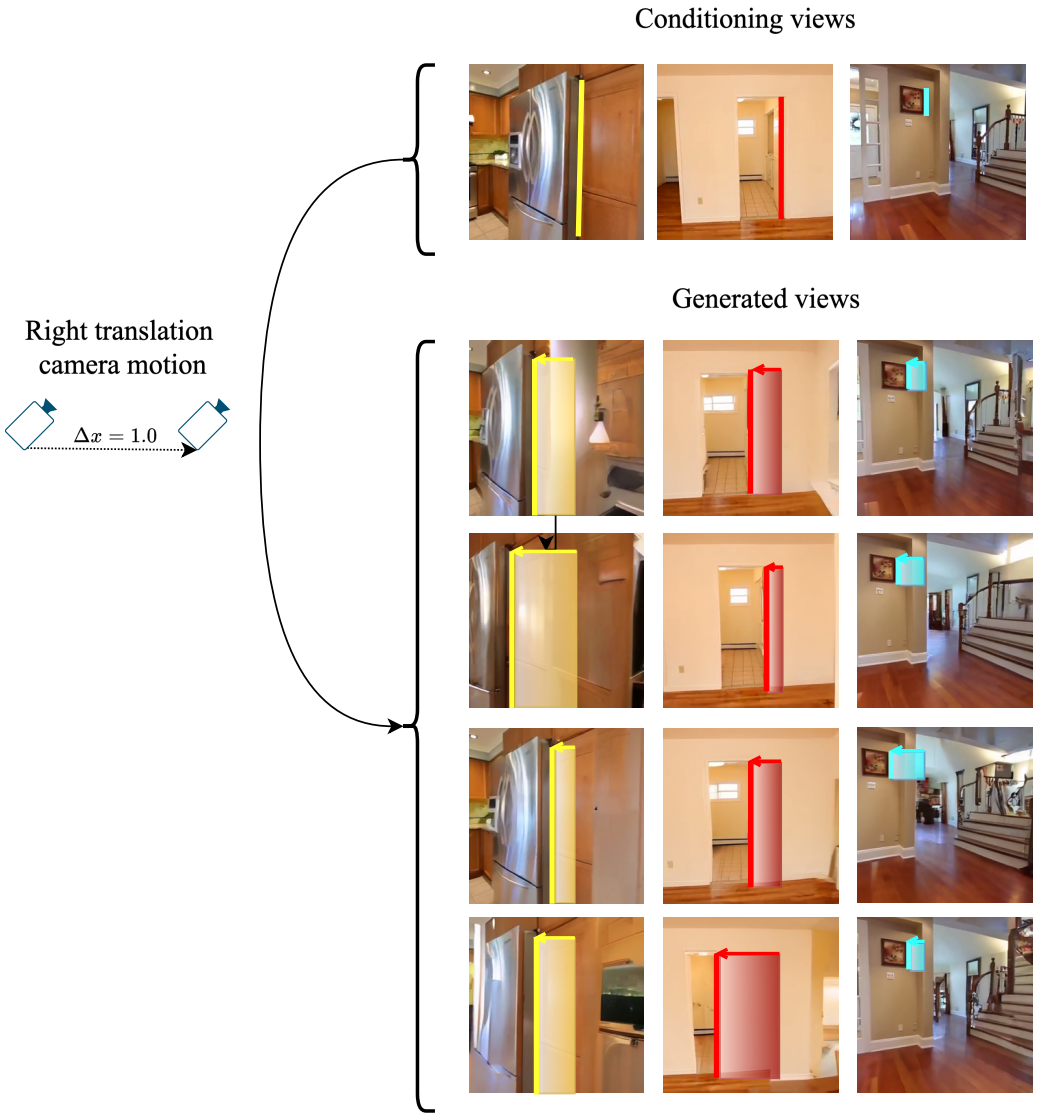


Figure 1.2: Sampling multiple times from a camera position that only contains an x-axis translation from the conditioned camera pose indicates that there is high variance in the movement scale in the model. The highlighted areas indicate the amount of edge movement in pixels.

reduction in movement variance is also visible in the generated samples shown in Figure 1.3. This experiment shows that GNVS models do get a sense of scene scale if they have access to more conditioning views.

Through the above experiments, we conclude that generative models learn distributions over the range of scales in the dataset and, at inference time with only one conditioning view, generate a view with the scene scale sampled from that distribution. This ambiguity in scale is a challenge in GNVS models, which we aim to address.

Some previous GNVS methods have developed heuristics to normalize scene scales across a dataset. Previous work [24] proposes a scene normalization scheme and an estimation of the scene scales to address this problem. Another method [31] introduces a scale-invariant view synthesis approach that uses multiplane images (MPI) as scene representations, and a per-scene scale factor is computed by minimizing a log-squared error between the predicted disparity map (predicted from the MPI) and point set of each scene. Such heuristics, although useful to some extent, do not solve the scale ambiguity in input data completely because a coherent scale is still lacking. To solve this, we propose optimizing a per-scene scale in the GNVS training process.

Due to having no available ground truth scale, we face the challenge of measuring the performance of the scale-learned generative models compared to previous ones, which did not use any scale information in the training process. Previous methods mostly use reference-based metrics, such as PSNR [31], or visually show heatmaps of the variance of the Sobel edge maps [24] to demonstrate improvements. However, to our knowledge, there exists no metric to compare the consistency of scale learned by GNVS models. To this end, we propose a metric based on optical flow, which measures the movement between

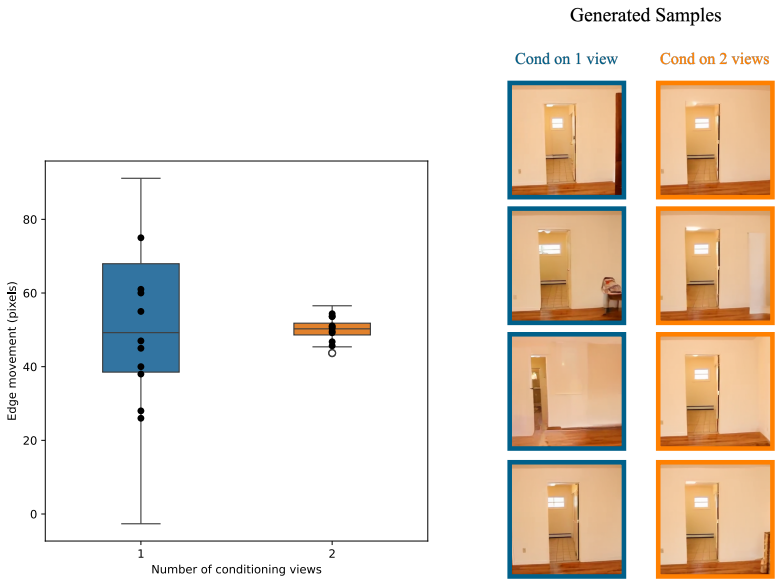


Figure 1.3: To compare the effect of the number of conditioning views on the edge movement variance, we visualize the edge movement in 10 samples as boxplots. On the right, we depict four samples generated conditioned on one and two views, respectively.

two views of the scene. First, we compare the optical flow between the conditioned view and multiple samples with the same camera movement. Then, we compute the pixel-wise variance of estimated optical flows. This variance can be shown as a heatmap, and its median is used as a quantitative point of comparison. The rationale behind designing such a metric is to measure the variance of scale in the samples generated by the model by measuring the movement variance. This metric measures the sample variance by analyzing the optical flow variance, where a lower variance indicates that the model has a more consistent sense of scale.

Our proposed scale learning schema is agnostic to the GNVS method. To demonstrate the effectiveness of our method, we adopt a recent state-of-the-art GNVS model, PolyOculus [41]. Although PolyOculus uses a latent diffusion model [2], it is still computationally

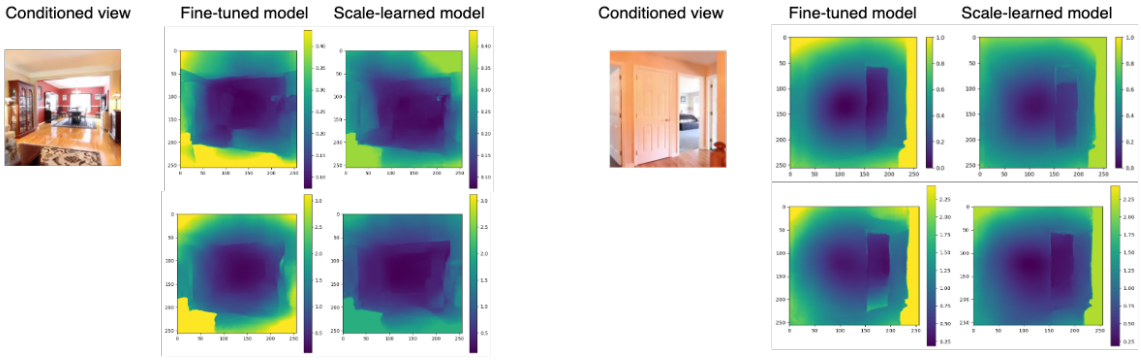


Figure 1.4: Example optical flow variance heatmaps of two scenes adapted from the 1k experiment. Darker colors correspond to lower optical flow variation, suggesting samples with more scale consistency.

expensive to train. Therefore, we first conduct experiments on a model fine-tuned on a smaller set of scenes to demonstrate the effectiveness of our proposed approach, and then we use the entire training set to train the model. For the experiment with a subset of the training set, we start from a PolyOculus checkpoint and fine-tune the pretrained model on a subset of RealEstate10k training set of size 1000, once with optimizing the scales (*scale-learned model*) and once without optimizing the scales (*fine-tuned model*). Fine-tuning without optimizing the scales is necessary since fewer training scenes generally reduce the sample variance. Our goal is to evaluate the effectiveness of scale learning, so we aim to ensure that the models differ solely in the scale learning. Finally, we use the entire train set to train the model to optimize the scales, and we compare it against PolyOculus publicly available checkpoint (*previous model*).

We compare the median optical flow per-pixel variance over 30 and 100 different samples of test set scenes for the validation experiment on 1k scenes and the final model trained on the entire training set, respectively. Results show improvements in both cases, reducing sample variance in the model trained to optimize the scales. We compare heatmaps,

histograms, and the median of the per-pixel variance of the optical flows over different amounts of translations over the camera trajectories from the dataset. Results are available as tables and heatmaps in Section 4.

As a preview, we visualize variance heatmaps of the optical flows of multiple samples from the 1k experiment models in Figure 1.4. These visualizations show that scale optimization during training reduces randomness from scale ambiguity in the model.

To conclude, we identify and explore the scale ambiguity problem in GNVS models, which, in this thesis, we address by proposing a method to learn the scales in the training process of the generative model. Also, to address the lack of a metric for scale noise quantification, we propose a Sample Flow Consistency (SFC) in which we use motion variation as a proxy for scale. We empirically show our proposed method outperforms baselines and that our metric can effectively measure scale entropy in the generated samples.

Chapter 2

Background

In this chapter, we start by defining the concepts of scale ambiguity and novel view synthesis, setting the stage for a deeper exploration. We then continue with generative novel view synthesis, providing a comprehensive explanation of the diffusion models, which is the most common method. Finally, we review key related works that tackled the scale ambiguity issue in the context of novel view synthesis, offering insights into how these approaches differ from ours.

2.1 Scale Ambiguity

Simply put, a small object nearby and a large object far away can look identical to a monocular camera. This scale ambiguity makes it impossible to determine the absolute scale of a scene (e.g., the height of a house in Figure 2.1b) from images alone.

Figure 2.1a illustrates this issue by scaling both the object and its distance from the camera. Essentially, suppose we scale the entire scene by a factor k and the camera pa-

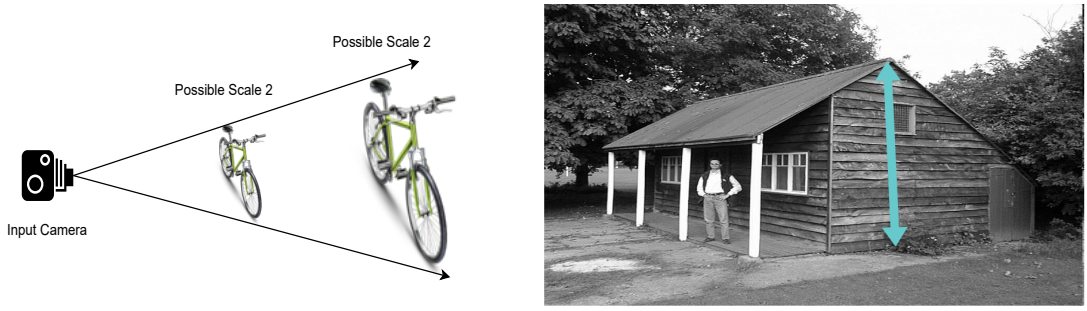
rameters by $\frac{1}{k}$. The projection of the scene is given by (2.1), where x represents the image coordinates, P denotes the camera parameters, and X refers to the real-world coordinates. In this case, the image projections remain unchanged.

$$x = PX = \left(\frac{1}{k}P\right)(kX), \tag{2.1}$$

Scale ambiguity is a well-known challenge in monocular visual odometry (VO) [6, 30, 35]. In previous work [6], two assumptions were introduced to address the scale ambiguity: (i) constant camera height from the ground throughout the video, and (ii) the ground being a horizontal plane on every image frame. Then, they utilize a local feature map and optimize the scales such that each feature corresponds to its real-world location. Another previous work [30] introduces Ground Point Extraction (GPE) and Ground Points Aggregation (GPA) algorithms to extract and aggregate ground points and optimize the ground plane parameters. Others [35] propose a scale recovery algorithm using 2D and 3D correspondences to find a relative distance between the camera and the ground plane, $\hat{h}$, and find the scale using the ratio of $\hat{h}$ to the absolute distance $\hat{h}$.

The scale ambiguity problem is introduced in datasets that use monocular VO, such as ORBSLAM [17] and COLMAP [4], for camera pose estimation. Monocular Simultaneous Localization and Mapping (SLAM) [3] systems yield consistent camera poses, but the metric units are ambiguous, requiring an unknown "scale" factor to convert the representation to metric scale.

Various methods have been developed to recover the absolute scale information for monocular VO. For example, adding another camera to create a binocular system can provide scale information using the length of the stereo baseline. However, these methods



(a) (b)

Figure 2.1: a) A nearby small object and a distant large object can look the same to a monocular camera despite representing different scene scales [24]. b) It is impossible to estimate the true height of the house only using the image alone [7] ©*2008, IEEE.*

have some limitations, *e.g*. the stereo pair images must be taken at the same time interval, which can be achieved by synchronizing shutter speed, which requires much more effort to set up. In the following subsections, we examine two datasets that use camera pose estimation methods and consequently exhibit scale ambiguity problems.

2.1.1 RealEstate10k

The RealEstate10k dataset [42] comprises YouTube videos of real estate tours. The camera pose parameters for each video clip were recovered using ORBSLAM2 [17], which is built on a monocular feature-based ORB-SLAM. SLAM methods process a series of frames to build a sparse or semi-dense 3D reconstruction of the scene and simultaneously estimate the current frame's viewpoint consistently with the 3D reconstruction. An example of a scene in this dataset is shown in Figure 2.2.

As noted by the authors, determining the global scene scale for each video clip is

Figure 2.2: An example of a point cloud estimations of a scene in RealEstate10k [42]. Images on the left are random frames of the video, and on the right, we show a point cloud estimation of the scene generated using the scene video frames.

impossible, and ORBSLAM reconstructs camera poses up to an arbitrary scale. Since the Multiplane image (MPI) representation, based on layers at specific depths of the point cloud, is used for view synthesis, they propose to "scale-normalize" each sequence relative to the estimated 3D point clouds. They compute the 5th percentile depth among all point depths for each frame in the sequence. By aggregating these depths across all cameras in a sequence, they derive a set of "near plane" depths. The sequence is then scaled such that the 10th percentile of this depth set is 1.25m, aligning with their MPI representation's near plane of 1m.

This scale normalizing method works well for their MPI approach but does not resolve the global scale ambiguity in the dataset. An unknown scale remains for each scene.

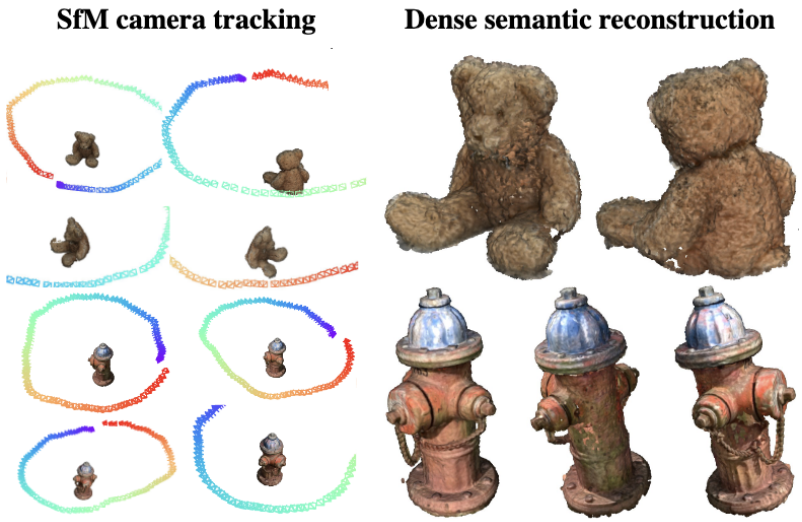


Figure 2.3: Example of object-centered scenes in CO3D camera poses and reconstructions from dense point cloud estimations [20] ©*2021, IEEE*. Camera poses extracted using SfM camera tracking and 3D reconstructions of the objects are depicted on the left and right, respectively.

2.1.2 Common Objects in 3D (CO3D)

The CO3D dataset [20] consists of object-centered videos collected via Amazon Mechanical Turk (AMT). Camera properties (extrinsic and intrinsic matrices) and dense point clouds for each video are estimated using the COLMAP structure-from-motion (SfM) pipeline [4]. Figure 2.3 shows examples of SfM camera trajectories and dense semantic reconstructions from this dataset.

In their experiment, they zero-center each point cloud and normalize the scale by dividing point cloud coordinates by the average per-axis standard deviation. However, an ambiguous scale still exists for each scene.

2.2 Novel View Synthesis

Novel view synthesis (NVS) is the problem of predicting unseen views of a scene given the seen views. NVS has long been a challenging task in computer vision [13, 1, 5]. Challenges arise from uncertainties such as elements outside the field of view and unseen or occluded parts of the scene in the given views. In addition, there is an uncertainty in the global scene scale, *e.g*. the perceived sizes and distances of objects, introduced by uncalibrated data. By uncalibrated data, we mean datasets consisting of images of a scene without any camera pose attached to them. The source of such ambiguity in the data scale is the pose recovery methods, like monocular SLAM, which was explained in the previous section. Such methods yield consistent camera poses; however, the metric units of the poses are unknown.

This scale ambiguity in the input view(s) camera parameters leads to uncertainty in predicting new views. For instance, in Figure 2.1a, given the image from the input view, the generated image from a novel camera could either be a small motorcycle on the left or a large motorcycle on the right.

Models used to solve NVS are generally trained and evaluated on multi-view datasets, which, due to the use of SLAM-based methods for camera pose estimation, have an ambiguous scale in their camera parameters. Therefore, models trained on such datasets also learn the ambiguous scales.

Recent advancements in NVS models have introduced several techniques to address the inherent challenges. One such method is ZeroNVS [24], a 3D-aware diffusion model designed for single-image novel view synthesis in complex, real-world scenes. ZeroNVS [24] employs a novel camera conditioning parametrization and normalization scheme

to mitigate depth-scale ambiguity, improving the diversity and realism of synthesized views. Another approach involves training on mixed datasets, such as CO3D [20], and RealEstate10K [42], which provide a variety of scenes captured with different camera settings. This diversity helps the model generalize unseen data better. 4DiM [36] tackles this challenge by calibrating RealEstate10k [42] scales with monocular depth estimates. This technique helped mitigate the scale issue. Moving forward, we will go through each of the methods mentioned above in detail.

Additionally, Score Distillation Sampling (SDS) has significantly enhanced the quality of generated views by leveraging large 3D datasets and ensuring photorealistic and 3D-consistent NVS. However, standard SDS can lead to mode collapse, particularly in background regions, resulting in less diverse synthesized images. To address this, techniques like SDS anchoring have been developed, which improve the diversity of synthesized views by sampling several "anchor" novel views and using them as conditions for SDS [19].

Another noteworthy method is the use of monocular depth estimation, which aims to predict depth maps from single images. Despite the progress in this area, models trained on relative depth datasets often face challenges due to unknown depth shifts and varying camera focal lengths, which can distort 3D scene reconstructions. To overcome these issues, a two-stage framework has been proposed [39], where depth is first predicted up to an unknown scale and shift and then refined using 3D point cloud data to estimate the depth shift and camera focal length.

2.3 Generative Novel View Synthesis

Generative models have recently become a popular solution for the NVS problem, known as Generative Novel View Synthesis (GNVS). The reason for such popularity is the NVS problem's uncertain nature and generative models' ability to learn and generate various possible outputs.

Several previous works addressed the GNVS problem [14, 21, 22, 37, 40, 41]. Infinite Nature [14] uses a Generative Adversarial Network (GAN) [9] based architecture for view generation. This approach focuses on natural scene generation, leveraging adversarial training to produce high-quality and realistic novel views. GeoGPT [22] is based on a combination of conditional generative modeling and neural discrete representation learning (VQVAE) [33]. This method employs a variational approach to capture and generate diverse scene geometries, allowing for flexible scene manipulation. LookOut [21] uses an encoder-decoder architecture with an autoregressive transformer to condition on up to two previous views. This method enhances temporal coherence in novel view sequences, making it suitable for video applications.

PhotoCon [40] employs a diffusion-based generative model designed to generate long sequences of consistent novel views from a single image. It uses an autoregressive conditioning chain that achieves geometric consistency, but it suffers from error accumulation coming from using generated frames as the conditioning information.

PolyOculus [41] uses a set-based diffusion model able to condition and generate any arbitrary number of views. This approach is particularly powerful for handling complex scenes with multiple objects and varying viewpoints, utilizing diffusion processes to ensure consistency across generated views.

Recent advancements in GNVS introduced new techniques to further improve the quality and diversity of generated views. ZeroNVS [24] employs a 3D-aware diffusion model for single-image novel view synthesis in real-world scenes. This model addresses the challenges of in-the-wild multi-object scenes with complex backgrounds by training on diverse datasets and introducing camera conditioning parametrization and normalization schemes to handle depth-scale ambiguity. ZeroNVS also introduces Score Distillation Sampling (SDS) anchoring to enhance the diversity of synthesized views, particularly in the background, overcoming the limitations of traditional SDS methods.

Generally, generative models used in GNVS methods learn the arbitrary visual scales presented in the dataset. At inference time, the model effectively randomly samples a scene scale from the distribution over all the training scene scales and uses it as the sample scale. Therefore, we observe high variance in scene scale by sampling conditioned on the same view and camera motion multiple times. This scale ambiguity remains a significant challenge in GNVS, as it impacts the consistency and realism of the synthesized views. Methods like ZeroNVS attempt to mitigate this by incorporating scale normalization techniques and leveraging large-scale, mixed dataset training to improve model robustness and generalization.

2.4 Diffusion Models

Diffusion models are a family of generative models that learn to generate new samples from the data distribution by reversing a fixed forward noising process [11, 28]. Figure Figure 2.4 depicts the diffusion process. To formulate the process, let x_0 be an input

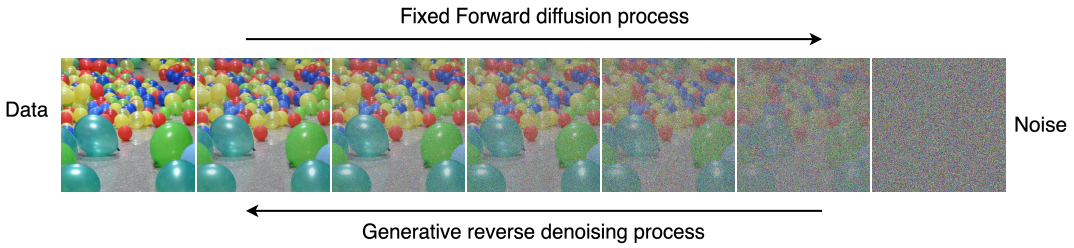


Figure 2.4: Diffusion process consists of two processes: Forward diffusion process which gradually adds noise to data and reverse denoising process that learns to generate data by denoising [32].

data point. The forward Gaussian noising process is divided into $t \in \{1, \cdots, T\}$ smaller noising steps. Each step adds a fixed small Gaussian noise to the input data. Assuming $q(x)$ is the input data distribution, each forward step is defined as follows:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}). \tag{2.2}$$

In the above equation, β_t is designed so that $x_T \sim \mathcal{N}(0, \mathbf{I})$. The reverse diffusion process also consists of small Gaussian denoising steps and is parameterized by θ such that:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)). \tag{2.3}$$

In this formulation, $\Sigma_\theta(x_t, t)$ is usually fixed and thus not learned. The backward function is generally implemented with a noise estimator function $\epsilon_\theta(x_t, t)$. This noise estimator takes a noisy input, x_t and predicts the noise added to x_t in the previous time step. By subtracting the predicted noise from the input, we get x_{t-1}, a less noisy version of x_t. The

loss used to train the backward process given by

$$\mathcal{L} = \|\epsilon - \epsilon_\theta(x_t, t)\|^2, \tag{2.4}$$

where ϵ refers to the added noise in time step t in the forward process. At inference time, we start from $x_T \sim \mathcal{N}(0, \mathbf{I})$ and apply the following equation T times to get a sample:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t z_t, \tag{2.5}$$

where $z_t \sim \mathcal{N}(0, \mathbf{I})$, $\alpha_t = 1 - \beta_t$, and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. Diffusion models can also be conditioned on additional inputs. In those cases, the conditions will be added to the noise estimator as an input $\epsilon_\theta(x_t, t, \mathbf{c})$. In the case of image generation, it is common to use a U-Net [23] for the noise estimator model.

Training diffusion models is computationally expensive since the model needs to repeatedly evaluate its loss across different noise levels (timesteps). This involves running a forward pass through the network for many randomly sampled timesteps for each data point. It becomes even more expensive as the dimensionality of the input increases, such as when images are used as high-dimensional input data. To reduce computational costs, the latent diffusion model (LDM) [2] is used. LDM employs a pre-trained encoder, $z = E(x)$, to map the input data into a lower dimension latent space. The diffusion model is then trained on the latent space, and finally, a pre-trained decoder, $x = D(z)$, maps the latent space back to the input data space.

2.5 Scale Ambiguity in NVS

Some related works in GNVS [24, 31, 36] mention the scale ambiguity problem of their models. They try to mitigate the problem by using an estimate of the scales to normalize different scene scales in the dataset. They show that the scale calibration makes the trained model robust to scale ambiguity. We go through some of these related works in the following subsections.

2.5.1 ZeroNVS: Zero-Shot 360-Degree View Synthesis from a Single Real Image

ZeroNVS [24] introduces a 3D-aware diffusion model for single-image novel view synthesis (NVS) in scenes including multiple objects and complex backgrounds. To address the challenges of scale ambiguity and background diversity, the authors propose several key innovations.

A significant contribution is their new camera conditioning parametrization and normalization scheme, which manages varying scales and perspectives from different datasets. This method is crucial for improving the model's robustness to scale ambiguities inherent in mixed data sources.

ZeroNVS incorporates a novel camera conditioning parametrization that allows for the normalization of scenes to address scale ambiguity. This scheme includes:

- **6DoF+1 Representation**: This representation captures all possible camera positions and orientations, including the field of view, making it suitable for diverse in-the-wild scenes. They use $\mathbf{M}_{6DoF+1}(D, F, E, I, j)$ for this representation, which

computes an embedding given the depth maps, D, and extrinsics, E, the field of view, F, and the indices i, j of the input and target view, respectively.

- **Normalization scheme**: Inspired by Stereo Magnification [42], ZeroNVS use the same method to find scene scales. They take each scene's depth-map 5-th quantile and use the 10-th quantile of this aggregated set of numbers as the scene scale q. The normalization is done by rescaling the translation component of the camera extrinsics by $\frac{1}{q}$.

This robust scale handling enables the ZeroNVS model to generate 3D-consistent and diverse novel views from a single image. This model leverages the normalization technique from previous work, Stereo Magnification [42]. This approach helps normalize the scales, but the global scale of the scenes remains ambiguous. Such a normalization scheme ensures that relative scales are consistent, but the absolute scale of the entire scene is not determined.

2.5.2 Single-View View Synthesis with Multiplane Images

Tucker and Snavely [31] introduce a method for synthesizing novel views from a single input image using multiplane images (MPIs). MPIs are a layered, camera-centric 3D representation, where each layer consists of fronto-parallel planes at different depths from the camera. The approach involves the following steps:

- **Prediction of MPIs**: The model predicts an MPI directly from a single image, estimating the scene's depth and the disoccluded content (hidden areas that become visible in the novel view).

- **Warping and Compositing**: Each plane is warped to the target viewpoint using homographies and composited to render the novel view, involving bilinear sampling and the over operation for layering.

Scale-Invariant Synthesis: A key innovation in this work is the use of scale-invariant synthesis to handle the scale ambiguity inherent in single-view setups, where Visual SLAM and structure-from-motion cannot determine absolute scale without external information. Therefore, each training sequence is equally valid if the world is scaled up or down by any constant factor. The authors address this by:

- **Computing a Scale Factor**: The scale factor is computed as the exponential of the average error between the predicted disparity log and inverse depth log. This scale factor is applied during rendering to ensure the rendered image does not vary with the scale of the input viewpoints and point set.

- **View Synthesis Loss**: This loss encourages the rendered image at the target viewpoint to match the ground truth, combined with a smoothness loss on the synthesized disparity and a sparse depth supervision loss.

This scale-invariant synthesis uses a pre-computed scale factor to adjust the rendering process. While this ensures scale consistency across different scenes, it still suffers from the same problem as ZeroNVS [24], not solving the global arbitrary scene scales, just normalizing them to some extent.

2.5.3 4DiM: Controlling Space and Time with Diffusion Models

4DiM [36] is a cascaded diffusion model capable of generating 4D consistent views. The model incorporates both spatial and temporal dimensions from one or more images of scenes, along with corresponding camera poses and timestamps.

One of 4DiM's key innovations is its strategy to use calibrated data to resolve the scale ambiguity associated with camera pose estimation in NVS tasks. A significant limitation in prior work is the use of uncalibrated datasets, such as those posed via SfM methods, which yield camera poses that are only accurate up to an arbitrary scale. This lack of metric information complicates the control over camera motions and the specification of camera poses in meaningful units during the inference stage.

To overcome this challenge, the 4DiM framework uses a calibrated version of the RealEstate10K dataset called cRE10K. The calibration process involves regressing the unknown metric scale from the monocular metric depth estimates provided by a state-of-the-art depth estimation model [25]. This calibration step is crucial as it aligns the 3D points inferred by SfM with metric depths, enabling 4DiM to generate views with accurate metric scales, thereby significantly improving the pose alignment and scale of the generated scenes.

The 4DiM model is trained on diverse data sources, including posed 3D images and unposed video data. The calibrated metric scale data from cRE10K plays a key role in teaching the model to understand and generate scenes with correct scale and spatial relations. Beyond improving pose alignment, 4DiM leverages this metric scale data with ground-truth camera poses to control the camera's trajectory in space and time.

Moreover, 4DiM introduces a novel evaluation metric based on SfM distances, de-

signed to assess the accuracy of pose alignment in generated views. Unlike traditional metrics that often overlook scale inconsistencies, the SfM-based metric is scale-invariant. The SfM distances metric computes the relative error in camera positions and the angular deviation in camera rotations by comparing the predicted camera extrinsics of generated view to ground truth poses. This metric is important for evaluating the model's ability to generate metrically accurate samples of 3D scenes.

In conclusion, 4DiM attempts to mitigate the scale issues by calibrating datasets like RealEstate10K, thereby improving the metric accuracy of the generated scenes. They use depth estimates as a proxy for scales and show improved results with the calibrated data. However, their way of estimating the scales adds a data calibration step to the training, whereas we propose to do such scale estimation during training. Moreover, they introduce an SfM-based metric to account for the scale inconsistencies. This metric, however, has the downside of having to run an SfM-based algorithm for all the evaluation scenes, making it slow and computationally expensive, and it may fail in case of not finding enough correspondences.

2.6 PolyOculus: Simultaneous Multi-view Generation

PolyOculus [41] addresses the GNVS challenge by generating multiple plausible views of a scene given a limited number of known views. In this thesis, we build on the PolyOculus model as a GNVS model capable of set-based conditioning and generation, enabling us to establish the problem of scale ambiguity in the case of conditioning on a single view and show the problem mitigates when conditioning on more views. Therefore, we go through

this method in more detail.

Unlike traditional methods that generate a single image at a time and rely on an ordered autoregressive generation approach, PolyOculus can simultaneously generate multiple self-consistent new views. This is particularly useful for maintaining image quality over large sets of images, even when the generated views have no natural sequential ordering, such as loops or binocular trajectories.

PolyOculus's primary innovation is its set-based generative model, which can condition on any number of views and generate sets of new views in a permutation-invariant manner. This model addresses the limitations of existing GNVS methods, such as error accumulation over long sequences and inconsistencies between generated frames, by leveraging set-based generative modeling. PolyOculus ensures that each generated view is coherent with others in the set, thus maintaining high image quality and consistency across multiple views.

The method employs a diffusion model [11], which iteratively refines random noise to generate realistic images. To handle the computational complexity associated with high-dimensional data, PolyOculus utilizes a latent diffusion model (LDM) [2]. This approach maps the input data into a lower-dimensional latent space for training, significantly reducing the computational burden and allowing for efficient processing. During inference, the model decodes the latent representations back to the original image space, producing high-quality novel views.

PolyOculus also introduces novel techniques for conditioning on multiple views, enhancing its ability to generate diverse and accurate scenes. The model incorporates a permutation-invariant set-based architecture, which ensures that the order of input views

does not affect the output. This characteristic is crucial for applications where the input views do not follow a sequential order, such as when using stereo cameras.

In comparison with other GNVS methods, PolyOculus offers several advantages:

- **Simultaneous Multi-view Generation:** Unlike autoregressive models that generate one view at a time, PolyOculus can produce multiple views simultaneously, ensuring consistency and reducing computation time.

- **Permutation-invariant Architecture:** The set-based approach allows PolyOculus to handle unordered inputs, making it versatile for different applications.

- **Efficient Training with Latent Diffusion:** By operating in a lower-dimensional latent space, PolyOculus achieves efficient training and inference, enabling the generation of high-quality views without excessive computational resources.

In conclusion, PolyOculus significantly advances GNVS by combining a set-based generative model with a diffusion framework, addressing key challenges such as view consistency. Future work can further enhance its capabilities by integrating additional depth cues and exploring more robust scale normalization techniques, paving the way for even more accurate and diverse novel view synthesis.

However, PolyOculus, like other GNVS models, faces the challenge of scale ambiguity. This issue is particularly prominent when generating views conditioned on a single input view. Scale ambiguity arises because the global scene scale cannot be accurately determined from a single image, leading to uncertainties in the synthesized views. For example, a small object close to the camera can appear similar to a larger object farther away, making it difficult for the model to infer the correct scale.

To address this, PolyOculus uses multiple conditioning views in training. This technique helps the model get a sense of scale for the scene and learn to generate views that are consistent in scale when conditioned on more than one input view. However, when conditioned on only one view, which is the focus of this thesis, the scale ambiguity problem still exists.

2.7 Summary

In this chapter, we start by presenting previous works addressing scale ambiguity in VO and two multi-view datasets that inherit scale ambiguity from camera pose estimation techniques. Then, we glance at several methods addressing NVS, including GNVS models, specifically focusing on PolyOculus, which is state-of-the-art. At last, we review three related works addressing scale ambiguity in NVS methods, where several heuristics were studied to mitigate the scale noise. Most of these heuristics normalize scales of the data and are not capable of estimating the global scale. They also struggled with quantifying the effect of such heuristics on scale noise. In this thesis, we address the first challenge by proposing a scale-learning scheme in which the GNVS model optimizes scales as parameters and the second challenge by measuring variations in motion as a proxy for scale.

Chapter 3

Technical Approach

In this section, we explain our scale-learning method in which we propose to learn a scale parameter for each training scene as a part of training the generative model. We also explain the motivation and definition of the proposed metric, Sample Flow Consistency (SFC), designed to measure scale noise.

3.1 Scale Formulation

In the multi-view datasets such as RealEstate10k [42], although the scale is unknown due to the use of SfM [27], the camera poses do have a consistent scale in the translation component for each training example. To address the scale ambiguity challenge in GNVS methods, we assume that the "base" scales are reasonably close to the true scales. This assumption allows us to initialize the scales to a uniform value of 1 and then refine them through optimization during training. Formally, let $S = \{s_i\}_{i=1}^{N}$ represent the per-scene scales for N scenes. We aim to optimize these scales to improve the accuracy and consis-

tency of the reconstructed scenes.

The scales are parameterized through an exponential transformation and clamping, ensuring that they remain positive and within a reasonable range. Specifically, we define the scales as follows:

$$s_i = exp(a[\beta_i]_{-1}^{+1}), \tag{3.1}$$

where β_i are the learnable per-scene parameters we optimize during training, ensuring they stay within the range $(-1, +1)$ through a clamping function $[z]_{-1}^{+1} = \mathbf{Clamp}(z, -1, +1)$. The parameter a is a learning speed hyperparameter, which we typically set to one. This approach ensures that the scales evolve smoothly and remain within a practical range for further processing.

Optimizing the scales in this manner has several important implications. First, it allows the model to dynamically adjust the scales based on the observed data, potentially improving synthesized views' realism. However, this approach also introduces particular challenges. For instance, the optimization process must be carefully controlled to prevent divergence or instability, especially given the sensitivity of the scales to small changes in β_i. Additionally, while the exponential formulation ensures positivity, it can also lead to rapid changes in scale, necessitating a cautious choice of the learning rate.

3.2 Scale Optimization

Optimizing the scale parameters, $\beta = \{\beta_i\}_{i=1}^n$, is a crucial step in achieving accurate and consistent scene reconstructions. This optimization is performed using a separate

optimizer from that used for the denoiser. Due to the inherent lack of ground truth scale in monocular depth estimation, the optimization process is not directly supervised with explicit loss terms for the scales. Instead, it relies solely on denoising loss signals, making the scale learning process slow and noisy.

3.2.1 Regularization

We introduce a regularizer term to prevent the optimization process from converging to excessively large β values. This regularizer helps constrain the magnitude of the β_i values, promoting stability in the scale optimization process. The regularizer term is defined as follows:

$$r = c \sum_{i=1}^{n} \beta_i^2, \tag{3.2}$$

where c is a hyperparameter that controls the strength of the regularization. The regularizer term is computed at each iteration using the β_i values available in the current batch. This term is then added to the diffusion loss to be used in the overall optimization process.

3.2.2 Optimization Process

The loss function used to train the scales along with the model parameters is

$$\mathcal{L} = \underbrace{\sum_{i=1}^{n} ||\epsilon - \epsilon_\theta((z_{i_1}, t, c_{i_1}), ..., (z_{i_N}, t, c_{i_N}), t)||^2}_{\text{Denoising loss}} + c \sum_{i=1}^{n} \beta_i^2, \tag{3.3}$$

which consists of two terms: denoising loss and the regularizer. In the denoising loss N is the randomly selected number of conditioning frames for scene i, c_{i_N}s are corresponding camera poses of each frame which we calibrated with our learned scales, $c_{i_N} = [R_{i_N}|s_i T_{i_N}]$, and ϵ_θ is the denoiser which takes in all the inputs including the camera poses and estimates the actual added noise, ϵ. The last term is the regularizer explained above.

The optimization of the scale factors and the U-Net [23] parameters is handled by two distinct optimizers. We use two different optimizers for optimizing scale factors and U-Net parameters. This separation of optimizers allows for finer control over the learning dynamics of each set of parameters, which is essential given their different roles in the reconstruction process.

The PolyOculus model [41] is implemented using Denoising Diffusion Probabilistic Models (DDPM) [11], which utilize a denoising loss, as shown in (2.4). The noisy nature of this loss contributes to the slow convergence of the scale optimization process.

3.2.3 Challenges in Scale Learning

A significant challenge in the training process is determining when the scales are fully learned, as there is no ground truth to validate the scale estimates directly. This lack of direct supervision necessitates careful monitoring of the training process and the reconstruction quality to infer the adequacy of the learned scales.

Additionally, the reliance on reconstruction loss signals means that the optimization can be affected by noise in the data and the inherent stochasticity of the diffusion model. This necessitates using techniques such as regularization and careful tuning of learning

rates to ensure stable and effective learning.

3.3 Sample Flow Consistency Metric (SFC)

A significant challenge in the training process is measuring the effect of scale learning. The goal of learning per-scene scales for the training examples is to reduce sample variance. However, no direct evaluation metric is available to measure this effect. To address this gap, we propose using the per-pixel variance of optical flow between frames as a metric to evaluate scale consistency.

3.3.1 Motivation

Existing evaluation metrics for NVS mostly focus on image quality (*e.g*. FID [10]), reconstruction (*e.g*. LPIPS and PSNR), geometric consistency (*e.g*. TSED [40]), and accuracy in camera pose alignment (*e.g*. SfM distances and SfM Keypoints [36]). However, these metrics neither capture the statistical entropy induced by the scale noise nor measure the inconsistencies in scale among the generated views. Reconstruction metrics do not target artifacts induced by scale variation and hence are overwhelmed by other sources of error (*e.g*. differences in semantic content, especially further from any observed view). Similarly, FID is not suitable for measuring the impact of scale noise, as it only compares distributions of images in a virtually arbitrary feature space with no explicit notion of scale. Finally, while TSED captures geometric consistency, it is insensitive by nature to scene scale changes. This is because TSED relies on the error incurred by points incorrectly moving off the epipolar lines, but such scale changes move points *along* those

lines. The SfM-distances and SfM-keypoints metrics, introduced in 4DiM [36], also use SfM-based methods (*i.e*. COLMAP), which inherently capture the camera motion up to an arbitrary scale. Hence, we lack a measure of scale variation among the generations.

To this end, we introduce a metric that uses the apparent motion of objects in the scene as a proxy to quantify the scale (since the same camera motion will move the objects in the scene differently whenever the scale changes). Thus, the variance of this apparent motion among generated views, using the same conditioning view and target camera parameters, is a natural measure of deviations in generated scales.

3.3.2 SFC Definition

Formally, we define our metric as follows. For each scene, we generate a sample set of size n conditioned on the same first frame, f_{cond}, with the same camera motion between the conditioned and generated frames, c as:

$$F = \{f^i_{\text{gen}}\}^n_{i=1}. \tag{3.4}$$

Next, we compute optical flow between f_{cond} and all members of F using an off-the-shelf optical flow model, RAFT [29]. The set of optical flows between the conditioning frame and the generated frames with camera motion c is defined as:

$$\mathcal{OF}_c = \{\text{RAFT}(f_{\text{cond}}, f_{\text{gen}_i})\}^n_{i=1}. \tag{3.5}$$

However, one shortcoming of optical flow methods is when new, unseen content is added to the scene or some pixels get occluded due to changing the camera's viewpoint. In

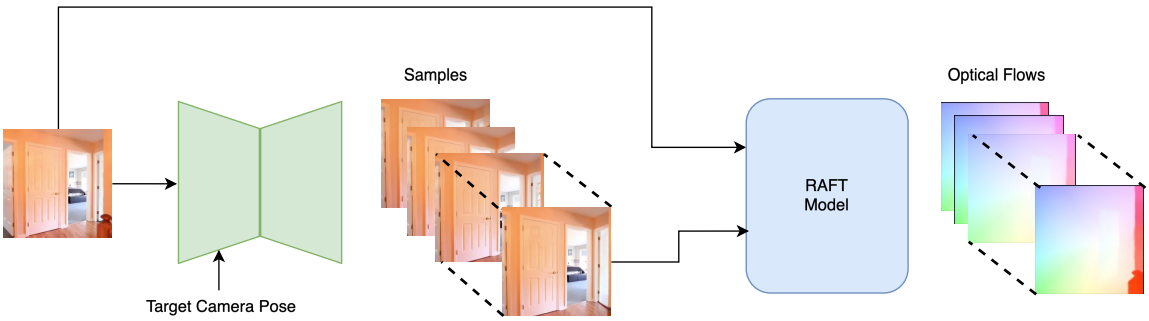


Figure 3.1: This figure depicts the generation process of multiple samples with the same conditioning frame and target camera pose and optical flows between the conditioning frame and the generated one.

such pixels, the optical flow is likely to be inaccurate and an outlier. To mitigate the effect of such occluded or outside-the-field-of-view pixels, we need to identify such outliers and mask them out from the optical flow map.

To generate the mask for two frames, $\mathcal{A}$ and $\mathcal{B}$, we use the notion of cycle consistency in back-and-forth optical flows [34]. We first compute two optical flows, one from frame $\mathcal{A}$ to frame $\mathcal{B}$, $\mathcal{OF}_{\mathcal{A}\to\mathcal{B}}$, and the other from frame $\mathcal{B}$ to frame $\mathcal{A}$, $\mathcal{OF}_{\mathcal{B}\to\mathcal{A}}$. Then, we translate frame $\mathcal{A}$'s pixels with $\mathcal{OF}_{\mathcal{A}\to\mathcal{B}}$ into $\mathcal{B}'$ and then translate pixels in $\mathcal{B}'$ with $\mathcal{OF}_{\mathcal{B}\to\mathcal{A}}$ into $\mathcal{A}'$. Now, we compare $\mathcal{A}'$ pixel positions with $\mathcal{A}$ and mask out pixels that are more than T pixels further in the x-axis or y-axis from their original location in $\mathcal{A}$. We define each value of mask, $\mathcal{M}_c$, at pixel ij as:

$$\mathcal{M}_{c_{ij}} = \begin{cases} 0 & \text{if } |\mathcal{A}'[i] - \mathcal{A}[i]| > T \text{ or } |\mathcal{A}'[j] - \mathcal{A}[j]| > T \\ 0 & \text{if the pixel goes out of image bounds} \\ 1 & \text{Otherwise} \end{cases} . \tag{3.6}$$

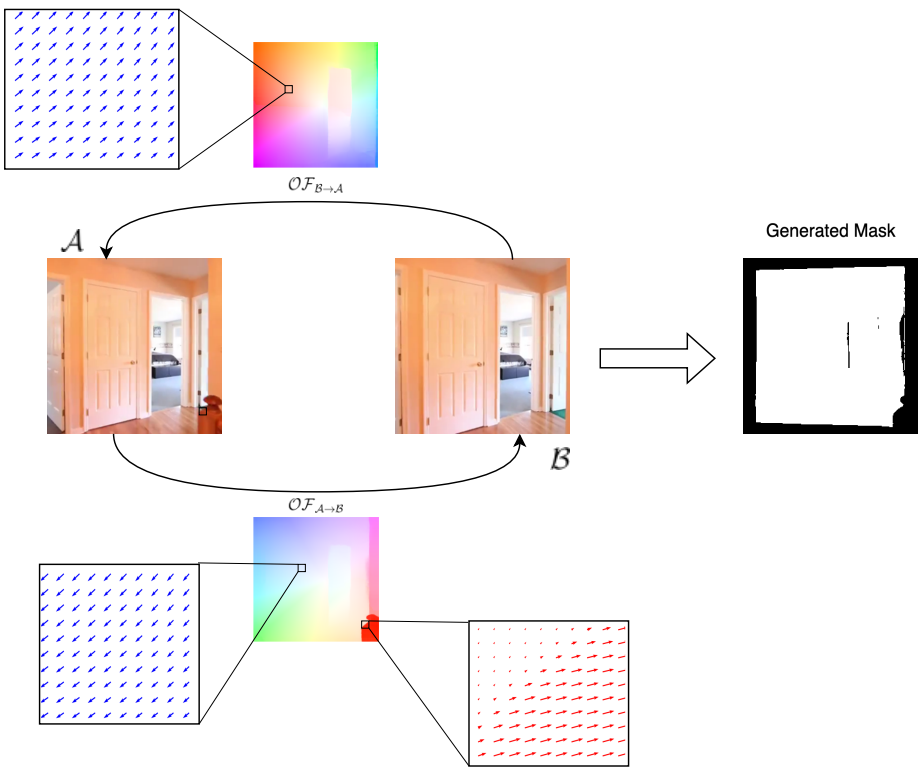


Figure 3.2: Outline of mask generation process for two frames $\mathcal{A}$ and $\mathcal{B}$ where black pixels are the masked ones. To simplify the mask generation process, optical flow vectors of two patches are shown: a masked patch and an unmasked patch. Red optical flow vectors correspond to a group of masked pixels because the pixels are moving out of the field-of-view by $\mathcal{OF}_{\mathcal{A}\to\mathcal{B}}$. Their optical flow vectors are larger and inconsistent with the neighboring pixels in frame $\mathcal{B}$. On the other hand, optical flow vectors shown in blue correspond to the same patch of pixels in $\mathcal{OF}_{\mathcal{A}\to\mathcal{B}}$ and $\mathcal{OF}_{\mathcal{B}\to\mathcal{A}}$ are in opposite directions and their addition result in a vector smaller than the masking threshold. In other words, the pixels in this patch get back to their location after translating with $\mathcal{OF}_{\mathcal{A}\to\mathcal{B}}$ and $\mathcal{OF}_{\mathcal{B}\to\mathcal{A}}$, respectively.

In this work, we set the threshold, T, to three. This threshold shows how sensitive we are in terms of cycle consistency. The lower the threshold, the more sensitive we are, meaning more pixels get masked in the process.

We depict the mask generation process in Figure 3.2. At last, to apply the mask to optical flows, we do a pointwise multiplication between the generated mask, $\mathcal{M}_c$, and each of the calculated optical flows, $\mathcal{OF}_c$:

$$\mathcal{MOF}_c = \{\mathcal{M}_c \odot \mathcal{OF}_i | \mathcal{OF}_i \in \mathcal{OF}_c\}. \tag{3.7}$$

We aim to measure the variance of motion in each pixel location. To this end, we compute the per-pixel median absolute deviation (MAD) among members of $\mathcal{MOF}_c$. We use MAD to mitigate the effect of optical flow outliers. This operation provides us with a map with the same dimensions as the input images, where each entry corresponds to the variation of optical flows at that particular pixel location. Hence, the MAD in pixel $[i, j]$ is calculated as

$$\mathcal{MAD}_c[i, j] = \text{Med}(|\mathcal{MOF}_c^k[i, j] - \overline{\mathcal{MOF}_c[i, j]}|), \tag{3.8}$$

where $\mathcal{MOF}_c^k$ corresponds to each member of $\mathcal{MOF}_c$ and $\overline{\mathcal{MOF}_c[i, j]}$ is the average of $\mathcal{MOF}_c$ in pixel location $[i, j]$. The desired outcome is a lower optical flow MAD among the scale-learned model samples. Therefore, we aggregate the per-pixel variances by computing the median over each variance map. The median is used to avoid the influence of outlier variance pixels. This median value, $\mathcal{SFC}_c$, is used to compare the scale consistency between the two models and is computed as:

$$\mathcal{SFC}_c = \text{Med}_{\text{pixels}}(\mathcal{MAD}_c). \tag{3.9}$$

3.3.3 Interpreting the Sample Flow Consistency Metric

The Sample Flow Consistency ($\mathcal{SFC}$) metric quantitatively measures how scale learning improves the generative model's understanding of scale. A smaller $\mathcal{SFC}$ indicates less variance in the samples generated with the same camera movement, suggesting better scale consistency. Furthermore, we expect the per-pixel variance maps of the scale-learned model to exhibit lower variance (darker areas) and more structure than those of the non-scale-learned model.

To aggregate the $\mathcal{SFC}$ values across multiple scenes, we compute the average $\mathcal{SFC}$ as follows:

$$\mathcal{SFC}_{\text{avg}} = \frac{1}{N} \sum_{n=1}^{N} \mathcal{SFC}_n. \tag{3.10}$$

This average provides an overall measure of sample consistency, allowing for a comprehensive comparison between models.

Figure 3.1 illustrates the process of generating samples and calculating the optical flow. The resulting optical flow heatmaps and $\mathcal{SFC}$ metrics offer insights into the model's performance in terms of maintaining scale consistency. By analyzing these metrics, we can assess the effectiveness of the scale-learning process and make necessary adjustments to

improve the model's performance. Figure 3.3 shows the effect of masking out occluded or outside the field-of-view pixels. Comparing the unmasked and masked heatmaps, we can see that the outlier optical flows have been removed; therefore, heatmaps became more accurate in the sense of showing the area with the highest optical flow variance. For a numerical comparison, we compare the max optical flow between the unmasked and masked optical flows in each example in Figure 3.3. In the first row of the figure, $\max$ optical flow variance in the fine-tuned model is 257.37, and in the scale-learned model is 109.85, whereas these numbers reduce to 96.04 and 21.12, respectively, in the masked optical flows. This meaningful reduction of $\max$ optical flow shows that masks are eliminating the pixels with outlier optical flows, resulting in a more robust and accurate SFC comparison.

3.4 Summary

In this chapter, we first went through the formulation and optimization of our scale-learning method. Next, we introduced our new metric, Sample Flow Consistency (SFC) and went through its definition. In the end, we briefly showed the effect of our proposed masking process in the SFC computation process.

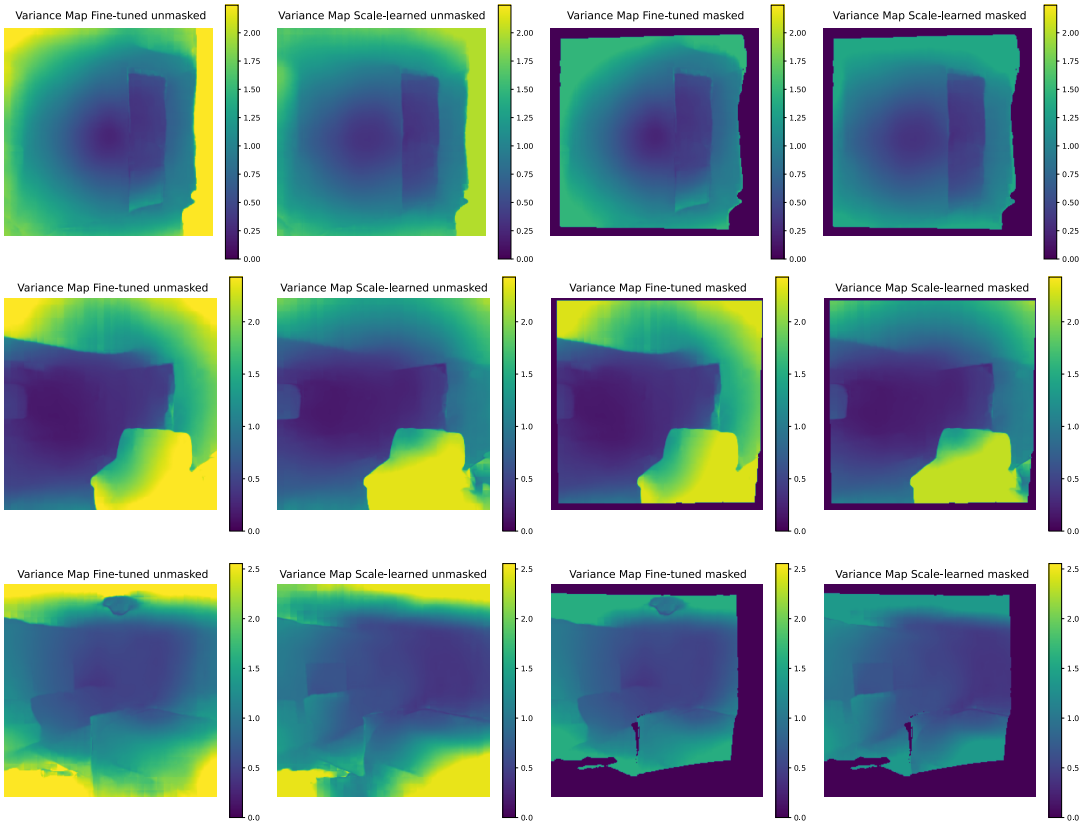


Figure 3.3: Here are three examples of unmasked and masked heatmaps of per-pixel optical flow variance. Each row corresponds to heatmaps of a test sequence. To generate each row, we first generate samples for that sequence with *Fine-tuned model* and *Scale-learned* model, and then for each model's samples, we compute a per-pixel variance of optical flows generated during SFC computation. This figure shows the benefit of masks in omitting the outliers and making the metric more robust.

Chapter 4

Experimental Results

In this chapter, we review the implementation details of our technical approach, define our quantitative metrics, and present both qualitative and quantitative results. We empirically show that our scale-learning approach outperforms the baselines. We also demonstrate the effectiveness of SFC in quantifying scale noise in the generated samples.

4.1 Implementation Details

We build on the PolyOculus [41] model, a multi-view latent diffusion model that leverages a DDPM [11] for the diffusion process, with U-Net [23] serving as its backbone. The latent representations of images are extracted through a VQ-GAN autoencoder [8], which effectively compresses the high-dimensional input while preserving essential structural details necessary for view synthesis. A critical component of the training process is learning a scale for each scene, where the scaling factor is dynamically adjusted as training progresses. The scaling coefficient, a, remains fixed at 1, while a regularization term is

applied with a coefficient $c = 0.0001$ to stabilize the optimization process.

To optimize the parameters of our model, we use different optimization strategies for the scale factors and the U-Net backbone. Specifically, we adopt the Adam optimizer [12] with a learning rate of 10^{-3} for adjusting the learned scales, ensuring that each scene is handled with an optimal scaling factor. For the U-Net parameters, we employ the AdamW optimizer [16] with a learning rate of 2.0×10^{-6}, which incorporates weight decay to prevent overfitting, particularly useful when training models with a large number of parameters over extensive iterations. This combination of optimizers helps ensure stable convergence despite the complexity of the task and the high dimensionality of the latent space.

Our training pipeline uses a batch size of eight to balance memory usage and convergence speed. Training is distributed across four NVIDIA L40 GPUs, while testing is performed on a single RTX6000 GPU. This hardware setup enables us to process large batches and complex scenes without sacrificing computational efficiency, which is crucial given the computational demands of diffusion-based models.

4.2 Dataset

RealEstate10K dataset [42] includes publicly available real estate tour videos on YouTube. These videos are segmented into disjoint sequences, with the corresponding camera parameters extracted using ORBSLAM2 [17]. The dataset offers a substantial collection of real, diverse, and structured environments, making it an ideal benchmark for evaluating novel view synthesis (NVS) methods. Its widespread adoption by several leading NVS

baselines [41, 40, 21, 22] further highlights its importance.

In this thesis, the videos are center-cropped and downsampled to a resolution of 256×256. The dataset provides a wide variety of indoor scenes with relatively large camera motions, making it a challenging dataset for NVS tasks.

4.3 Metrics

For our evaluations, we introduce the average Sample Flow Consistency (SFC) metric to measure how consistently each model captures scale across different samples with the same target camera pose. This metric focuses on the accuracy and coherence of the scale factor learned during training, providing an objective comparison between models. By computing the average consistency across samples, we can quantify the stability of the model's predictions.

In addition to the proposed metric, we evaluate the consistency and quality of the generated samples using standard image-based metrics. We evaluate based on TSED [40] to measure the percentage of consistent pairs in the generated samples. We also show results in terms of PSNR, LPIPS, and FID to assess the samples' reconstruction quality

In addition to the quantitative metric, we provide extensive qualitative analyses to further support our findings. This includes visual representations such as MAD heatmaps, which highlight regions of high and low optical flow consistency, offering a clear understanding of where models succeed or struggle in maintaining coherent scene structure. These heatmaps allow us to visually inspect the spatial distribution of errors, particularly in complex or occluded areas, where maintaining consistent scale is more challenging.

To complement these visual results, we also present histograms of per-pixel optical flow variances for several scenes. These histograms provide a statistical overview of the models' performance, showing the distribution of flow variances. By combining qualitative and quantitative results, we comprehensively evaluate each model's ability to maintain consistent scaling across diverse scenes.

4.4 Experimental Setup

Our objective is to evaluate each model's capacity to produce scale-consistent samples under specific camera movements. To this end, we generate 10 samples for each scene in the test set, with the same camera movement across all models. We use the translation vector's norm between the conditioned and generated frames as a standardized measure for comparison.

Rather than selecting specific frame numbers, which introduces variability in the amount of camera movement between scenes, we adopt a threshold-based approach. For each frame sequence, we identify the first frame where the norm of the camera translation exceeds predefined thresholds, $T = \{0.05, 0.1, 0.15, 0.2, 0.25, 0.3\}$. This ensures that the generated frames are selected based on a specific amount of camera movement rather than frame numbers, providing a more accurate and fair comparison. In this setting, we use ground truth frames to generate the masks used in computing SFC.

This method eliminates the potential inconsistencies that could arise from variations in camera translation when using fixed frame indices. Since the magnitude of camera movement directly influences the appearance of generated frames, controlling for translation

ensures that comparisons across different scenes and models remain valid, leading to a more robust evaluation of scale consistency across the test set.

Most trajectories in the RealEstate10k dataset [42] naturally contain larger forward-backward motions rather than lateral or vertical motions (*e.g.* most videos are walking into or out of a room). Therefore the model trained on this dataset is susceptible to being biased on forward-backward trajectories. We also design another experiment to test the model's performance in lateral and vertical motions.

To show the effect of scale learning in a more controlled setting, we explore camera motions in which we only translate the camera along one of the axes without any camera rotation and compare the performance in terms of SFC. In this setting, due to the lack of ground truth frames with the sampling camera motion, we generate SFC masks by comparing optical flows between the conditioning frame and the generated samples and then aggregate all masks by averaging them. To have a fair comparison among models, we average masks generated by each model's sample into one so that all the models use the same mask in SFC computations.

4.5 Results

We present qualitative results for the model trained on 1k scenes as a showcase of the method's effectiveness. For more comprehensive evaluations, we present quantitative results of the model trained on 70k scenes with both SFC and standard metrics. We assess the proposed method in terms of scale entropy, 3D geometric consistency, and image quality with meticulous experiments.

4.5.1 Model Finetuned on 1K Scenes

We qualitatively evaluate both scale-learned and fine-tuned models trained on 1000 training scenes. The effect of joint scale learning on the apparent movement of scene contents is shown with heatmaps and histograms of per-pixel optical flow variances. This is a showcase of our proposed method being successful in lowering the entropy of sample scales on a small chunk of the dataset.

We visualize the per-pixel optical flow variance as heatmaps in Figure 4.1 and the corresponding histograms in Figure 4.2. As shown in the heatmaps, the scale-learned model samples have the lowest per-pixel optical flow variance among all three heatmaps. The fine-tuned model samples also have less per-pixel optical flow variance compared to the previous model, which is the PolyOculus [41] checkpoint.

In Figure 4.2, we depict per-pixel optical flow variance histograms of samples from three different models of four test scenes. These histograms show that, first, fine-tuning the PolyOculus [41] model's checkpoint on 1000 scenes generally lowers the sampling variance. This is because fewer training samples are presented to the generative model. The second conclusion drawn from these histograms is that scale learning while training the model lowers the variance of optical flow variances in comparison to vanilla fine-tuning. These histograms empirically show that on a subset of the training set, the proposed scale learning approach can successfully lower the sample movement variance.

4.5.2 Model Finetuned on 70k Scenes

We now evaluate our proposed learning-based method by fine-tuning the model on all the RealEstate10K training scenes for 200 epochs, allowing us to compare its performance

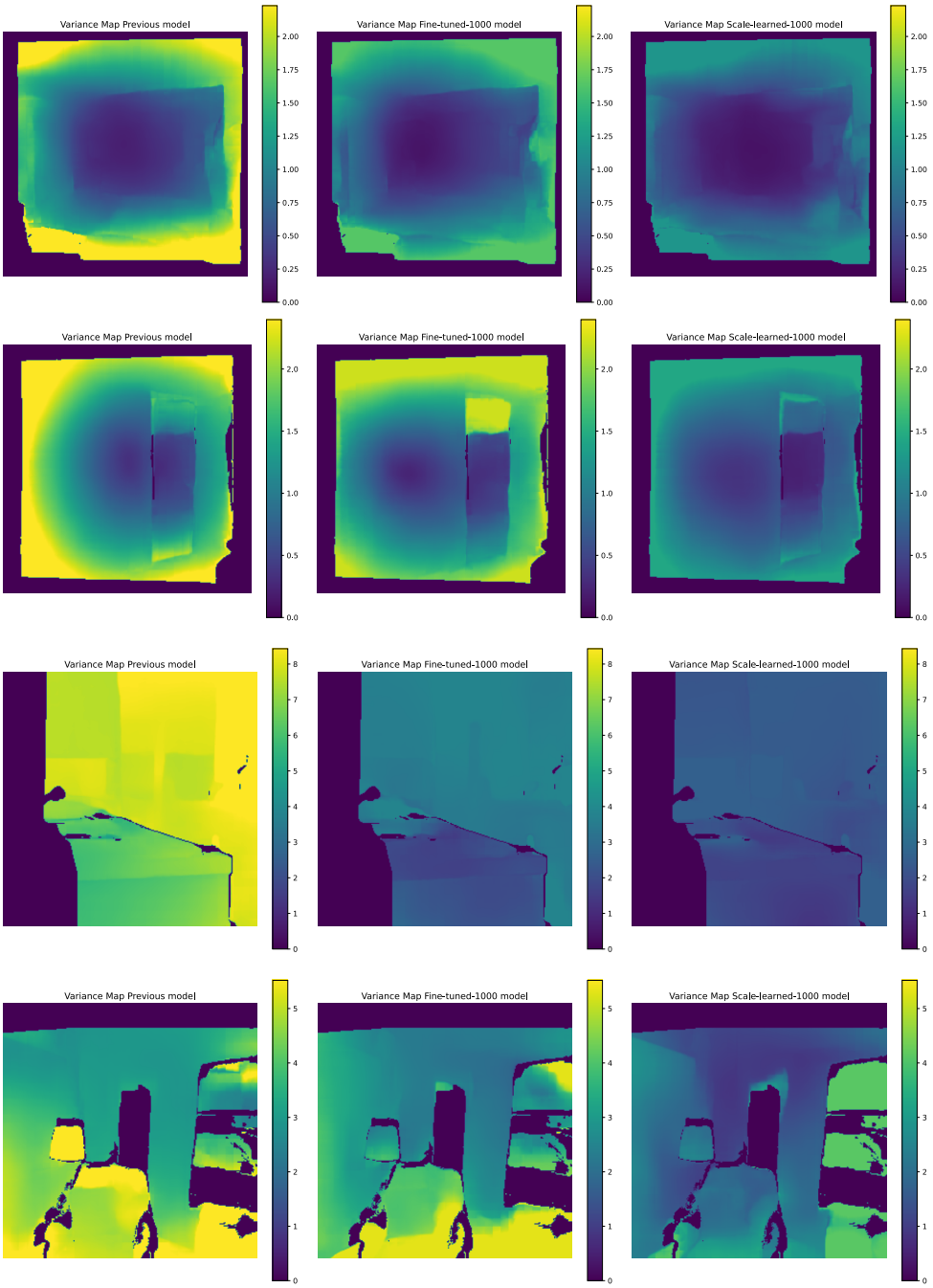


Figure 4.1: Example heatmaps of per-pixel optical flow variances among samples of our three baselines. As shown by the bar next to each heatmap, the darker the heatmap, the lower the per-pixel optical flow variance.

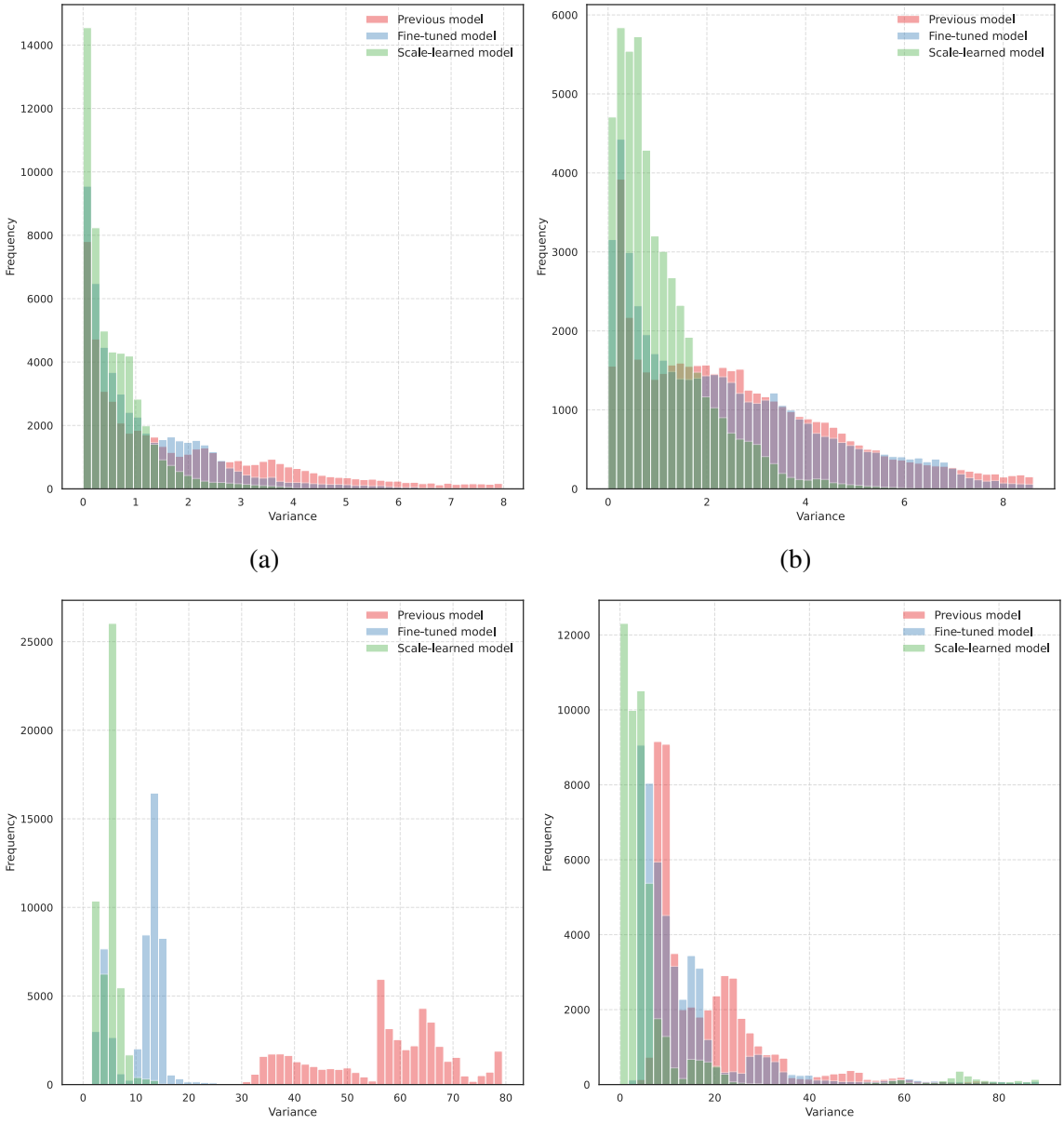


Figure 4.2: Each histogram shows the distribution of per-pixel optical flow variances of different model's generated samples of a scene. Histogram (a) to (d) correspond to scenes in Figure 4.1 from top to bottom, respectively.

Sample Flow Consistency ($\mathcal{SFC}$)(Avg/SD)↓						
Model \ Δt	0.05	0.1	0.15	0.2	0.25	0.3
PolyOculus$_{700}$	0.50/0.29	0.83/0.54	1.12/0.85	1.41/0.85	1.74/0.97	1.88/0.97
PolyOculus$_{900}$	0.51/0.21	0.90/0.52	1.23/0.64	1.46/0.79	1.75/1.00	2.06/1.13
ScaleLearned$_{900}$	**0.47/0.19**	**0.79/0.39**	**1.08/0.56**	**1.33/0.66**	**1.61/0.95**	**1.84/0.94**

Table 4.1: SFC results for generated samples of test scene with camera poses extracted from the test set. PolyOculus$_{700}$ and PolyOculus$_{900}$ are trained without any scale training for 700 and 900 epochs, respectively, and ScaleLearned$_{900}$ is trained with the proposed method starting from PolyOculus$_{700}$ for 200 epochs.

against the previous model, which lacked explicit scale training. We compare three models against each other: PolyOculus$_{700}$, which is the publicly available checkpoint of PolyOculus [41], PolyOculus$_{900}$, which is the PolyOculus model checkpoint trained for 200 epochs without any scale learning, and ScaleLearned$_{900}$, which is the PolyOculus model checkpoint trained for 200 epochs with our proposed scale learning approach.

Measuring Scale Variance (SFC). The comparison between the three models highlights a clear improvement in terms of the SFC metric. As shown in Table 4.1, the ScaleLearned model has the lowest average sample flow consistency (SFC) among the three models. This result indicates that learning scale during training enhances the model's ability to maintain a coherent sense of scale among each scene sample. We also report the standard deviation (SD) of per-scene SFC scores. Results show that the *ScaleLearned* model also has less variance in SFC among various scenes, showing that the improvements in SFC averages are meaningful.

The effect of scale noise on generations is observable in qualitative results as well. To show these effects more clearly, we create Sobel edge maps from each sample of a scene from ScaleLearned$_{900}$ and PolyOculus$_{900}$ models and overlay them to create heatmaps,

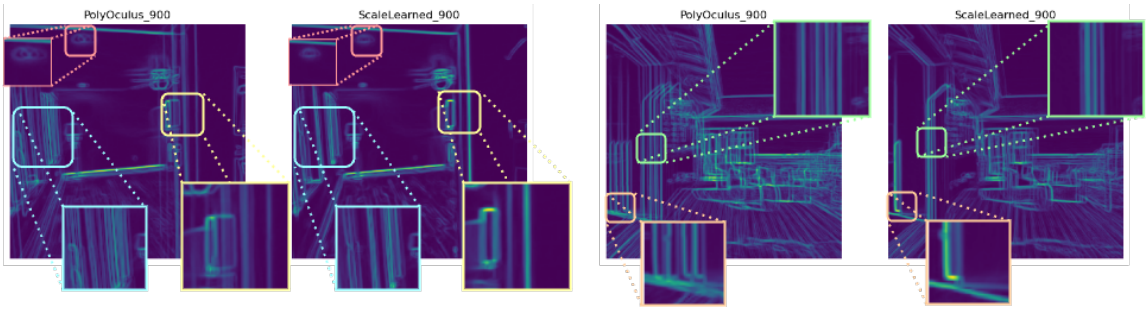


Figure 4.3: Heatmaps of overlaying the Sobel edge maps of multiple samples from the ScaleLearned$_{900}$ and PolyOculus$_{900}$. We show Sobel edges to highlight distinct regions of the scene structure. Locations of the ScaleLearned$_{900}$ samples' edges are more consistent compared to the PolyOculus$_{900}$ samples' edges at multiple locations. Such consistency results in more clarity in the heatmaps, which suggests a reduction in randomness caused by scale ambiguity. Additional visualizations of scene scale entropy are provided in the supplement.

shown in Figure 4.3. From visual inspection, we can see that when there is high uncertainty in the scene scale, such as with PolyOculus$_{900}$, an edge in the observed view moves to a wide range of location positions throughout all the novel view samples. For our *ScaleLearned* model, we can qualitatively see the location of edges has a lower variance, which shows that our *ScaleLearned* model has a more consistent sense of scale.

Visualizations of the mean absolute deviation (MAD) heatmaps obtained during computing SFC in Figure 4.4 also show the *ScaleLearned* model has less per-pixel MAD. These results convey less variation in each pixel's optical flow for the *ScaleLearned* model compared to both baselines, suggesting less variance in the scales of generated samples. Such qualitative results illustrate a reduction in per-scene sample scale entropy, which supports the claim that optimizing scales alongside the generative model parameters helped the model to have a more consistent sense of scale.

To have a more comprehensive evaluation, we use a second setup in which the camera

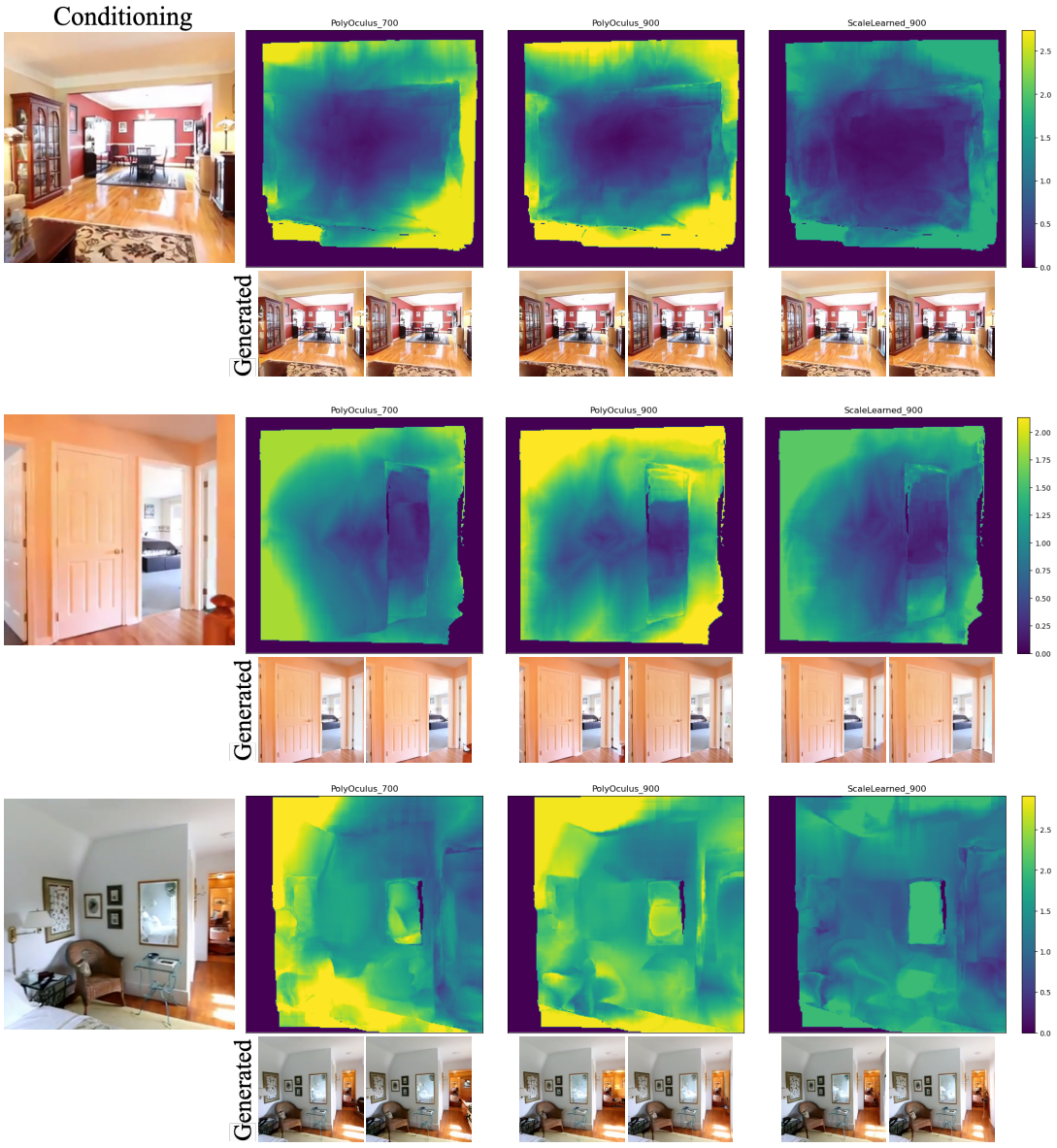


Figure 4.4: Heatmaps of per-pixel optical flow. We present examples of per-pixel optical flow MAD maps. The darker the pixel, the lower the variation in the per-pixel optical flow is in that pixel, showing a more consistent scale among the generated samples. The entropy in scale can be seen by closely comparing the generated frames. In the first row, the visible width of the coffee table and china cabinet varies in PolyOculus$_{700}$ and PolyOculus$_{900}$ samples, whereas those widths are more consistent in the ScaleLearned$_{900}$ samples. It is the same with the red vase, the rightmost wall in the second row, and the door width in the third row.

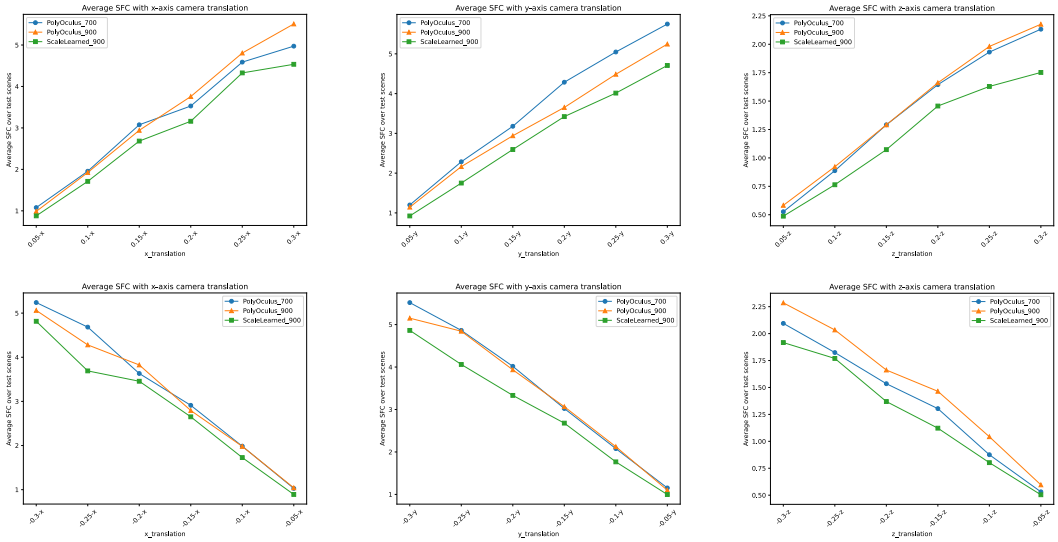


Figure 4.5: SFC results for samples with the camera motion only in one axis. We depict two plots for each axis, one in the positive direction and one in the negative direction. For example, the first column depicts results for translating the camera along the x-axis. The top plot shows SFC values for translating in the positive direction and the bottom one in the negative direction.

motion only consists of translation along one of the axes. We use the same amount of translation we used in the previous experiment. According to plots in Figure 4.5, the *ScaleLearned* model has less sample variance in all cases of this controlled setting as well, showing that not only it has less variation in scale in the ground truth trajectories but also it has a more consistent sense of scale in translations along each axis. It is worth mentioning that RealEstate10k mostly consists of videos with forward or backward motion, which introduces bias to the model. As a result, the amount of SFC is lower for camera motions in the z-axis.

Measuring Scale Inconsistencies (TSED). We evaluate all the methods with TSED [40] to compare the consistency of the generated samples in terms of 3D geometry. The TSED

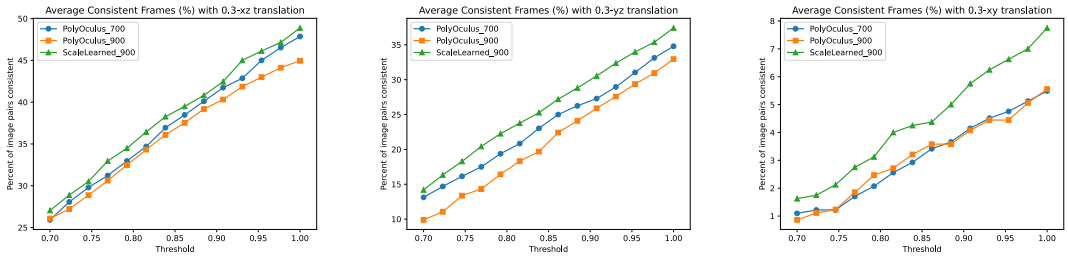


Figure 4.6: Percent consistent image pairs computed with TSED on sample pairs with camera translations in different directions.

metric is not sensitive to motion in the same direction by nature since it uses epipolar lines to measure inconsistency, and epipolar lines do not change by scaling the camera translation. On the other hand, most of the trajectories in the RealEstate10K trace along the same direction for a long time, where TSED is insensitive to the translation scale.

With this in mind, we design this experiment to translate the target camera of the generated samples in different directions to address the scale insensitivity of TSED. We use the samples from the previous experiment with camera translation in different axes. We match the samples from each direction with the samples from a different direction, *i.e.* we match samples generated by translating the camera in the x-axis (Gen_x) to samples generated by translating the camera in the y-axis (Gen_y), and z-axis (Gen_z), producing 100 sample pairs for each scene with each combination of axes (xy, xz, yz). Then, we compute the percentage of consistent pairs among the matched pairs, *i.e.* (Gen_x, Gen_xy) and (Gen_x, Gen_z), based on TSED with different thresholds. The results depicted as plots in Figure 4.6 suggest that the scale learning helped make the samples more consistent in scale in terms of 3D geometry measurements as well.

In the TSED computation, we compute the symmetric epipolar distance (SED) of the matched features in the two frames, which we then threshold. To evaluate the models

SED (Avg/SD) ↓			
Model \ Direction	x-z	y-z	x-y
PolyOculus_{700}	1.35/0.62	1.77/0.71	4.19/2.55
PolyOculus_{900}	1.40/0.67	**1.60**/0.66	4.33/2.48
$\text{ScaleLearned}_{900}$	**1.29/ 0.59**	<u>1.69</u>/**0.61**	**3.89/2.26**

Table 4.2: Average and SD of SED scores used in computing TSED. Lower SED scores show less distance from the epipolar line, showing more 3D consistency.

in terms of 3D consistency, we report statistics on the SED scores in Table 4.2. The ScaleLearned model has less epipolar distance (in pixels) on average compared to the other baselines, supporting the claim that scale learning helps the model become more 3D consistent.

Measuring Image Quality. We evaluate the sample image quality of our method to ensure that the scale learning process does not degrade the quality of the samples. We report results in terms of PSNR and LPIPS to assess the samples' reconstruction quality in Table 4.3. We use the samples generated for the SFC evaluation. The results show that our proposed approach performs well in reconstructing images, conveying that scale learning schema maintains the high quality of samples while increasing scale consistency.

Moreover, we evaluate image quality based on FID. As FID compares distributions of images in a virtually arbitrary feature space, it works better with more images. Hence, we use long-term generations, in which we generate a trajectory of 21 consecutive frames for this comparison. We use Markov and Keyframed sampling setups from Polyoculus to generate samples where Markov sampling uses past frames, and Keyframed sampling uses keyframes for conditioning. Results are shown in Table 4.4. In the Markov sampling, we have the smallest FID in all comparisons, suggesting that by removing scale noise,

LPIPS ↓						
Model	0.05	0.1	0.15	0.2	0.25	0.3
PolyOculus_{700}	0.067	0.092	0.118	0.143	0.164	0.192
PolyOculus_{900}	0.065	0.093	0.119	0.143	0.165	0.190
$\text{ScaleLearned}_{900}$	**0.064**	**0.092**	**0.117**	**0.143**	**0.163**	**0.188**
PSNR ↑						
PolyOculus_{700}	24.6	22.4	20.9	19.8	19.0	**18.4**
PolyOculus_{900}	24.7	22.3	20.8	19.9	19.0	18.3
$\text{ScaleLearned}_{900}$	**24.8**	22.4	**20.9**	**19.9**	**19.1**	<u>18.4</u>

Table 4.3: Reconstruction metrics on samples generated with test set poses, where ground-truth reference frames are available.

	Markov			Keyframed		
Model	All	Last 5	Last 3	All	Last 5	Last 3
PolyOculus_{700}	37.6	53.1	55.2	26.9	30.5	31.2
PolyOculus_{900}	36.9	52.1	54.3	**26.8**	**30.4**	**31.2**
$\text{ScaleLearned}_{900}$	**36.7**	**51.4**	**53.5**	<u>26.9</u>	<u>30.5</u>	<u>31.2</u>

Table 4.4: Reporting FID on twenty-one generated samples using Markov and Keyframed sampling specs. The "All" column corresponds to the average FID over all of twenty-one, the "Last 5" to the average FID over the last five, and the "Last 3" to the average FID over the last three generated frames.

we help the model accumulate less error and improve quality over long-term trajectories. However, in the keyframed setup, this effect is not visible since keyframing reduces error accumulation by nature.

4.5.3 Summary

In this chapter, we presented both quantitative and qualitative results suggesting that learning the scales along other GNVS model parameters lowers the entropy of scale inherited from uncalibrated data by the model. Furthermore, our model maintains high image quality and 3D consistency.

Chapter 5

Conclusion and Future Work

In this thesis, we identified scale ambiguity as a fundamental challenge in generative novel view synthesis (GNVS). Models tasked with synthesizing new views often work with uncalibrated data, which results in inconsistencies in scale during inference. As the models operate on arbitrary scales, the synthesized images lack a consistent scale for the same camera movement. This work highlights how generative models inherit these scale inconsistencies from their training data, emphasizing the need for a more structured approach to addressing this ambiguity.

To address the issue of scale ambiguity, we introduced a novel method where the generative model is trained to learn the scale jointly with other model parameters. By embedding this within a denoising diffusion probabilistic model, we enable the model to correct for the inherent scale mismatches in the input data. This joint learning approach ensures that the model maintains consistency in scale while simultaneously optimizing for other generative tasks. The result is a more robust inference process that aligns better with real-world

scene scales.

Due to the lack of a standardized quantitative metric to measure scale consistency across generated samples of the same camera movement, we introduced the Sample Flow Consistency (SFC) metric. SFC offers a means to quantitatively evaluate how well a generative model maintains scale when synthesizing multiple samples of the same camera pose from the same scene. This metric fills an important gap in the field, as it provides a direct measure of consistency, enabling more objective comparisons between methods and offering a new benchmark for evaluating GNVS models.

Through experimentation and evaluation using the SFC metric, we demonstrated that our approach significantly improves the scale consistency of generated samples over other baselines. We presented both quantitative and qualitative results that highlight these improvements, showing that our method can generate novel views with a more consistent sense of scale. The success of this method validates the effectiveness of joint scale learning and its ability to mitigate one of the key limitations in GNVS.

Our work offers an approach to addressing the scale ambiguity problem in generative models, but it also opens the door for further exploration. The introduction of the SFC metric and our proposed training methodology provide a solid framework upon which future studies can build. We believe that this research not only mitigates the scale ambiguity in GNVS but also sets the stage for more advanced techniques that could push the boundaries of scale consistency and generative view synthesis.

5.1 Limitation and Future Work

While the proposed scale-learning approach demonstrates promising results, especially in learning scene scales effectively, there are several areas where further improvements could be explored. One key limitation of this work is that we tested our scale-learning scheme on a single GNVS method. This choice was primarily influenced by the considerable computational resources and time required for training GNVS models. Given the encouraging results, extending this scheme to other GNVS methods could lead to a more generalizable solution for addressing scale ambiguity. Investigating the adaptability of the proposed approach across diverse models would be an interesting direction for future research, potentially leading to more robust and versatile GNVS techniques.

Another limitation is that our current work does not establish whether the learned scales converge to the actual scene scales. This gap is mainly due to the absence of ground-truth scale information in available datasets. As a result, we could not assess the accuracy of scale convergence quantitatively. A potential avenue for future work could involve experimenting with metric-calibrated datasets where ground-truth scales are available. Introducing controlled perturbations to camera positions in such datasets would allow us to evaluate whether the proposed method can effectively recover accurate camera poses by learning the correct scale factors.

Additionally, during our experiments, we observed that the scale between different types of scenes could be inconsistent. For instance, outdoor scenes captured by drones tend to exhibit significantly larger scales than indoor scenes typically captured by humans. This inconsistency suggests that the current method may not fully adapt to varying scene contexts. Further investigation is required to understand and address this disparity,

possibly by incorporating scene-specific features or conditioning the model on additional contextual information.

Overall, addressing these limitations would significantly enhance the proposed scale-learning approach's robustness and generalization capabilities, paving the way for broader applications in diverse GNVS settings.

5.2 Broader Impact

This work can have a positive societal impact on 3D scene generation applications. An appealing case of this is in autonomous driving, where the accuracy of scene scale plays an important role in downstream tasks such as trajectory prediction and collision avoidance. It can also be useful in real estate-related cases as it can generate tour videos of houses with a few images. From a negative perspective, this work could be used to create realistic fake scenes as it uses a generative model backbone that is capable of generating non-existence scenarios.

Bibliography

[1] Shai Avidan and Amnon Shashua. Novel view synthesis in tensor space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1034–1040, 1997. 15

[2] Andreas Blattmann, Robin Rombach, Kaan Oktay, Jonas Müller, and Björn Ommer. Retrieval-augmented diffusion models. In *Neural Information Processing Systems (NeurIPS)*, 2022. 7, 20, 26

[3] Georges Chahine and Cédric Pradalier. Survey of monocular SLAM algorithms in natural environments. In *CRV*, pages 345–352, 2018. 11

[4] Raja Chatila and Jean-Paul Laumond. Position referencing and consistent world modeling for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 138–145, 1985. 3, 11, 14

[5] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *Proceedings of SIGGRAPH*, pages 279–288, 1993. 15

[6] Sunglok Choi, Jaehyun Park, and Wonpil Yu. Resolving scale ambiguity for monocu-

lar visual odometry. In *International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 604–608, 2013. 11

[7] Antonio Criminisi, Ian Reid, and Andrew Zisserman. Single view metrology. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 434–441, 1999. 12

[8] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12873–12883, 2021. 41

[9] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Neural Information Processing Systems (NeurIPS)*, pages 2672–2680, 2014. 17

[10] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Neural Information Processing Systems (NeurIPS)*, pages 6626–6637, 2017. 33

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Neural Information Processing Systems (NeurIPS)*, 2020. 4, 18, 26, 32, 41

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014. 42

[13] Stéphane Laveau and Olivier D. Faugeras. 3-d scene representation as a collection of images. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 689–691, 1994. 15

[14] Andrew Liu, Ameesh Makadia, Richard Tucker, Noah Snavely, Varun Jampani, and Angjoo Kanazawa. Infinite nature: Perpetual view generation of natural scenes from a single image. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 14438–14447, 2021. 1, 17

[15] H.C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. In *Readings in Computer Vision*, pages 61–62. 1987. 3

[16] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101, 2017. 42

[17] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics (T-RO)*, 31(5): 1147–1163, 2015. 11, 12, 42

[18] David Nistér. An efficient solution to the five-point relative pose problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 195–202, 2003. 3

[19] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023. 16

[20] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotný. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 10881–10891, 2021. 14, 16

[21] Xuanchi Ren and Xiaolong Wang. Look outside the room: Synthesizing a consistent long-term 3d scene video from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3553–3563, 2022. 1, 17, 43

[22] Robin Rombach, Patrick Esser, and Björn Ommer. Geometry-free view synthesis: Transformers and no 3d priors. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 14336–14346, 2021. 1, 17, 43

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pages 234–241, 2015. 20, 32, 41

[24] Kyle Sargent, Zizhang Li, Tanmay Shah, Charles Herrmann, Hong-Xing Yu, Yunzhi Zhang, Eric Ryan Chan, Dmitry Lagun, Li Fei-Fei, Deqing Sun, and Jiajun Wu. Zeronvs: Zero-shot 360-degree view synthesis from a single real image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 6, 12, 15, 18, 21, 23

[25] Saurabh Saxena, Junhwa Hur, Charles Herrmann, Deqing Sun, and David J. Fleet.

Zero-shot metric depth with a field-of-view conditioned diffusion model. *CoRR*, abs/2312.13252, 2023. 24

[26] Davide Scaramuzza and Friedrich Fraundorfer. Visual odometry [tutorial]. *IEEE Robotics Autom. Mag.*, pages 80–92, 2011. 1

[27] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016. 29

[28] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 18

[29] Zachary Teed and Jia Deng. RAFT: recurrent all-pairs field transforms for optical flow. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 12347 of *Lecture Notes in Computer Science*, pages 402–419, 2020. 34

[30] Rui Tian, Yunzhou Zhang, Delong Zhu, Shiwen Liang, Sonya Coleman, and Dermot Kerr. Accurate and robust scale recovery for monocular visual odometry based on plane geometry. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 5296–5302, 2021. 11

[31] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 548–557, 2020. 6, 21, 22

[32] A. Vahdat and K. Kreis. Improving diffusion models as an alternative to gans, part 1, April 2022. URL https://developer.nvidia.com/blog/improving-diffusion-models-as-an-alternative-to-gans-part-1/. 19

[33] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Neural Information Processing Systems (NeurIPS)*, pages 6306–6315, 2017. 17

[34] Xiaolong Wang, Allan Jabri, and Alexei A. Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2566–2576, 2019. 35

[35] Zhongyi Wang, Mengjiao Shen, and Qijun Chen. Eliminating scale ambiguity of unsupervised monocular visual odometry. *Neural Process. Lett.*, 55(7):9743–9764, 2023. 11

[36] Daniel Watson, Saurabh Saxena, Lala Li, Andrea Tagliasacchi, and David J. Fleet. Controlling space and time with diffusion models. *CoRR*, abs/2407.07860, 2024. 16, 21, 24, 33, 34

[37] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7465–7475, 2020. 1, 17

[38] Disai Yang, Haiping Ai, Jiantao Liu, and Bingwei He. Absolute scale estimation for

underwater monocular visual odometry based on 2-d imaging sonar. *Measurement*, 190, 2022. 2

[39] Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Simon Chen, Yifan Liu, and Chunhua Shen. Towards accurate reconstruction of 3d scene shape from A single monocular image. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 45(5):6480–6494, 2023. 16

[40] Jason J. Yu, Fereshteh Forghani, Konstantinos G. Derpanis, and Marcus A. Brubaker. Long-term photometric consistent novel view synthesis with diffusion models. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 7071–7081, 2023. 1, 17, 33, 43, 52

[41] Jason J. Yu, Tristan Aumentado-Armstrong, Fereshteh Forghani, Konstantinos G. Derpanis, and Marcus A. Brubaker. Polyoculus: Simultaneous multi-view image-based novel view synthesis. *CoRR*, abs/2402.17986, 2024. 1, 4, 7, 17, 25, 32, 41, 43, 46, 49

[42] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM Transactions on Graphics (TOG)*, 37(4):65, 2018. 3, 12, 13, 16, 22, 29, 42, 45