

Zero-Shot Novel View and Depth Synthesis with Multi-View Geometric Diffusion — Supplementary Material —

Vitor Guizilini¹ Muhammad Zubair Irshad¹ Dian Chen¹
Greg Shakhnarovich² Rares Ambrus¹

Toyota Research Institute (TRI)¹ Toyota Technological Institute at Chicago (TTIC)²

A. Datasets Preparation

Each dataset was downloaded from its official source, following standard procedures. Whenever possible, we used officially provided PyTorch dataloaders to access the relevant information (i.e., images, intrinsics, extrinsics, and depth maps), and wrote our own if not available. In all cases, data was mapped into a common format to enable mixed-batch training across all data sources. This includes padding images and intrinsics of different resolutions, and using empty depth maps whenever this label is not available. We treat video and multi-view datasets equally, by positioning all available cameras from the same scene on a global frame of reference, and considering all possible pairs as a potential source of training data. To select valid pairs for our purposes, we utilized three criteria, described below:

Camera center distance. Conditioning views must have a camera center distance within \mathbf{t}_c^n within $c_{min} < \|\mathbf{t}_c^n - \mathbf{t}_t\| < c_{max}$, where \mathbf{t} is the camera’s translation vector. In the case of dynamic datasets, we also apply the same constraint in a temporal sense to mitigate the impact of moving objects, such that $t_{min} < \|t_c^n - t_t\| < t_{max}$, where t is the timestep of each camera within the sequence (fractional timesteps are used in the case of datasets with multiple asynchronized cameras). Assuming c_M to be the maximum distance across any two cameras in a sequence, we set $c_{min} = 0.05 c_M$ and $c_{max} = 0.2 c_M$, and $t_{min} = -8$ and $t_{max} = 8$.

Camera viewpoint angle. Conditioning views must have a viewing direction with cosine similarity within $\alpha_{min} < \cos^{-1} \frac{\mathbf{v}_c^n \cdot \mathbf{v}_t}{\|\mathbf{v}_c^n\| \|\mathbf{v}_t\|} < \alpha_{max}$, where $\mathbf{v} = \mathbf{R}^{-1} \times [0, 0, 1]^T$ is a vector pointing forward (positive z) relative to a world-to-camera rotation matrix \mathbf{R} . In all experiments, we set $\alpha_{min} = 0$ and $\alpha_{max} = \pi/2$ for depth generation, to avoid supervision from sparse reconstructions, and $\alpha_{min} = 0$ and $\alpha_{max} = \pi$ for image generation, to promote extrapolation to novel viewpoints.

Pointcloud overlapping. Whenever depth maps are available, we set a threshold p_{min} on the percentage of how many valid pixels of each conditioning view are projected

onto the target view. For each pixel $\mathbf{p}_c^n = \{u, v\}$ with depth d from a conditioning view \mathbf{I}_c^n with depth \mathbf{D}_c^n , we can obtain its projection \mathbf{p}_t' and depth d_t' onto the target view via:

$$\begin{bmatrix} u' \\ v' \\ d' \\ 1 \end{bmatrix}_t = \left(\tilde{\mathbf{K}}_c \mathbf{T}_c^n \right)^{-1} \begin{bmatrix} u \\ v \\ d \\ 1 \end{bmatrix}_c \left(\mathbf{T}_t \tilde{\mathbf{K}}_t \right) \quad (1)$$

where $\tilde{\mathbf{K}} = \begin{bmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 4}$ is a homogeneous intrinsics matrix. A projected point is considered valid if $u' \in [0, H]$ and $v' \in [0, W]$, i.e. if its projected coordinates are inside the target view. We set $p_{min} = 30\%$, and additionally discard samples with less than 64 valid projected pixels.

These criteria were used as a pre-processing step, to generate a list of valid training samples. We will open-source dataloaders for all training and validation datasets, to facilitate the reproduction of our work, as well as the list of valid samples used in our experiments. We use Webdataset [1] to optimize storage and training efficiency.

B. Additional Qualitative Examples

In Figure 1 we provide additional qualitative results of MVGD in different evaluation benchmarks, as well as in-the-wild images from different sources (complementing Figure 3 of the main paper). Conditioning images are shown in the top left, with corresponding cameras (denoted by different colors) positioned relative to the target camera (denote by black). On the bottom, from left to right, we show: ground-truth image, predicted novel image, and predicted novel depth, all from the target viewpoint. We emphasize that novel images and depth maps are generated *directly as an output of our diffusion model*, rather than rendered from a 3D neural field or set of 3D Gaussians.

To highlight the multi-view consistency of MVGD, in Figure 2 we show qualitative results obtained using the same conditioning views to generate multiple predictions

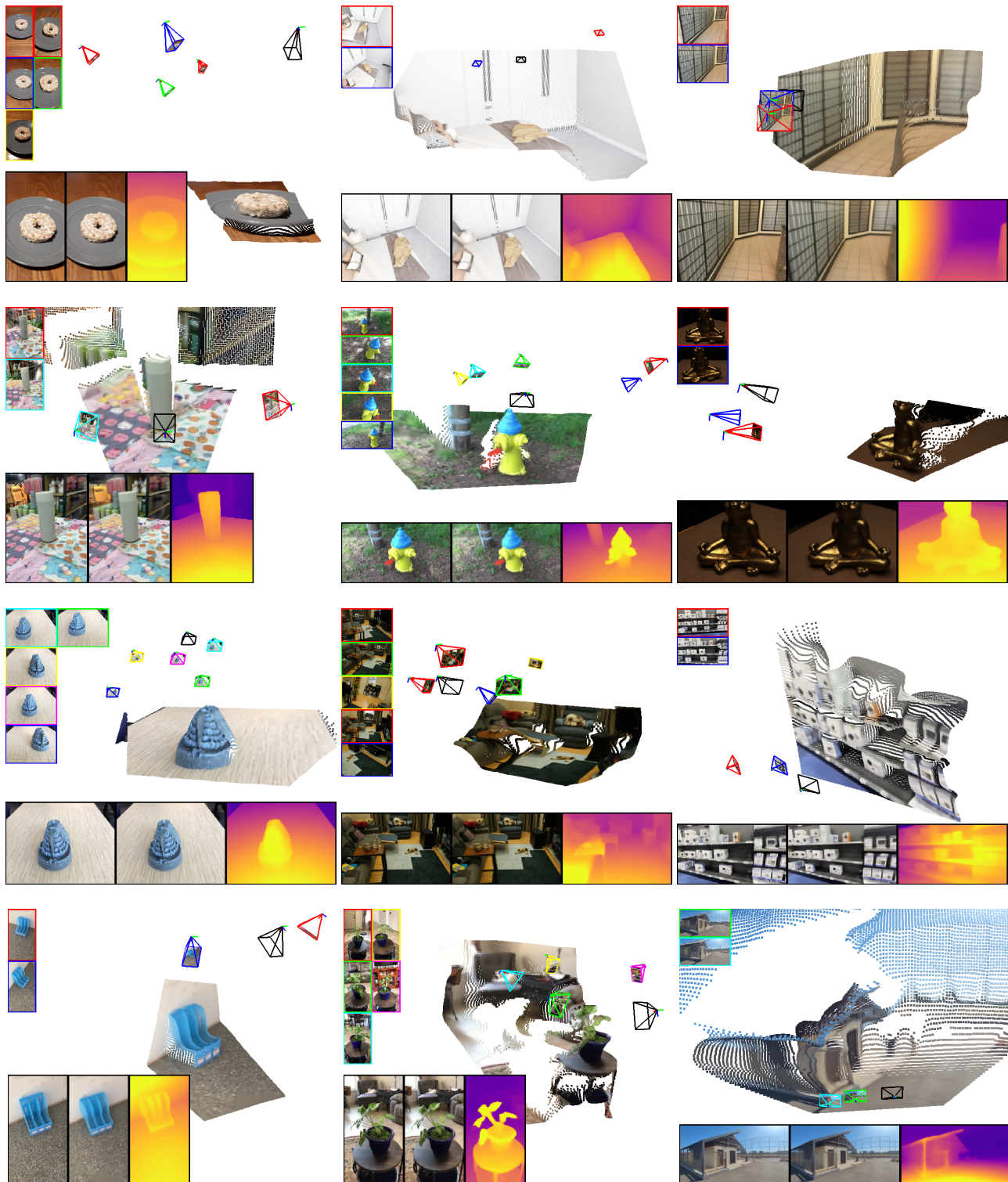


Figure 1. **Zero-Shot MVGD novel view and depth synthesis results** randomly sampled from different evaluation benchmarks and in-the-wild datasets. Top left images are conditioning views (colored cameras), and bottom images are the target view (black camera), showing from left-to-right: ground-truth image, predicted image, and predicted depth map. These predictions are used to produce a colored 3D pointcloud observed from the target viewpoint.



Figure 2. **Accumulated MVGD pointclouds**, obtained by generating novel images and depth maps from various viewpoints (black cameras), using the same conditioning views (colored cameras), and stacking them together without any post-processing. Our zero-shot architecture is capable of directly generating multi-view consistent predictions that match the scale from conditioning cameras.

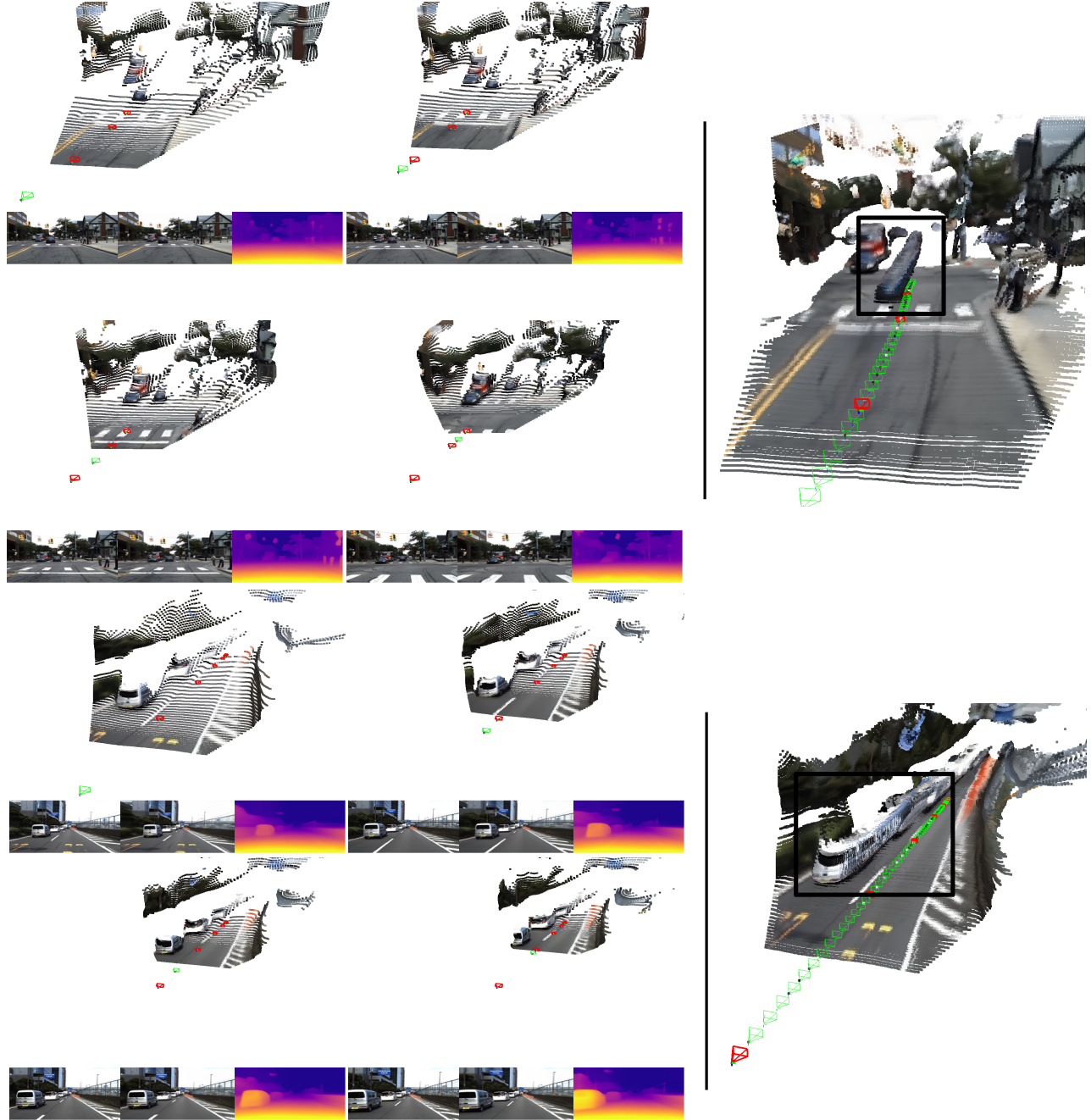


Figure 3. **Accumulated MVGD pointclouds on a dynamic dataset [4].** (left) Red cameras are used as conditioning views, and novel images and depth maps are generated from green cameras. (right) Colored pointclouds are calculated from these predictions and stacked together without any post-processing. Even though MVGD does not explicitly model dynamic objects, it implicitly learns how the scene should change when interpolating between views with objects in different locations (e.g., moving cars), while keeping the remainder static.

from novel viewpoints, and stacking the predicted colored pointclouds together without any post-processing. Each prediction is generated independently, by setting the novel viewpoint as the origin and positioning the conditioning views relative to it. Even so, they yielded highly consistent pointclouds, both in terms of appearance as well as

reconstructed 3D geometry. We attribute this consistency (and ablate it in Table 5 of the main paper) to our proposed scene scale normalization (SSN) procedure, that promotes the generation of depth maps that share the same scale as the one provided by conditioning cameras, even in very different settings (e.g., driving, indoors, object-centric).

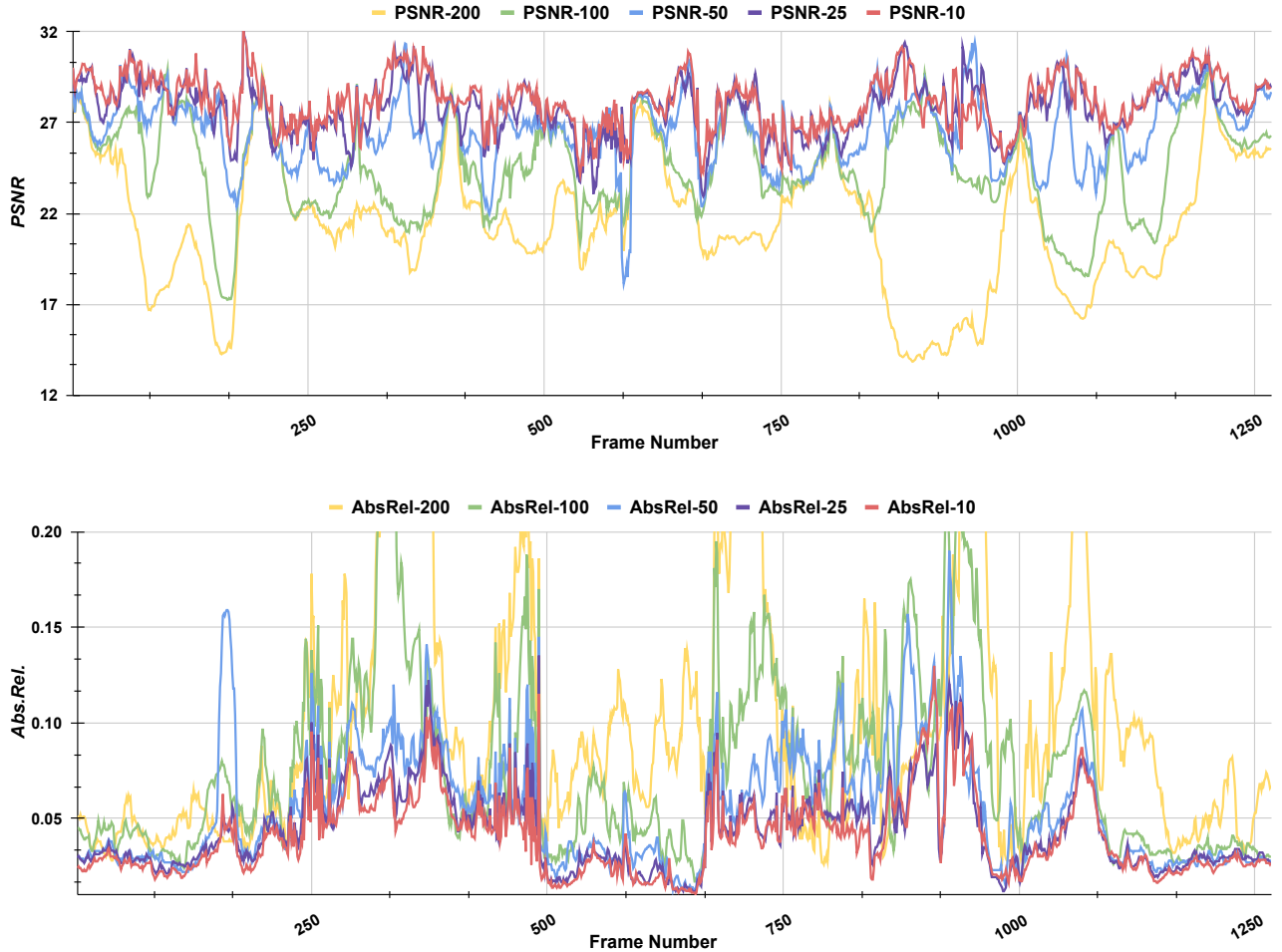


Figure 4. **MVGd per-frame novel view and depth synthesis results** using different numbers of fixed conditioning views, on ScanNet (scene 0086_02, with 1267 images). The same model from all experiments reported in the main paper was used. Legend numbers indicate the stride (i.e., how many target images are positioned between each conditioning view). As expected, results consistently improve as more input information is available, eventually plateauing at around 100 conditioning views.

C. Implicit Dynamics Modeling

Although MVGD does not explicitly model dynamic objects, we elected to include datasets with such behavior to increase the diversity of our training data, and report non-trivial improvements relative to a baseline that only considers static datasets (Table 8 of the main paper). We attribute this behavior to a learned robustness to the presence of dynamic objects [6], similar to other methods that rely on self-supervised multi-view consistency with a static environment assumption [3–5].

However, upon further inspection we observed some degree of implicit motion understanding in our learned representation. Examples are shown in Figure 3, using the DDAD [4] dataset. In those examples, every 10th frame from a 100-frame sequence was used as conditioning, and

remaining cameras were used to generate novel images and depth maps. As we can observe, moving cars are correctly rendered in different locations to ensure a smooth transition between frames, while static portions of the environment are rendered in the same location, taking into consideration only camera motion.

D. Incremental Conditioning

Here we explore how MVGD scales in terms of the number of conditioning views. Due to the use of latent tokens, computational complexity is largely independent of the number of input tokens, which enables (a) pixel-level diffusion without the need for dedicated auto-encoders; and (b) the simultaneous use of more conditioning views. In fact, one target 256×256 image generates 65536 prediction tokens, while each conditioning views adds only 4096 scene tokens,

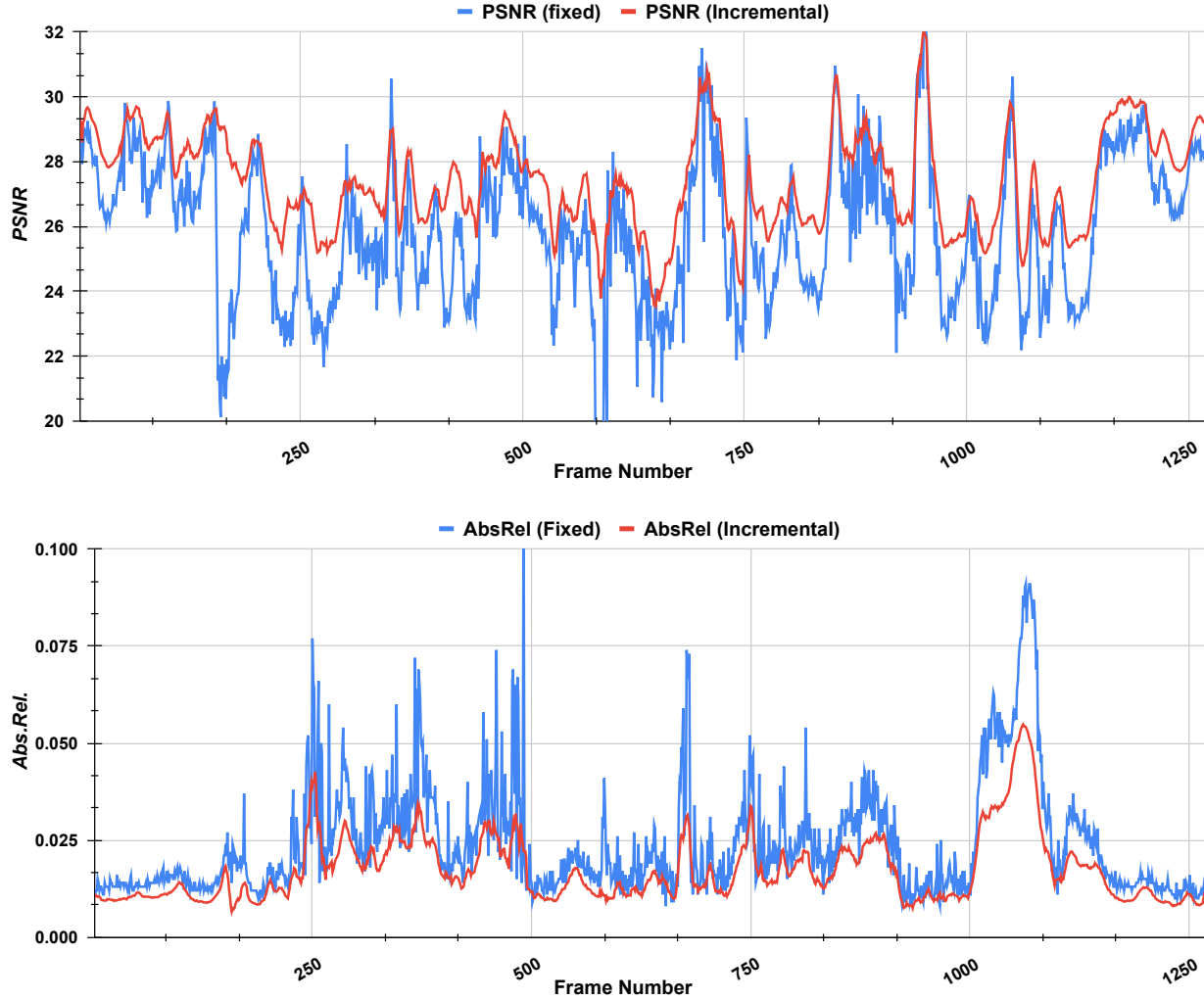


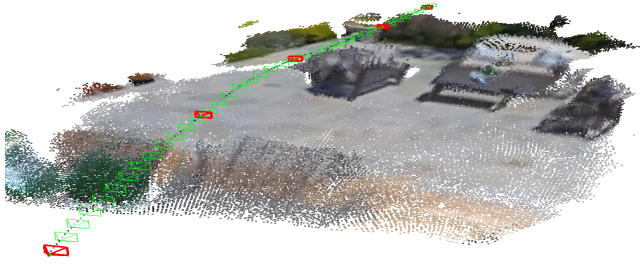
Figure 5. **MVGd per-frame novel view and depth synthesis results** with and without our proposed incremental conditioning strategy, on ScanNet (scene 0086_02, with 1267 images). The same model from all experiments reported in the main paper was used. The blue line indicates a fixed number (25) of conditioning views, evenly spaced with a stride of 50. The red line indicates our proposed incremental conditioning strategy, in which each new generation uses previously generated images as additional conditioning. This strategy leads to consistently better and more stable results in novel view and depth synthesis, especially in regions further away from conditioning views.

since image features are produced at $1/4$ the original resolution. In contrast, our largest model has 2048 latent tokens, which is only 3% of the number of prediction tokens.

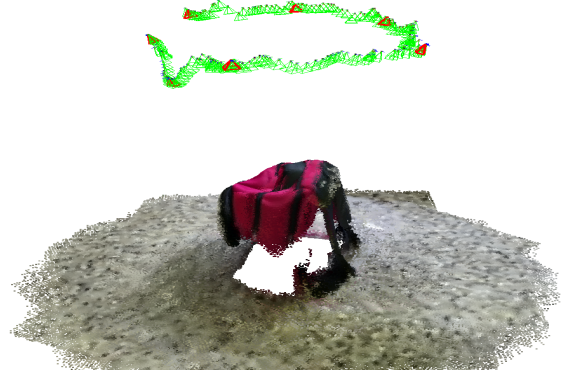
In Figure 4 we show the impact of using more conditioning views over a 1267-frame ScanNet sequence, in terms of novel view (PSNR) and depth (AbsRel) synthesis. We take every N -th frame as conditioning views (given by the legend number), and generate predictions for all remaining frames, using the same model from all experiments reported in the main paper. As expected, results degrade in areas further away from available views, and consistently improve as more conditioning views are provided, eventually plateauing at around 100 (stride 20). Interestingly, independent

experiments using subsets of the sequence (5 subsets of 250 frames) yielded worse results, as evidence that large-scale conditioning (i) does not degrade local performance; and (ii) provides better global context for local predictions.

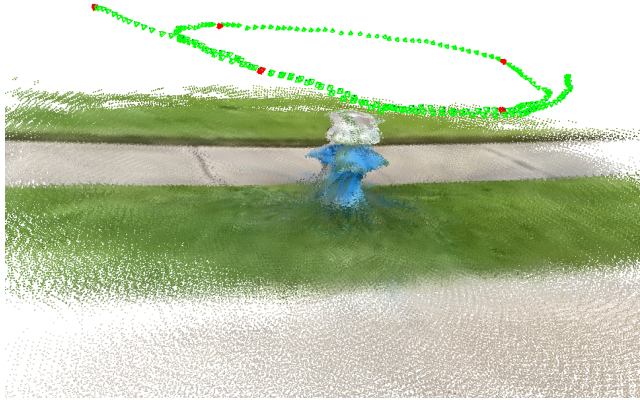
As mentioned in Section 3.6 of the main paper, we also take advantage of this highly efficient architecture to investigate how images generated from novel viewpoints can be added as additional conditioning views, thus increasing the amount of available information for future generations. This incremental conditioning strategy should further improve multi-view consistency in cases where model stochasticity becomes relevant, since each novel view is generated independently and thus might come from different parts of



(a) 149 frames, 5 initial conditioning views.



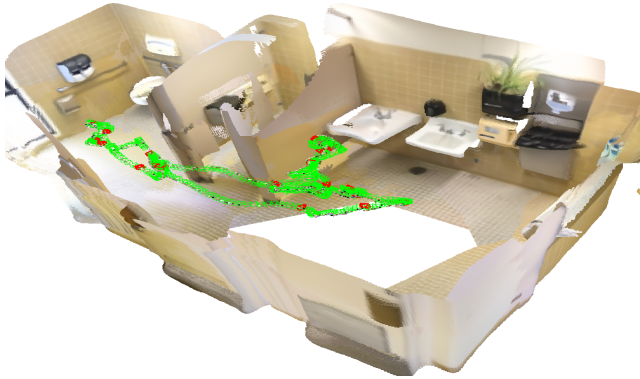
(b) 202 frames, 7 initial conditioning views.



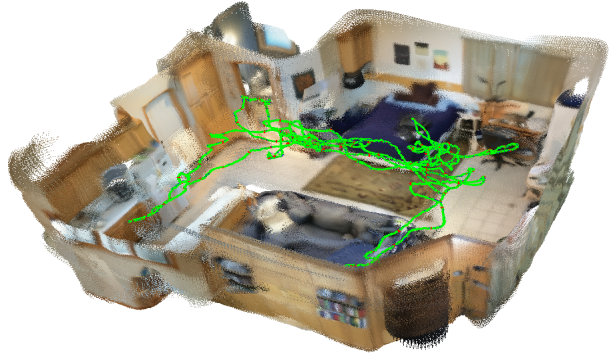
(c) 252 frames, 5 initial conditioning views.



(d) 337 frames, 6 initial conditioning views.



(e) 1268 frames, 25 initial conditioning views.



(f) 5578 frames, 60 initial conditioning views.

Figure 6. **MVGD novel view and depth synthesis results** using our proposed incremental conditioning strategy. Red cameras indicate initial conditioning views, used to generate predictions for green cameras (ordered from closest to furthest away from the initial conditioning views). After each generation, the predicted image is added to the set of conditioning views for future generations. Even though MVGD was trained using only 2 – 5 conditioning views, it can directly scale to thousands on a single GPU.

the underlying distribution, especially in unobserved areas. Figure 5 provides a quantitative evaluation of our proposed incremental conditioning strategy in terms of novel view (PSNR) and depth (AbsRel) synthesis, compared to the use of a fixed number of conditioning views. As we can observe, the introduction of additional conditioning from gen-

erated views consistently improves generation quality,

In Figure 6 we qualitatively show incremental conditioning results on different sequences. Red cameras serve as initial conditioning, and novel images and depth maps are generated from green cameras. After each generation, the predicted image is used as additional conditioning. Since

generation order matters in this setting, each new generation is performed on the green camera closest to the initial set of conditioning cameras, that still has not been processed. Note that *all* previously generated views are used as additional conditioning, which in some scenarios could lead to thousands of images. Even so, we were able to generate novel predictions on a single A100 GPU with 40GB. In terms of inference speed, generations with 25 conditioning views in this setting take 0.5s, and generations with 1250 ($50\times$) conditioning views take around 20s ($40\times$). Additional heuristics, such as using only generated views close to the target view, should lead to increased efficiency while still improving generation quality.

E. Limitations

A limitation of our proposed Scene Scale Normalization (SSN) procedure is its inability to simultaneously generate predictions from multiple viewpoints, since the target camera is always assumed to be at the origin. In Section D we describe an incremental conditioning strategy that mitigates stochasticity when generating predictions from unobserved regions, leading to multi-view consistency over very long sequences (2000+ frames). Another current limitation of MVGD is the lack of dynamics modeling. In Section C we show some evidence of implicit modeling of moving objects, however the proper handling of dynamic scenes (e.g., via temporal embeddings and motion tokens, such as [2]) could lead to improvements and spatio-temporal control over novel view and depth synthesis. Moreover, we believe the lack of large-scale dynamic datasets with accurate camera information still constitutes a challenge for the generation of such spatio-temporal implicit foundation model.

References

- [1] Webdataset. <https://github.com/webdataset/webdataset>, 2024. 1
- [2] Anonymous. STORM: Spatio-temporal reconstruction model for large-scale outdoor scenes. In *Submitted to The Thirteenth International Conference on Learning Representations*, 2024. under review. 8
- [3] Clément Godard, Oisín Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth prediction. In *ICCV*, 2019. 5
- [4] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *CVPR*, 2020. 4, 5
- [5] Chang Shu, Kun Yu, Zhixiang Duan, and Kuiyuan Yang. Feature-metric loss for self-supervised learning of depth and egomotion. In *ECCV*, 2020. 5
- [6] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arxiv:2410.03825*, 2024. 5