MOSAIC: Generating Consistent, Privacy-Preserving Scenes from Multiple Depth Views in Multi-Room Environments (Supplementary Material)

Zhixuan Liu[1] Haokun Zhu[1] Rui Chen[1] Jonathan Francis[1,2]
Soonmin Hwang[3] Ji Zhang[1] Jean Oh[1]
[1]Carnegie Mellon University [2]Bosch Center for AI [3]Hanyang University

1. Overview

In this supplementary material, more details about the proposed MOSAIC and more experimental results are provided, including:

- Discusstion about multi-channel inference-time optimization. (Sec. 2);
- Implementation details of our viewpoint selection algorithm. (Sec. 3);
- Details of dataset collection process. (Sec. 4);
- More qualitative results. (Sec. 5);

2. Multi-channel Inference-time Optimization

To effectively achieve the objective mentioned in main paper in Eq. 16, we incorporate our optimization directly into the diffusion generation process and propose a multi-channel test-time optimization approach. At each timestep t during the sampling process, we optimize the latent variables $z_t^{[1:N]}$ to minimize the depth-weighted projection loss while maintaining high-quality generation. Specifically, for each timestep t, we first obtain the predicted clean latents $\hat{z}_0^{[i]} = f_\theta^{(t)}(z_t^{[i]})$ for each view i. We then calculate the depth-weighted projection loss $L_{\text{Proj}}^{\text{DW}}(z_t^{[1:N]})$ and optimize $z_t^{[1:N]}$ through stochastic gradient descent:

$$z_t^{[i]} \leftarrow z_t^{[i]} - \eta \cdot \frac{\partial L_{\text{Proj}}^{\text{DW}}(z_t^{[1:N]})}{\partial z_t^{[i]}},$$

where η is the learning rate. Ideally, this gradient computation would involve backpropagation through the noise predictor $f_\theta^{(t)}$. However, this approach would incur prohibitive memory costs, especially when dealing with multiple views simultaneously. To address this challenge, we employ a gradient stopping technique where we detach the noise predictor from the computational graph during optimization. Since finding the optimal solution with a single iteration is challenging, we perform multiple optimization steps (n_{opt}) at each timestep during the early denoising stages when the latent structure is still being formed.

3. Viewpoint Selection Algorithm

Our viewpoint selection algorithm is based on the following key assumption: If we discretize the 3D space into voxels, the primary factor affecting consistency between images is the set of voxels occupied by the object, while free voxels do not influence our task.

Thus, our algorithm first identifies all occupied voxels observed in the scene and designates them as **interested points**, assigning each interested point an initial score of 2. Next, for all candidate viewpoints along the trajectory, we compute the total score of the visible interested points based on the camera's intrinsic and extrinsic parameters. We then iteratively select the viewpoint with the highest score, decrement the score of each interested point observed by this viewpoint by 1, and remove the selected viewpoint from the candidate set. This process continues until the accumulated score of the observed interested points reaches a predefined termination threshold.

The pseudocode is shown in Algorithm 1. We also visualize this process in Fig. 1.

4. Dataset Collection Details

For MP3D [1] scenes, we manually collect the trajectories for our dataset.

For HM3D [2] scenes, we begin by randomly selecting a start position for the robot within the scene. Next, we choose a target position such that its geometric distance from the start position falls within the range of $8.5m$ to $11m$. The robot then navigates to the target by following the shortest geometric path. Its action space consists of three discrete actions: moving forward by $0.25m$, turning right by $30°$, and turning left by $30°$. We show the detailed dataset information in Tab. 1. We also show some trajectory examples in Fig. 2.

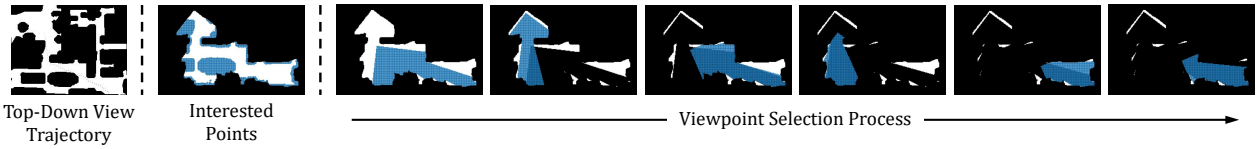

Figure 1. Visualization of the viewpoint selection process.

Algorithm 1 Viewpoint Selection Algorithm

Input: A set of candidate viewpoints V, depth observations $Depth$.
Output: A set of selected viewpoints.
Compute Interested Points: Extract the set of interested points I using $Depth$ and V.
Initialize: Assign an initial score of 2 to each point in I and set `Score_sum` = 0.
while `Score_sum` $< Threshold$ **do**
    for all $v \in V$ **do**
        Compute scores for all visible interested points.
    end for
    Select the candidate viewpoint v^* with the highest score.
    for all $v \in V$ **do**
        Remove viewpoints that exhibit highly similar visibility to v^*.
    end for
    Update scores for observed interested points.
    `Score_sum` $\leftarrow$ `Score_sum` + Score(v^*).
end while

After collecting the data in the scenes, we run the viewpoint selection algorithm introduced in Sec. 3 to get our final dataset used for test.

5. More Qualitative Results

We show more qualitative results in Figs. 3 to 7.

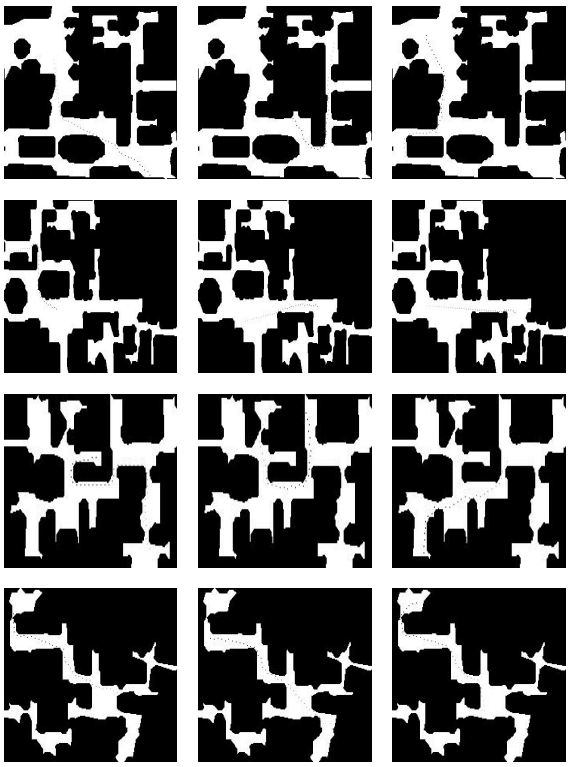

Figure 2. Visualization of trajectories in our collected dataset.

References

[1] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments, 2017. 1

[2] Santhosh K. Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X. Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai, 2021. 1

Table 1. Summary of HM3D Dataset Information

Episode Idx	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Total Step Mean	55.90	48.63	55.90	48.50	49.89	58.00	58.13	38.50	54.30	39.30	59.80	44.50	49.10	57.20	40.00	54.40
Geo Distance Mean	9.72	9.33	8.99	9.21	9.10	9.68	9.40	7.44	9.81	6.34	9.69	7.04	8.89	9.49	7.17	9.09

Figure 3. More Qualitative Results.

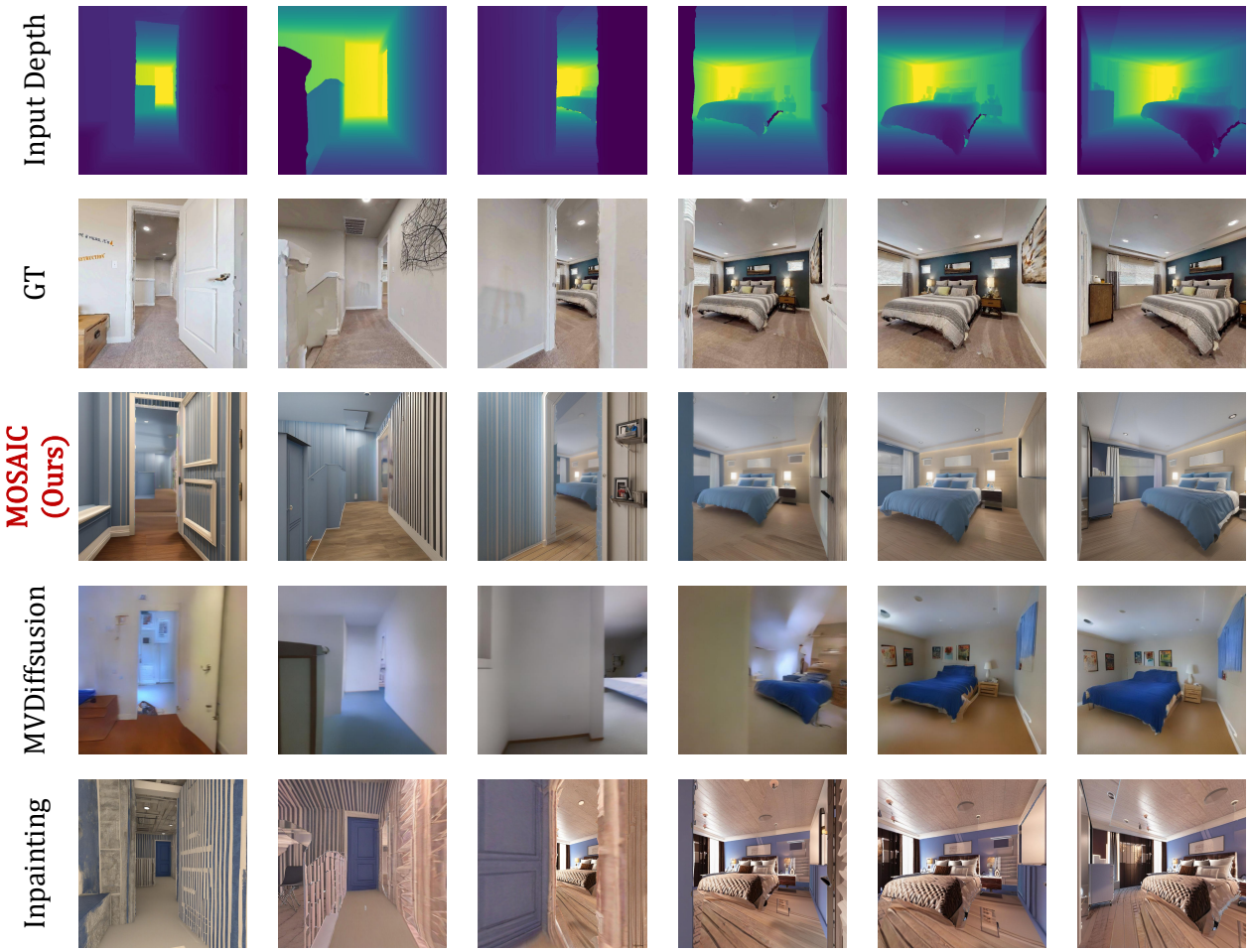

Figure 4. More Qualitative Results.

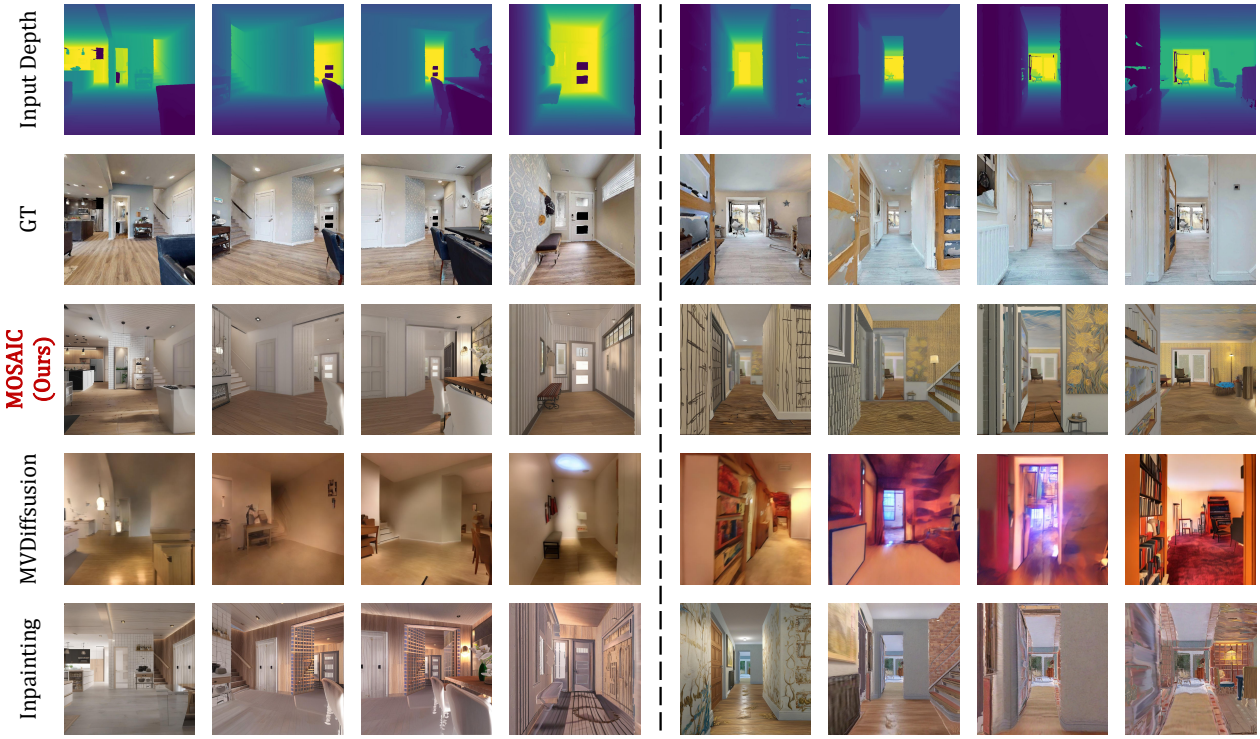

Figure 5. More Qualitative Results.

Figure 6. More Qualitative Results.

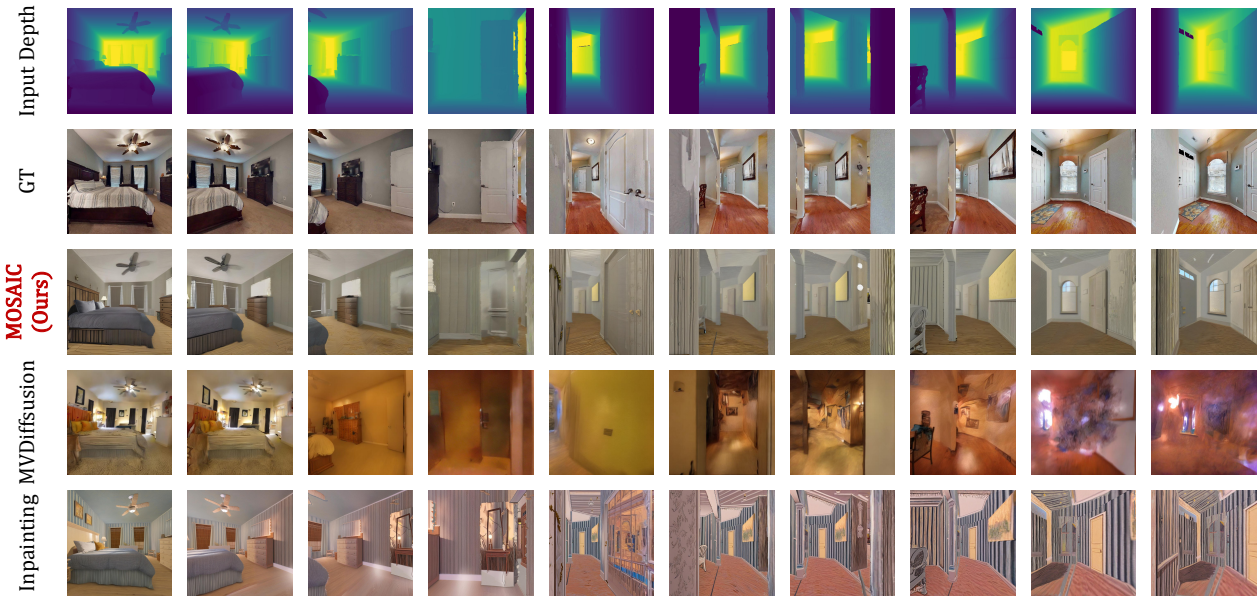

Figure 7. More Qualitative Results.