

Masks make discriminative models great again!

Tianshi Cao¹ * Marie-Julie Rakotosaona² Ben Poole² Federico Tombari² Michael Niemeyer²
¹ University of Toronto ² Google

Abstract

We present *Image2GS*, a novel approach that addresses the challenging problem of reconstructing photorealistic 3D scenes from a single image by focusing specifically on the image-to-3D lifting component of the reconstruction process. By decoupling the lifting problem (converting an image to a 3D model representing what is visible) from the completion problem (hallucinating content not present in the input), we create a more deterministic task suitable for discriminative models. Our method employs visibility masks derived from optimized 3D Gaussian splats to exclude areas not visible from the source view during training. This masked training strategy significantly improves reconstruction quality in visible regions compared to strong baselines. Notably, despite being trained only on masked regions, *Image2GS* remains competitive with state-of-the-art discriminative models trained on full target images when evaluated on complete scenes. Our findings highlight the fundamental struggle discriminative models face when fitting unseen regions and demonstrate the advantages of addressing image-to-3D lifting as a distinct problem with specialized techniques.

1. Introduction

Reconstructing photorealistic 3D scenes from a single image is a difficult but valuable problem that has sought many solutions. The challenge of 3D scene reconstruction can be dissected into three layers: image structure prediction (i.e. coarse 3D information such as depth maps or point clouds), image-to-3D lifting (i.e. a viewable representation from novel viewpoints), and 3D completion (i.e. filling unseen areas). This paper focuses specifically on image-to-3D lifting, which we define as the conversion of image to a 3D model that encompasses what is visible in the image. Studying this separately from 3D completion offers several advantages. First, it avoids the hallucination problem inherent in completion tasks which requires the model to in-paint unseen regions. Secondly, 3D lifting requires only sparse

multi-image data for training, whereas 3D reconstruction generally requires data of scenes that are more complete.

Recent works [7, 10] have used 3D structure prediction pre-training to improve 3D appearance modeling generalization, but they have not made the distinction between lifting and completion. We observe that these are fundamentally different problems: lifting is discriminative with definable correct predictions, while completion is generative with a distribution of outcomes. Forcing discriminative models into generative tasks causes the model to predict “mean”-answers that minimize loss across outcome distributions. Translated to the image-to-3D task, this problem materializes as blurriness in the outputs. In the most extreme single image to 3D scene setting, this problem is further exacerbated due to the limited information provided by the single image of unseen and occluded regions in the 3D scene.

However, discriminative models offer several advantages over generative models. Generative approaches (e.g., diffusion [6, 8] or auto-regressive [9, 13]) require multiple network passes for outputs, increasing compute costs and slow inference. Another consequence is that indirect loss functions, such as LPIPS [19] computed on rendered images of 3D outputs, are difficult to apply in the optimization of auto-regressive or diffusion models. As a result, there is significant interest in retaining the discriminative model framework when approaching the image-to-3D problem.

In this paper, we present *Image2GS*, an image to 3D Gaussian Splats [4] model capable of high-fidelity image-to-3D lifting (see Fig. 1). Our hypothesis is that by separating the 3D lifting problem from the 3D completion problem, we can make the problem more well-defined, and hence suitable for a discriminative model. We convert 3D completion into the 3D lifting problem using visibility masks to remove areas in novel view images that are not visible under the source view from loss computation. Since real world 3D scenes have no known 3D ground truth, we optimize per-scene 3D Gaussians to obtain these visibility masks. We show that this masked training strategy significantly improves reconstruction quality in visible regions compared to baselines. Furthermore, we find our method trained only on masked regions to be competitive with SotA discriminative

*Work done during internship at Google



Figure 1. In this paper, we focus on the task of image-to-3D lifting - from a single input image, we reconstruct partial 3D scenes that can be rendered from novel views. Our proposed method achieves

models trained on full target images when evaluated also on full target images, illustrating that discriminative models struggle to meaningfully fit unseen regions.

2. Related Works and Background

Feed-forward 3D Reconstruction Feed-forward reconstruction approaches generate 3D reconstructions directly from a single or a few images through neural networks. MINE [5] and SV-MPI [14] predict multi-plane images, while [16] uses a neural field representation. More recently, LGM [12] and Splatter Image [11] directly predict 3D Gaussian Splat from input images, but are designed for object reconstruction with no background. More related to this paper are works that directly predict pixel-aligned Gaussians, such as pixelSplat [1], MVSSplat [2], latentSplat [15], and Flash3D [10]. The last of which also focuses on the single image input setting and leverages depth prediction pre-training.

3D Gaussian Splat as Scene Representation 3D Gaussian Splat [4] offer an efficient and high-quality representation for 3D scene reconstruction by modeling the scene as a collection of 3D Gaussian primitives. Each Gaussian primitive is defined by its position $\mu \in \mathbb{R}^3$, covariance matrix $\Sigma \in \mathbb{R}^{3 \times 3}$ (which determines the shape and orientation), and appearance attributes including a view-dependent color $c(v) \in \mathbb{R}^3$ and opacity $\alpha \in [0, 1]$. The unnormalized density function of each 3D Gaussian is given by $g(x) = \exp(-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu))$. During rendering, these 3D Gaussians are projected onto the image plane, resulting in 2D Gaussians with covariance $\Sigma' = JW\Sigma W^T J^T$, where J is the Jacobian of the projection and W is the camera matrix. The opacity of each Gaussian at each pixel is computed as $\alpha' = \alpha_i g_i(x)$. The final color at each pixel is computed through alpha compositing: $C = \sum_{i=1}^n \alpha'_i c_i \prod_{j=1}^{i-1} (1 - \alpha'_j)$, where Gaussians are sorted front-to-back. This representation enables efficient differentiable rendering and optimization

of scene parameters through gradient-based methods, making it particularly suitable for novel view synthesis and dynamic scene reconstruction.

3. Method

In the image-to-3D lifting task, the goal is to reconstruct a 3D model of visible regions in the input view. Image2GS is an image-to-3D lifting model that takes an image as input and outputs a 3D Gaussian splat scene representing visible areas in the input image. The Image2GS model is a ViT[3]-based architecture that predicts Gaussian splat attributes per pixel from the input image. Similar to other image-to-Gaussian splat methods, we train this model on paired input-view, target view data by supervising the rendered appearance of the Gaussian splats in target views. Unlike previous works, we specialize the model for the image-to-3D lifting task by masking unseen regions in the target view. This requires knowing the 3D scene geometry to establish visibility between the input and target view, which is not available in real-world multiview datasets. To overcome this problem, we preprocess each scene in the dataset by reconstructing per-scene 3D Gaussian splats, which enables the computation of view-to-view visibility masks that is consistent with a faithful 3D reconstruction of the scene. In this section, we first detail the model architecture of Image2GS, followed by details of how we preprocess the dataset to create per-scene Gaussian splats, and lastly describe how the training protocol of Image2GS is tailored to the image-to-3D lifting task.

3.1. Model

Given an RGB input image $x \in \mathbb{R}^{3 \times H \times W}$ of a scene, the model f predicts a set of 3D Gaussians that represents the 3D scene, $G = f(x)$. We parameterize the predicted Gaussians as $G = \{(\mu_i, \alpha_i, \theta_i, s_i, c_i)\}_{i=1}^{k \times H \times W}$, where $\mu_i \in \mathbb{R}^3$ is the center of the Gaussian in \mathbb{R}^3 , $\alpha_i \in [0, 1]$ is the opacity, $s_i \in \mathbb{R}^3$ is the scales of the Gaussians, $\theta_i \in SO(3)$ is the rotation of the Gaussian represented as a quaternion (\mathbb{R}^4), and c_i is a set of spherical harmonics coefficients representing directional varying color of the gaussian. In our model, we constrain the prediction to k Gaussians per pixel, and in our experiments we set $k = 1$.

To transfer knowledge from large-scale pretraining, we base our architecture on DepthAnythingV2 (DAV2) [17] and parameterize our model f as $f = d_w(l_\phi(x))$, comprising a decoder head d_w built on top of a ViT backbone l_ϕ . Following prior work, we assume a pinhole camera model for the input image with known focal length, which allows us to unproject depth maps predicted by DAV2's decoder head into a 3D point cloud. We extend the depth decoder head of DAV2 by adding additional channels to its final convolution layer for predicting $\{\delta_i, \alpha_i, \theta_i, s_i, c_i\}_{i=1}^{H \times W}$. Specifically, $\delta_i \in \mathbb{R}^3$ is added to the point cloud obtained



Figure 2. Novel view renders of Image2GS outputs, trained with vs without masking.

Table 1. Comparing Image2GS trained with masking versus without masking on RealEstate10K under different target view settings.

Setting	Model	Input frame			5 frames			10 frames			$\mathcal{U}[-30, 30]$ frames		
		PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
3D Completion	No mask	32.72	0.943	0.095	26.96	0.855	0.209	24.40	0.813	0.277	24.50	0.802	0.225
	With mask	42.57	0.993	0.005	28.02	0.878	0.061	24.57	0.820	0.105	25.30	0.827	0.112
3D Lifting	No mask	32.72	0.943	-	33.60	0.919	-	31.35	0.895	-	32.20	0.902	-
	With mask	42.57	0.993	-	34.84	0.935	-	31.95	0.908	-	33.05	0.915	-

Table 2. Image2GS shows state-of-the-art in-domain performance on RealEstate10k on small, medium and large baseline ranges. Performance of prior methods are taken from [10].

Model	5 frames			10 frames			$\mathcal{U}[-30, 30]$ frames		
	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
SV-MPI [14]	27.10	0.870	-	24.40	0.812	-	23.52	0.785	-
BTS [16]	-	-	-	-	-	-	24.00	0.755	0.194
Splatler Image [11]	28.15	0.894	0.110	25.34	0.842	0.144	24.15	0.810	0.177
MINE [5]	28.45	0.897	0.111	25.89	0.850	0.150	24.75	0.820	0.179
Flash3D [10]	28.46	0.899	0.100	25.94	0.857	0.133	24.93	0.833	0.160
Image2GS no masking	26.97	0.855	0.158	24.40	0.813	0.205	24.50	0.802	0.219
Image2GS	28.02	0.878	0.061	24.57	0.820	0.105	25.30	0.827	0.112

by unprojecting depth at pixel i to obtain μ_i . We initialize d_w and l_ϕ with parameters of DAV2 where possible, and the additional channels in d_w from scratch.

3.2. Dataset Preparation

Starting from a multiview image dataset $\mathcal{D} = \{S_1, \dots, S_n\}$ consisting of n scenes, where each scene $S = \{(x_i, c_i)\}_{i=1}^k$ contains a set of images x and corresponding camera poses c_i , we want to compute masks $m_{i \rightarrow j}$ for each pair of images i, j in S such that $\forall u, v \in [1, H] \times [1, W], x_j[u, v]$ is visible from x_i if $m_{i \rightarrow j}[u, v] = 1$, otherwise $m_{i \rightarrow j}[u, v] = 0$. A

possible way to compute $m_{i \rightarrow j}$ would be to use monocular depth prediction to predict relative depth maps for each image, and then use key points obtained in SfM to align the scale of these depth maps across images. However, initial experiments found this process to be quite fragile in cases of large perspective changes, and hence we opt to directly optimize 3D Gaussian splats for each scene to obtain a dataset of 3D Gaussian scenes $\mathcal{G} = \{G_1, \dots, G_n\}$. Following [4], we optimize each scene for 30000 iterations with default settings supplied by [18], with the exception of reducing the 3D scale threshold for duplication and pruning to 0.002 and 0.02 respectively, due to many scenes in the dataset only occupying a corner of the scene (hence smaller sizes) after scene scale normalization.

3.3. Model Training

We train Image2GS through masked novel view prediction. Each datapoint during training consists of a pair of input image x_{input} and target image x_{target} from the same scene, the camera transformation matrix from the input view to the target view K , and a mask $M \in [0, 1]^{H \times W}$ indicating the visibility of each pixel location in target view from the

input view. First, the model predicts $G = d_w(l_\phi(x))$ given input x . Then, the predicted image $\hat{x}_{target} = Splat(G, K)$ is rendered from G (implemented in gsplat[18]). Lastly, the loss is computed between \hat{x}_{target} and x_{target} , with the mask M providing weights per pixel. In this subsection, we detail how the mask is obtained from pretrained per-scene Gaussian splats, and how the mask is used in loss computation.

3.3.1 Obstruction Masking via projection

We project the per-scene Gaussian splats $G_i \in \mathcal{G}$ in the input view and the target view to obtain mean depth maps D_{input} and D_{target} . We then re-project D_{target} to the input view¹ and compare with D_{input} to determine whether each pixel in the target view is visible in the input view. We apply a scaled and shifted sigmoid activation to the difference in depth to obtain the soft mask: $M = \sigma(-3|D_{input} - D_{target}| - 0.05)$.

3.3.2 Loss formulation

We utilize the obstruction mask to train Image2GS with the pixel-based L2 loss and the feature-based LPIPS loss. For the L2 loss, the mask is multiplied pixel-wise to the L2 difference between the rendered image in the target view and the target image. For the LPIPs loss, early experiments found that directly applying the mask to the rendered and target images resulted in artifacts. Instead, we compute the proportion of unobstructed areas in the target view and multiply it with the LPIPS loss of the entire image. These loss functions are combined as follows:

$$L_{total} = \alpha_{L2} \|M \cdot (x_{target} - \hat{x}_{target})\|_2^2 + \alpha_{LPIPS} \frac{\sum_{i,j} M_{i,j}}{H \times W} LPIPS(x_{target}, \hat{x}_{target}),$$

with α_{L2} and α_{LPIPS} set to 1.0 and 0.1 respectively.

4. Experiments

Experiment settings We use the vitb backbone from DepthAnythingV2 for all of our experiments. We train the model on the RealEstate10K[20] dataset, which contains videos of mostly static indoor scenes captured in real-estate sales videos. We use the training and testing splits provided with RealEstate10K. The model is trained for 300,000 steps with a batch size of 32 using the Adam optimizer with a learning rate 5×10^{-5} for the decoder head, reduced by a factor of 10 for the backbone.

4.1. Effectiveness of Masking

First, we establish the effect of obstruction masking during training in both 3D lifting and 3D completion. We quan-

tify the quality of 3D lifting through masked PSNR and SSIM. For each input in the test set, we evaluate these metrics on four sets of target views: the original input view, a novel view 5 frames into the future/past, a novel view 10 frames into the future/past, and a novel view sampled uniformly within a 30 frame window extending from the input view. We opt not to evaluate LPIPS values as it tends to behave erratically when masking is applied to the rendered and generated images. To evaluate 3D completion, we use the standard PSNR, SSIM, and LPIPS metrics on full novel view images. In Table 1, we find that Image2GS trained with masking significantly outperforms the baseline in 3D lifting and 3D completion metrics. Of particular significant improvement is that Image2GS achieves near perfect reconstruction of the input view, showing a near 10 dB improvement in PSNR over the baseline, and over 2× reduction in LPIPS across all views.

We show several results of our masked model and no-mask model for qualitative comparison in Figure 2. Training with obstruction masking significantly reduces artifacts on foreground objects with large perspective changes. The reconstruction image with the masked model is also sharper (e.g., the cage of the cradle in the top left example and the slats of the chair in the bottom left example).

4.2. Comparison to SotA 3D reconstruction Models

In Table 2, we report the performance of Image2GS in the context of current state-of-art models in feed-forward 3D reconstruction. Following standard protocol[5, 10], we perform a 5% border crop of the image during evaluation. We find that Image2GS is competitive with other SotA methods in PSNR and SSIM, and outperforms in LPIPS. We hypothesize that this is due to our evaluation being performed at higher resolution than baselines (518×518 pixels versus 384×256 pixels), which is disadvantageous to our PSNR and SSIM numbers. Despite being trained for 3D lifting, Image2GS is competitive with purpose-trained 3D reconstruction models on the reconstruction task.

5. Conclusion

In this paper, we showed that in a controlled-setting, a discriminative feed-forward image-to-GS model training on image-to-3D lifting achieved better performance than that trained on image-to-3D completion in both image-to-3D lifting and image-to-3D completion tasks. Furthermore, we found that the lifting model is competitive with state-of-art 3D reconstruction models. We argue that this shows that these discriminative 3D reconstruction models struggle to model and learn meaningful content in obstructed regions, therefore they do not benefit from training on the full 3D reconstruction task.

¹by projecting to 3D and unprojecting to the input view

References

- [1] David Charatan, Sizhe Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *CVPR*, 2024. 2
- [2] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeyns, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. *arXiv preprint arXiv:2403.14627*, 2024. 2
- [3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2
- [4] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023. 1, 2, 3
- [5] Jiaxin Li, Zijian Feng, Qi She, Henghui Ding, Changhu Wang, and Gim Hee Lee. Mine: Towards continuous depth mpi with nerf for novel view synthesis. In *ICCV*, 2021. 2, 3, 4
- [6] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 1
- [7] Brandon Smart, Chuanxia Zheng, Iro Laina, and Victor Adrian Prisacariu. Splatt3r: Zero-shot gaussian splatting from uncalibrated image pairs. 2024. 1
- [8] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 1
- [9] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024. 1
- [10] Stanislaw Szymanowicz, Eldar Insafutdinov, Chuanxia Zheng, Dylan Campbell, Joao Henriques, Christian Rupprecht, and Andrea Vedaldi. Flash3d: Feed-forward generalisable 3d scene reconstruction from a single image. *arxiv*, 2024. 1, 2, 3, 4
- [11] Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 3
- [12] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. *arXiv preprint arXiv:2402.05054*, 2024. 2
- [13] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. 2024. 1
- [14] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 3
- [15] Christopher Wewer, Kevin Raj, Eddy Ilg, Bernt Schiele, and Jan Eric Lenssen. latentsplat: Autoencoding variational gaussians for fast generalizable 3d reconstruction. In *European Conference on Computer Vision (ECCV)*, 2024. 2
- [16] Felix Wimbauer, Nan Yang, Christian Rupprecht, and Daniel Cremers. Behind the scenes: Density fields for single view reconstruction. *arXiv preprint arXiv:2301.07668*, 2023. 2, 3
- [17] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024. 2
- [18] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An open-source library for Gaussian splatting. *arXiv preprint arXiv:2409.06765*, 2024. 3, 4
- [19] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 1
- [20] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018. 4