OracleGS: Grounding Generative Priors for Sparse-View Gaussian Splatting

Supplementary Material

In this supplementary material, we provide more detailed explanations of the proposed method. First, we explain the implementation details including the process of point cloud generation in Section A.1, and depth map generation in Section A.2. Subsequently, we offer an in-depth explanation of our 3D-Aware Oracle's construction (Section A.3). This includes the technical process of extracting attention maps from the VGGT model, a layer-by-layer analysis of the attention's evolution, and a detailed rationale for our design choice of using a weighted average of layers 0 and 22 to form the final uncertainty signal. Afterwards, we share training hyperparameters in section A.5. Extended qualitative and quantitative results on Mip-NeRF 360 [1] and NeRF Synthetic [6] are provided in Section B. Importantly, we share our ablation study on our novel Progressive Augmentation Strategy in Section C. Finally, we discuss the limitations and future work in Section D.

A. Implementation Details

A.1. Point Cloud Generation from Sparse-Views.

We generate a sparse 3D point cloud for a predefined subset of 12-24 training views for Mip-NeRF 360 [1] and 8 training views for NeRF Synthetic [6] using a custom COLMAP [8, 9] pipeline. The process begins with GPU-accelerated SIFT feature extraction [5], configured to extract up to 16,384 features per image with a maximum dimension of 4032 pixels, with both affine shape estimation and domain-size pooling enabled. Following extraction, an exhaustive matching strategy is employed, retaining up to 32,768 matches per image pair and utilizing guided matching. We bypass the standard Structure-from-Motion (SfM) pipeline for pose estimation. Instead, we leverage known camera extrinsics (rotation quaternions and translation vectors) from a pre-existing, complete reconstruction of the scene. These ground-truth poses are programmatically fixed for the training views, after which COLMAP's point triangulator is executed to compute the 3D point locations by minimizing reprojection error across all feature correspondences and includes bundle adjustment refinement steps, configured with 40 local iterations, 3 local refinements, and 100 global iterations. The final output is a sparse point cloud whose 3D points are newly triangulated from the selected views but are inherently registered within the coordinate system of the original, complete model.

A.2. Depth Map Generation

For depth regularization, we generate depth maps using the Depth Anything V2 Large model [11], which contains 335.3M parameters and operates on full-resolution sparse-input training images. As our focus is not on depth regularization, we adopt the out-of-the-box depth regularization strategy from the original 3DGS codebase [4], which is compatible with Gaussian Splatting training. To set scene scale, for each image, we project the sparse 3D points from COLMAP into the camera view to get a set of sparse, metrically-scaled depth values. We then compare these values to the corresponding depth values sampled from the monocular depth map at the same pixel locations. Finally, we compute and save a per-image scale and offset, which allows the arbitrarily-scaled monocular depth maps to be transformed into the metrically consistent scale of the COLMAP [8, 9] world.

A.3. Uncertainty Map Extraction

We repurposed the global-attention maps of VGGT [10], the regressive MVS model as a proxy for 3D uncertainty. Our visualization utility extracts attention maps from any specified global attention layer. Extracting maps for multiple layers, such as layers 0 and 22, requires separate executions. Each execution performs one complete forward pass through the model to generate the attention map for the single specified layer. This design ensures that the attention mechanism is analyzed in its true context at that specific depth, without altering the model's standard inference flow.

A key technical consideration is the incompatibility of visualization with fused attention mechanisms like Flash Attention [2, 3]. Our model leverages `functional.scaled_dot_product_attention` from torch.nn for optimized training and inference. However, when a layer's attention map needs to be visualized, our implementation explicitly disables this fused kernel and reverts to the slower and memory hungry classical attention computation. This is a necessary step because fused attention is an opaque operation that computes the final output directly from queries, keys, and values, without ever materializing the intermediate $N \times N$ attention matrix in memory. To inspect the attention scores, this matrix must be explicitly computed, which is handled by the non-fused fallback path.

The visualization path is further optimized for memory efficiency. Upon reaching the target global attention block, a specialized computation is triggered. Rather than computing the full query matrix $\mathbf{Q} \in \mathbb{R}^{N_{total} \times d_k}$, we isolate the single query vector $\mathbf{q}_{cam}$ corresponding to the camera token of the designated source frame. The attention scores are then computed efficiently via the matrix-vector product: $\mathbf{A} = \text{softmax}(\frac{\mathbf{q}_{cam}\mathbf{K}^T}{\sqrt{d_k}})$, where $\mathbf{K} \in \mathbb{R}^{N_{total} \times d_k}$ con-

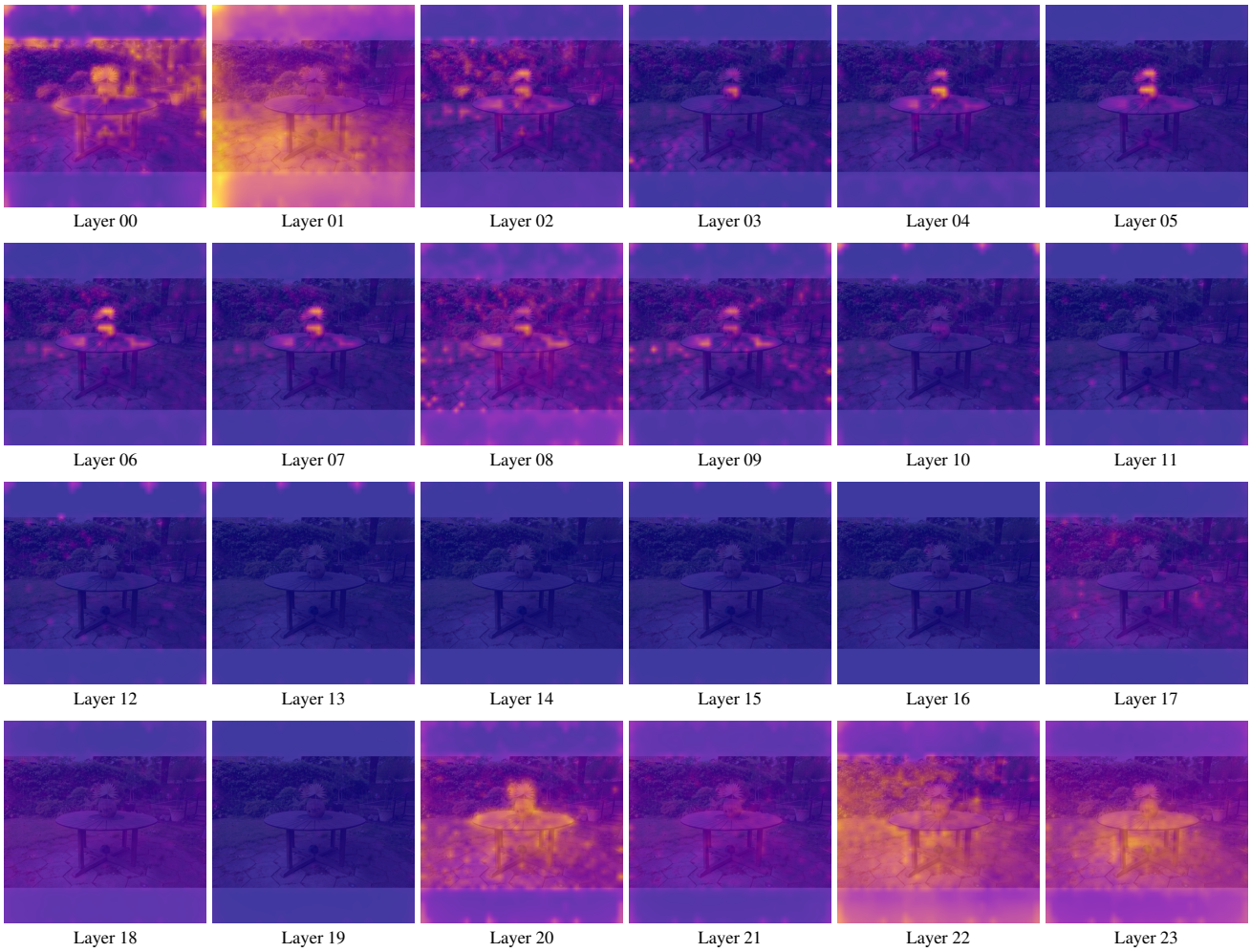

Figure 1. **Evolution of Global Attention in VGGT for Oracle Layer Selection.** Visualization of global attention from a single source view's camera token to all patch tokens across all 24 global layers of the VGGT aggregator. This sequence illustrates how the model's understanding of 3D consistency evolves. **Early layers (e.g., 00-02)** exhibit diffuse attention based on low-level feature matching. **Mid-layers (e.g., 03-09)** sharply focus on high-confidence foreground geometry, demonstrating strong regressive fidelity but ignoring the wider context. **Late layers (e.g., 20-23)** develop a more semantic understanding, highlighting the main object and its immediate, plausible surroundings. Our oracle's use of a weighted average of Layer 0 and Layer 22 is informed by this evolution, balancing broad scene completeness with high-level structural and semantic confidence.

tains the keys of all tokens (camera, register, and patch tokens) from all frames. This targeted approach yields a single attention map of shape $(B, H, 1, N_{total})$, where B is the batch size and H is the number of heads. This computation is performed in parallel to the standard feature propagation and does not alter the block's output features; its sole purpose is to extract the attention weights. The resulting map is then post-processed: scores are averaged across all attention heads, and the weights corresponding to the image patches are reshaped into a normalized 2D grid for each input view, producing the final heatmaps.

A.3.1. Justification for Oracle Layer Selection

Our "3D-Aware Oracle" repurposes the global attention maps from a pretrained MVS model, VGGT [10], as a proxy for 3D uncertainty. VGGT's aggregator consists of 24 sequential global attention blocks. As input tokens propagate through these layers, the nature of the attention signal evolves from capturing low-level feature correspondences to encoding high-level semantic and structural relationships. To determine the optimal layers for our uncertainty signal, we visualized the attention map from a single source view to all other views across all 24 global layers, as shown in Figure 1. This analysis reveals a distinct pattern that validates our choice of layers $\mathcal{L} = \{0, 22\}$.

Analysis of Attention Evolution The attention maps in Figure 1 reveal three distinct phases of processing within the MVS model:

- **Early Layers (0-2):** In the initial layers, attention is diffuse and widespread, indicating that the model is matching simple, low-level features like color and texture across all views. Layer 0, in particular, shows high correlation across nearly the entire scene. While this signal provides a useful baseline for scene *completeness*, it lacks the geometric specificity to distinguish between structurally sound and inconsistent regions.
- **Mid Layers (3-9):** The attention mechanism rapidly converges, sharply focusing on the foreground object where multi-view geometric consistency is highest. The background and less certain regions receive almost no attention. While these layers provide a strong signal for *regressive fidelity*, they are too conservative for our purpose. Using them would cause the oracle to aggressively reject nearly all plausible completions proposed by the generative model, undermining our goal of reconciling completeness with fidelity.
- **Late Layers (20-23):** In the final layers, the attention re-emerges with a more semantic and contextual understanding. Layer 22, which we select, provides the ideal balance. It assigns high confidence to the primary foreground object, yet it also assigns moderate confidence to the surrounding background foliage. This demonstrates a holistic understanding of the scene's composition. It recognizes not just the object, but its plausible environment.

Justification for $\mathcal{L} = \{0, 22\}$ Our final uncertainty map is a weighted average of attention from Layer 0 (weight 1/4) and Layer 22 (weight 3/4). This specific combination is designed to balance two objectives:

1. **Layer 0** provides a broad, low-level signal that ensures the entire scene is considered, preventing the oracle from being overly punitive in under-observed regions.
2. **Layer 22** provides a mature, high-level signal that strongly grounds the uncertainty in semantically and structurally coherent regions, effectively identifying and filtering generative hallucinations.

By heavily weighting the semantic signal from Layer 22 while retaining a baseline from Layer 0, our oracle produces a nuanced uncertainty map that robustly guides the 3DGS optimization, fulfilling the core principle of our "propose-and-validate" framework.

A.4. Synthetic Image Generation

We use Stable-Virtual-Camera [13] with the `img2img` task, with an `orbital` trajectory prior and a `nearest-gt` chunking strategy. The generation parameters are set as `CFG=3.0`, training context window length `T=80`, and shortest size length$\texttt{L}_{\texttt{short}} = 576$.

Scene	PSNR ↑	SSIM ↑	LPIPS ↓
Chair	28.369	0.955	0.036
Drums	21.149	0.885	0.073
Ficus	23.879	0.921	0.058
Hotdog	29.134	0.957	0.037
Lego	23.400	0.871	0.087
Materials	19.381	0.860	0.091
Mic	26.950	0.959	0.027
Ship	25.709	0.830	0.124
AVG	**24.746**	**0.905**	**0.067**

Table 1. **Per-scene results on the NeRF Synthetic [6] dataset** for 8 input views. We report PSNR, SSIM, and LPIPS metrics. The average scores across all scenes are highlighted in bold.

A.5. Hyperparameters

For **Mip-NeRF 360** [1], we train and evaluate scenes under both 12- and 24-view settings with the following where for 12-view, all scenes are trained to 15k iterations, and for 24-view, all scenes are trained to 22k iterations.

The maximum number of splats used per scene is as follows: bicycle 1,200,000; bonsai 700,000; counter 900,000; garden 900,000; kitchen 700,000; room 800,000; stump 600,000.

For NeRF Synthetic [6], all evaluations are performed at 15k iterations with white backgrounds. The number of maximum splats used per scene for chair, drums, ficus, hotdog, lego scenes is 200,000; for materials, mic, and ship 400,000 splats are used.

Figure 2. **Mode of failure due to Generative NVS Model.** Stable-Virtual-Camera (SEVA) [13] fails to produce results that reflect scene structure correctly in the "'stump"' scene from Mip-NeRF 360 [1]. (Left) synthetic images generated by SEVA. (Right) ground truth images from similar viewpoints. Despite this, OracleGS robustly handles stump scene and is included in full for all evaluations and ablations.

CoR-GS [12] DropGaussian [7] OracleGS (Ours) Ground Truth

Figure 3. **Further results on visual comparison with state-of-the-art methods on the Mip-NeRF360 [1] dataset**.

Scene	12-view			24-view		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Bicycle	18.781	0.383	0.511	20.450	0.452	0.415
Bonsai	20.634	0.713	0.274	25.983	0.868	0.152
Counter	20.116	0.663	0.299	24.237	0.813	0.172
Garden	20.287	0.544	0.328	23.980	0.749	0.194
Kitchen	21.483	0.745	0.218	25.494	0.880	0.122
Room	22.341	0.773	0.253	25.630	0.859	0.177
Stump	18.575	0.353	0.568	20.291	0.446	0.477
AVG	**20.317**	**0.596**	**0.350**	**23.723**	**0.723**	**0.244**

Table 2. **Per-scene results on Mip-NeRF 360 dataset.** We report PSNR, SSIM, and LPIPS metrics for our method trained with 12 and 24 input views. Average scores are highlighted in bold.

B. More Experiment Results

B.1. Results on Mip-NeRF 360

The per-scene quantitative results on the Mip-NeRF 360 dataset are presented in Table 2. A notable case is the *stump* scene, where the underlying generative NVS model failed to produce coherent novel views, as visualized in Figure 2. This failure of the generative prior directly impacted our final metrics, underscoring that the performance of OracleGS is inherently dependent on the quality of the initial scene proposal. Further qualitative comparisons on the Mip-NeRF 360 dataset are provided in Figure 3.

Row 1 (Bonsai, 24 views): DropGaussian [7] exhibits significant artifacts from stretched, anisotropic Gaussians that fail to represent the high-frequency texture of the knitted purple fabric. CoR-GS [12] suffers from under-

reconstruction and oversmoothing, particularly in the upper portion of the scene. In contrast, our method faithfully reconstructs these fine details while maintaining overall structural integrity.

Row 2 (Counter, 24 views): On the `counter` scene, DropGaussian [7] produces prominent brown floating artifacts in the foreground. CoR-GS [12] also generates floaters, concentrated along the under-observed sides of the kitchen island. Our method substantially suppresses these artifacts, though it introduces minor oversmoothing on the far sides of the counter as a trade-off for improved cleanliness.

Row 3 (Kitchen, 12 views): In this highly sparse setting, CoR-GS [12] exhibits severe floater artifacts near the camera, which is indicative of overfitting to ambiguous regions. Our method successfully eliminates these foreground floaters, yielding a much cleaner result. However, minor inconsistencies persist in distant background elements, such as the partially reconstructed red broomstick, where multi-view constraints are weakest.

Row 4 (Bicycle, 12 views): The `bicycle` scene with 12 views is among the most challenging scenarios, as reflected by the quantitative metrics in Table 2. Competing methods fail to reconstruct a coherent structure, producing blurry and fragmented results. While OracleGS does not achieve a perfect reconstruction, exhibiting some texture artifacts on the bench and oversmoothing in the background, it preserves the global scene structure and object coherence far more effectively.

Row 5 (Garden, 12 views): The background foliage proves difficult for competing methods, which either display severe smearing (DropGaussian) [7] or suffer from a loss of detail, resulting in blurry and incomplete regions (CoR-GS) [12]. Guided by our uncertainty-aware optimization, our method reconstructs the background with significantly higher fidelity and completeness.

B.2. Results on NeRF Synthetic

We present our quantitative results on NeRF Synthetic [6] on Table 1.

Figure 4 provides a qualitative comparison on the NeRF Synthetic dataset, which features object-centric scenes with complex materials and fine structures. Our method consistently produces cleaner and more faithful reconstructions compared to prior work.

Lego (Row 1): This scene highlights our method's effectiveness at eliminating floating artifacts. CoR-GS [12] produces significant spurious geometry and high-frequency noise around the object. While DropGaussian [7] reduces these artifacts, some residual floaters persist. OracleGS generates a significantly cleaner reconstruction, preserving sharp object boundaries without the distracting peripheral noise.

Figure 4. **Further visual comparison with state-of-the-art methods on the NeRF Synthetic [6] dataset** on 8 training views.

Materials (Row 2): The primary challenge in this scene is the accurate rendering of non-Lambertian surfaces with complex specular reflections. CoR-GS [12] suffers from catastrophic failure, with severe degradation in both appearance and structure. DropGaussian [7] manages to preserve the basic shapes but introduces a diffuse, hazy glow that corrupts the reflections and scene background. Our method reconstructs the challenging specularities on the metallic spheres with high fidelity, closely matching the ground truth with few floater artifacts.

Mic (Row 3): This scene tests the ability to reconstruct both smooth textures and fine-grained structures. The metallic texture of the microphone in the CoR-GS [12] reconstruction is corrupted by grainy, high-frequency noise. In contrast, OracleGS robustly reconstructs these thin structures while rendering the smooth metallic surface of the microphone accurately.

Ship (Row 4): Here, the difficulty lies in modeling the boundary between the central object and its surrounding medium. CoR-GS [12] fails to reconstruct the water, resulting in a severe geometric collapse around the ship. DropGaussian [7] defines the water's shape but renders it with a blurry, hazy appearance that lacks a crisp boundary. Our method successfully reconstructs the entire scene, rendering the water with a well-defined surface and a sharp, clean edge that aligns almost perfectly with the ground truth.

C. Ablation Study on Progressive Augmentation Strategy

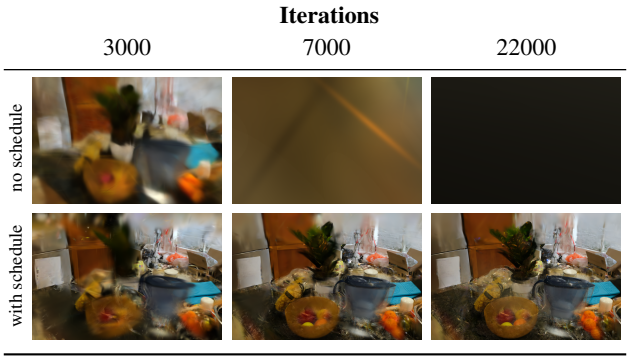

Figure 5. **Qualitative ablation of the training schedule across iterations** to demonstrate the stabilizing effect of our novel curriculum. Without the curriculum (top row), naively incorporating synthetic views leads to catastrophic geometric collapse as early as 7k iterations. With our curriculum (bottom row), a coherent geometric scaffold is established early on and is refined to a clean final result. Note that both methods are shown without our other contributions (e.g., depth supervision, LPIPS) to purely isolate the effect of the schedule.

A direct comparison between our progressive augmentation strategy and a naive, static approach reveals the critical role our curriculum plays in stabilizing the training process, as visualized in Figure 5. The method without the schedule (top row) fails catastrophically because it is immediately subjected to conflicting supervisory signals. The dense but imperfect synthetic views introduce a high volume of noisy gradients that overwhelm the sparse, high-fidelity signal from the ground-truth images. This conflict corrupts the fragile initial structure, causing the optimization to diverge completely. In contrast, our method with the progressive curriculum (bottom row) succeeds by strategically staging the introduction of information. By prioritizing ground-truth views in the early stages, it first establishes a stable and geometrically coherent foundation. Once this anchor is in place, the curriculum introduces the synthetic views, allowing them to safely densify and complete the scene without the risk of destabilizing the entire representation. This controlled process transforms the synthetic data from a source of destructive noise into a constructive scaffold, enabling stable convergence towards a detailed and complete final result.

D. Limitations and Future Work

The performance of OracleGS is inherently upper-bounded by the quality of the 3D-aware generative model used in the "propose" stage. While our geometric oracle is effective at filtering inconsistencies, it cannot correct a fundamentally flawed or collapsed generative prior, as observed in challenging cases like the 'stump' scene, as demonstrated in Figure 2. However, the modular, plug-and-play nature of our framework means it is set to automatically inherit improvements from the rapidly advancing field of 3D generative modeling. As more powerful generative priors become available, they can be integrated into our pipeline, promising a path toward continued state-of-the-art performance.

References

[1] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR*, 2022. 1, 3, 4

[2] Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*, 2024. 1

[3] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 1

[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

[5] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 1

[6] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1, 3, 5

[7] Hyunwoo Park, Gun Ryu, and Wonjun Kim. Dropgaussian: Structural regularization for sparse-view gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21600–21609, 2025. 4, 5

[8] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[9] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 1

[10] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 1, 2

[11] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv:2406.09414*, 2024. 1

[12] Jiawei Zhang, Jiahe Li, Xiaohan Yu, Lei Huang, Lin Gu, Jin Zheng, and Xiao Bai. Cor-gs: Sparse-view 3d gaussian splatting via co-regularization. *arXiv preprint arXiv:2405.12110*, 2024. 4, 5

[13] Jensen (Jinghao) Zhou, Hang Gao, Vikram Voleti, Aaryaman Vasishta, Chun-Han Yao, Mark Boss, Philip Torr, Christian

Rupprecht, and Varun Jampani. Stable virtual camera: Generative view synthesis with diffusion models. *arXiv preprint arXiv:2503.14489*, 2025. 3