DreamScene360: Unconstrained Text-to-3D Scene Generation with Panoramic Gaussian Splatting

Shijie Zhou[1⋆], Zhiwen Fan[2⋆], Dejia Xu[2⋆], Haoran Chang[1], Pradyumna Chari[1], Tejas Bharadwaj[1], Suya You[3], Zhangyang Wang[2], and Achuta Kadambi[1]

[1] University of California, Los Angeles
[2] University of Texas at Austin
[3] DEVCOM Army Research Laboratory

Supplementary Content

This supplement is organized as follows:

– Section A contains the algorithmic details of our proposed method.
– Section B contains the experimental details of our self-refined panorama generation module with GPT-4V.
– Section C contains the details of the environment setup for our experiments.
– Section D contains additional qualitative results.
– Section E contains additional results on baseline comparison.
– Section F contains additional ablation studies.

A Algorithmic Details

Given the input text prompt with any level of specificity, we utilize GPT-4V to perform prompt revision and image quality assessment inspired by Idea2Img [7]. Different from their original setup which only works for generating ordinary 2D images, our multi-round self-refinement module is incorporated with the pretrained text-to-360° panoramic image diffusion model [6] which guarantees the generated panorama images fully satisfy the equirectangular format for an ominidirectional image representation. An upsampling process by bilinear interpolation is followed by obtaining the best candidate panorama image result, to ensure a sufficiently dense point cloud for the 3D Gaussians initialization. In practice, we resize the image from 512×1024 to 1024×2048 using OpenCV [1]. For the geometric field optimization, we use one set of Icosphere projection: 20 perspective images with FoV 80° and resolution 512×512 to cover the whole panoramic sphere, following [4] and [5]. For the optimization of panoramic Gaussian splatting, the training set consists of 12 sets of perspective images of the Icosphere projection with random rotation in radian. Additionally, different from 3D Gaussian Splatting (3DGS) [3], we turn off the Gaussians densification as

⋆ Equal contribution.

our dense point cloud initialization has already accurately captured the surfaces in the scene and redundant 3D Gaussians produced from densification will result in floaters near the surfaces and in the empty space. Algorithm 1 illustrates the pseudo code of our DreamScene360.

Algorithm 1 DreamScene360

Input: $TextPrompt_0$
$i \leftarrow 0$ ▷ Iteration Counter
while not maximum iteration **do**
  $TextPrompt_i \leftarrow$ GPT-4V($TextPrompt_i$) ▷ Prompt Revision
  $P_i \leftarrow$ PreTrainedDiffusionModel($TextPrompt_i$) ▷ Panorama Image Candidate
  $S_i \leftarrow$ GPT-4V(P_i) ▷ Candidate Score
  $TextPrompt_{i+1} \leftarrow$ GPT-4V($TextPrompt_i$) ▷ Self-Refinement
end while
$P \leftarrow MaxScore(P_s)$ ▷ Select Panorama with Best Score
$P \leftarrow UpSample(P)$ ▷ Upsample by bilinear interpolation
$I_N, C_N \leftarrow$ Projection(P) ▷ Get Perspective Images and Camera Poses
$D_N^{Mono} \leftarrow$ DepthEstimator(I_N) ▷ Get Monocular Depth
GeometricField $\leftarrow D_N^{Mono} + v$ ▷ Learn GeometricField with view direction
$PointCloud \leftarrow$ GeometricField ▷ Global Depth Alignment
$Gaussians \leftarrow PointCloud$ ▷ 3D Gaussians Initialization
while not converged **do**
  $I, C, D \leftarrow$ GetTrainingView() ▷ Given Image, Camera Pose, Depth Map
  $\hat{I}, \hat{D} \leftarrow$ Rasterizer($Gaussians, C$) ▷ Rendered Image, Remdered Depth Map
  $L \leftarrow \text{Loss}(I, \hat{I}) + \lambda_d \text{Loss}(D, \hat{D})$ ▷ Loss Calculation
  if PerturbationIterationStages **then**
    $I, C' \leftarrow$ GetVirtualView() ▷ Virtual Camera Pose, Imagep
    $\hat{I}', \hat{D}' \leftarrow$ Rasterizer($Gaussians, C'$)
    ▷ Rendered Perturbation Image, Rendered Perturbation Depth Map
    $D' \leftarrow$ DepthEstimator($\hat{I}'$) ▷ Perturbation Depth Map
    $[CLS] \leftarrow$ FeatureExtractor(I) ▷ Reference DINO Token
    $[CLS]' \leftarrow$ FeatureExtractor(I') ▷ Perturbation DINO Token
    $L \leftarrow L + \lambda_f \text{Loss}([CLS], [CLS]') + \lambda_d \text{Loss}(D', \hat{D}')$ ▷ Loss Calculation
  end if
  $Gaussians \leftarrow$ Adam(L) ▷ Optimization
end while

B Self-Refinement for Panoramic Image Generation

We demonstrate the workflow of our self-refinement module using GPT-4V for multi-round text prompt revision and image quality evaluation for our text to 360° panoramas generation. In our implementation, we fix the trigger word *360-degree panoramic image* as a compulsory part of the input text prompt for the Text2Pano diffusion model [6]. We also adopt negative prompts to ensure the generation quality: *lowres, bad anatomy, bad hands, text, error, missing fingers,* *extra digit, fewer digits, cropped, worst quality, low quality, normal quality, jpeg artifacts, signature, watermark, username, blurry.*

Take the input text prompt **"Castle on the hill"** as an example, we showcase one round of refinement and assessment by GPT-4V as follows:

iteration 0
prompt: 360-degree panoramic image, Majestic castle perched atop a lush green hill, medieval architecture, serene sky backdrop.

iteration 1
prompt: 360-degree panoramic image, Ancient stone castle standing on a windswept hill, ominous clouds gathering above, twilight setting.

iteration 2
prompt: 360-degree panoramic image, Grand castle with towering spires on a sunlit hill, surrounded by wildflowers, clear blue sky.

iteration 3
prompt: 360-degree panoramic image, Imposing castle on a steep hill, silhouetted against a fiery sunset, foreboding atmosphere.

iteration 4
prompt: 360-degree panoramic image, Enchanted castle nestled on a misty hill, fairy tale inspired, surrounded by an enchanted forest, dawn light.
===============
End of round 0:
user prompt: Castle on the hill.
select history: Round selection: 2.
To evaluate how well these images match the user's idea of a "Castle on the hill," I will consider the presence of a castle, its placement on a hill, and the overall composition that aligns with this concept.

Image 0: There is a castle on a hill in the distance, surrounded by a green landscape and a clear sky. This image closely matches the idea. Score: 8/10

Image 1: This image shows a hill, but there is no castle present. It does not match the idea well. Score: 2/10

Image 2: There is a castle with multiple spires on a hill with a sunny sky and some flowers in the foreground. This image aligns well with the idea. Score: 9/10

Image 3: A castle is present on a hill with a dramatic sunset sky. This image also aligns well with the idea. Score: 8/10

Image 4: This image shows a hill with trees and fog but lacks a castle. It does not match the idea. Score: 1/10
===============
Considering the scores and the adherence to the idea of a "Castle on the hill," the image with the best overall quality is: 2

Correspondingly, Fig. A presents the panoramic images generated at each iteration. It is evident that the selection made by GPT-4V aligns with both input text description and human evaluation outcomes.

Fig. A: Self-refinement by GPT-4V. Input text prompt: "Castle on the hill".

C Environment Setup:

We describe the detailed setup that has been used in our experiments and runtime tests:

- CUDA version: 11.8
- PyTorch version: 2.0.1
- GPU: single NVIDIA GeForce RTX 4090
- CPU: 13th Gen Intel(R) Core(TM) i9-13900KF 3.00 GHz

Generally, one text-to-3D generation experiment takes no more than 16GB GPU memory.

D Additional Qualitative Results

We present further qualitative findings in Fig. B, showcasing the generated 2D panorama, along with the rendered perspective images and depth maps from our panoramic Gaussian radiance field. The results demonstrate that our DreamScene360 is adept at generating varied 3D scenes with precise geometry, including both unbounded outdoor scenes and bounded indoor scenes with different kinds of styles. Additionally, we highly recommend viewers explore our video demonstrations for an immersive experience of "flying" (freely translating and rotating) within the generated 3D scene.

E Additional Results on Baseline Comparison

E.a Text-to-3D Generation

We present more visual results compared to LucidDreamer [2] in Fig. C, captured with a clockwise rotation in yaw and random translation. Unlike LucidDreamer, which exhibits repetitive patterns in its first (*Machu Picchu*) and second (*Christ the Redeemer*) examples, our method maintains consistent photographic quality throughout the scene. Furthermore, while LucidDreamer's prior front views diverge from its generated the back views, our approach ensures a more coherent visual transition. Notably, in the third example (*Old Hall*), LucidDreamer's outputs demonstrate inconsistencies in geometry and feature black spots under some random translations, showcasing our advantages in maintaining geometric consistency.

E.b Panorama-to-3D Lifting

PERF [5] is a great NeRF-based panorama-to-3D lifting work demonstrated on indoor scenes, whereas DreamScene360 is for text-to-3D scene generation using Gaussian Splatting. Thanks to our geometric initialization and refinement step, we can lift panoramic images showcasing a wide range of scene contexts and artistic styles (e.g., indoor, outdoor, cartoon) to 3D. The comparison with

Table A: Quantitative results: ours vs. PERF [5]

	PSNR↑	SSIM↑	LPIPS↓	Training Time
PERF	36.881	0.957	0.074	44min.01sec.
Ours	**40.179**	**0.970**	**0.035**	9min.32sec.

PERF is not included in the main paper since PERF is not a 3DGS-based text-to-3D scene generator. However, since the panorama-to-3D part is comparable, we present comparison results using the same panorama image as input. Fig. D shows ours achieving more accurate geometry (PERF has zigzaging artifacts at bed edge, see red arrows) because PERF directly performs inpainting on 2D images to train a NeRF, while we use rendered images and both semantic and geometric supervision to guide optimization in 3D. Tab. A shows that ours is superior across all metrics of rendering quality, while being significantly faster.

Besides rendering quality, occlusion is a fundamental problem in single view panorama to 3D reconstruction. DreamScene360 starts from an omnidirectional panoramic image and lift it to 3D using global depth cues. The proposals in "Distilling Semantic Similarities" and "Regularizing Geometric Correspondences" on unseen views are specifically designed to rectify occlusions. We compare our method with PERF by translating right from given single training view (leftmost Fig. D), and we observe that our 3D approach achieves similar degree of occlusion compared to PERF's 2D inpainting (see Fig. D, rightmost column) but with 4.6× faster training (Tab. A).

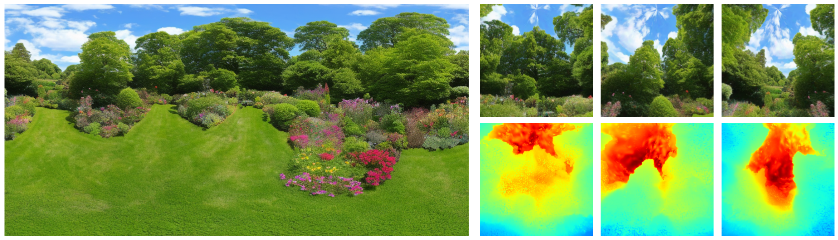
F Additional Ablation Studies

F.a Ablations of Virtual Camera Perturbations.

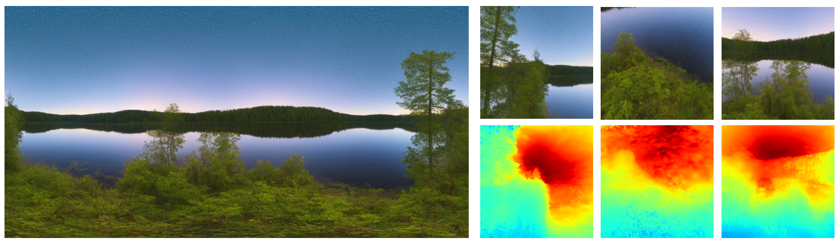
To emulate the camera's movement from the original view point, we introduce virtual cameras with 3-stage progressive perturbations with 3 levels. In our experiments, we start the 3-stage progressive camera perturbation from 5400 iterations during the training, the interval of each stage is adjustable, and we set it to 1200 iterations in practice. With the same observation in [3], most of the scenes at 9000 iterations can already achieve very high quality. We study the influence of perturbations on the results. Fig. E shows the qualitative results with different level of perturbations while keeping the same total number of iterations.

F.b Ablations of FOV.

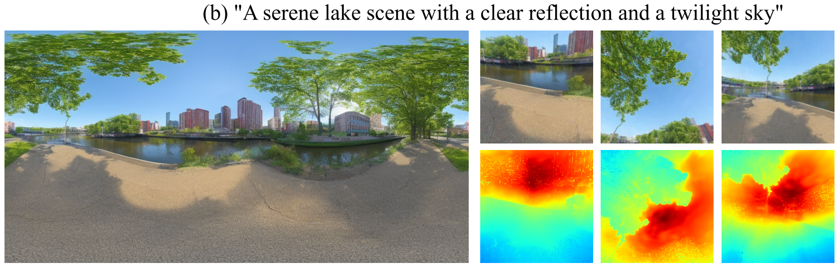
During inference, the field of view (FOV) is one of the adjustable parameters. In practice, we set a large camera FOV (80°) when rendering. Since we use the pinhole camera model, a large FOV may result in distortion. We show results with different FOVs in Fig. F, where it is obvious that a smaller FOV alleviates these distortions.



(a) "English garden"

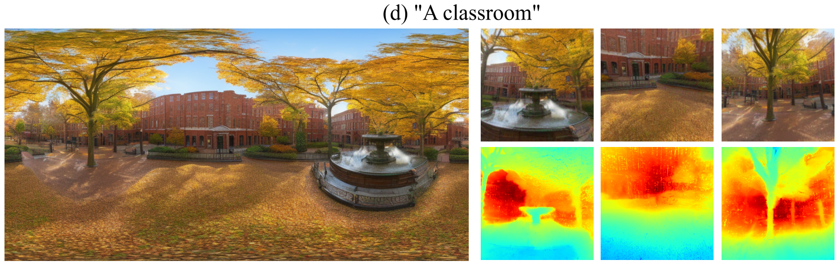


(b) "A serene lake scene with a clear reflection and a twilight sky"



(c) "River bank with a sidewalk and buildings"

(d) "A classroom"



(e) "Painting of college campus"

Fig. B: Additional Qualitative Results on Diverse Generation.

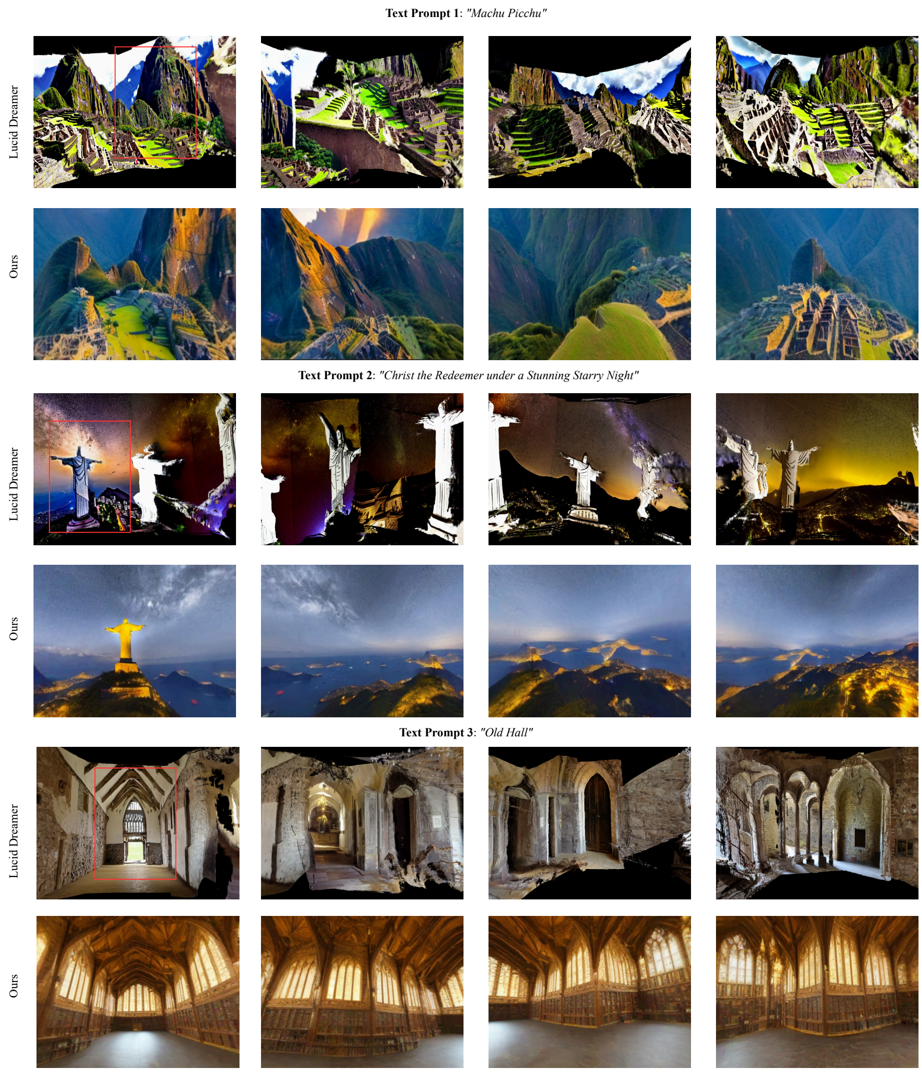


Fig. C: Additional Baseline Comparisons. DreamScene360 v.s. LucidDreamer [2] regarding text-to-3D scene generation.

Fig. D: Additional Baseline Comparisons. DreamScene360 v.s. PERF [5] regarding panorama-to-3D lifting. We show comparable visual results on novel views to PERF. In addtion to that, our work shows better geometry on the edges.

(a) No Pert. (b) One-stage Pert. (c) Two-stage Pert. (d) Three-stage Pert.

Fig. E: Ablation of Virtual Camera Perturbations. We show the influence of adding levels of virtual camera perturbations; the results improve progressively with more stages involved.

Fig. F: Ablation of Inference FOV. FOV set to 36°, 80°, 102°, 118° (left to right).

References

1. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)
2. Chung, J., Lee, S., Nam, H., Lee, J., Lee, K.M.: Luciddreamer: Domain-free generation of 3d gaussian splatting scenes. arXiv preprint arXiv:2311.13384 (2023)
3. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics (ToG) **42**(4), 1–14 (2023)
4. Rey-Area, M., Yuan, M., Richardt, C.: 360monodepth: High-resolution 360deg monocular depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3762–3772 (2022)
5. Wang, G., Wang, P., Chen, Z., Wang, W., Loy, C.C., Liu, Z.: Perf: Panoramic neural radiance field from a single panorama. arXiv preprint arXiv:2310.16831 (2023)
6. Wang, H., Xiang, X., Fan, Y., Xue, J.H.: Customizing 360-degree panoramas through text-to-image diffusion models. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 4933–4943 (2024)
7. Yang, Z., Wang, J., Li, L., Lin, K., Lin, C.C., Liu, Z., Wang, L.: Idea2img: Iterative self-refinement with gpt-4v (ision) for automatic image design and generation. arXiv preprint arXiv:2310.08541 (2023)