Uncertainty-Aware Diffusion-Guided Refinement of 3D Scenes

Supplementary Material

1. Preliminaries

1.1. 3D Gaussian Splatting

Gaussian Primitives (γ_n). A 3D scene can be explicitly represented by a set of anisotropic Gaussian ellipsoids with positions $\mu_n \in \mathbb{R}^3$, covariance matrix $\Sigma_n \in \mathbb{R}^{3\times3}$, color $c_n \in \mathbb{R}^{3\times(k+1)^2}$ for order k, typically represented using Spherical Harmonic (SH) coefficients and opacity $\alpha_n \in [0, 1]$. For each Gaussian point **x**, it's 3D position is given as

$$G(x) = e^{-\frac{1}{2}(x-\mu_n)^\top \Sigma_n^{-1}(x-\mu_n)}, \tag{1}$$

The Σ_n is decomposed into two learnable components represented by the scaling matrix S_n and a rotation matrix R_n as follows:

$$\Sigma_n = R_n\, S_n\, S_n^\top\, R_n^\top. \tag{2}$$

Therefore any scene can be represented as a collection of Gaussian primitives where each primitive can be represented as $\gamma_n := (\mu_n, R_n, S_n, \alpha_n)$.

Rasterization. The trainable parameters acquired within the primitive γ_n can be optimized via the application of the ensuing differentiable rendering function:

$$I_0(p) = \sum_{n=1}^{N} c_n\, \tilde{\alpha}_n \prod_{m=1}^{n-1} (1 - \tilde{\alpha}_m), \tag{3}$$

where $I_0(p)$ represents the rendered color at pixel **p** in rendered image I_0 and $\tilde{\alpha}_n$ is calculated from the back-projected 2D Gaussians.

1.2. Latent Video Diffusion Models (LVDMs)

Video Diffusion Model. Latent Video Diffusion Models consist of a pre-trained encoder $\mathcal{E}$, a U-Net denoiser ϵ_θ and a pre-tained decoder $\mathcal{D}$. The diffusion process occurs in the latent space. Given an image $\mathcal{I}$, it is initially embedded in the latent space via the frozen encoder $\mathcal{E}$ yielding latent $z_0^{1:M} = \mathcal{E}(x_0^{1:M})$ by progressively sampling noise from a Gaussian distribution $\epsilon \sim \mathcal{N}(0, I)$ to produce noise z_T over **T** progressive timesteps. This could be given by the equation:

$$z_t^{1..M} = \sqrt{\overline{\alpha}_t}\, z_0^{1..M} + \sqrt{1-\overline{\alpha}_t}\, \epsilon_t^{1..M}, \tag{4}$$

where $\alpha_t \in (0, 1)$, and $\overline{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. The denoiser ϵ_θ is then trained by minimizing the reconstruction loss:

$$\mathbb{E}_{x_0^{1..M},\, y_t^{1..M},\, \epsilon_t^{1..M} \sim \mathcal{N}(\mathbf{0}, I)} \left\| \epsilon_t^{1..M} - \epsilon_\theta\left(z_t^{1..M}, t, y\right) \right\|_2, \tag{5}$$

where y is the input conditioning signal. This trained denoiser can then be used to generate a sequence of M images $I_{1...M}$ given a conditioning image $\mathcal{I}$ at the test time.

1.3. Training Details

Pseudo View Pre-Processing. To prepare the pseudo views, we generate 14 frames using MotionCtrl [3] in both the forward and backward directions and continue to progressively do so until paired pseudo views have been generated for all the frames corresponding to each particular scene in RealEstate-10k [4]. For the out-domain KITTI-v2 [5] dataset, since it follows a stereo format, we generate the pseudo views for the right camera following the standard protocol followed by existing works [6, 7] all of which reconstruct the scene based on views obtained from the left camera and test on novel views from the right camera. We keep the standard test resolution of 375×1242 and crop the outer 5% from all the images following the baseline [7–10] protocols.

LVDM Details. To denoise the pseudo views, we perform 50 denoising inference steps per image. We follow the same resolution of 256×384 in RealEstate-10K to ensure that the generated images are consistent with the rendered images from the 3D scene. We keep a FPS of 6 for all our experiments and set speed=1. We don't utilize the motion_bucket_id parameter since it is irrelevant for our use case.

1.4. MLLM for Object Tagging

In this work, we utilize the BLIP2-Flan-T5-XL [1] model for extracting both partially and fully visible objects. To ensure that the MLLM adheres to the task in hand, we leverage a one-shot in-context learning setup [11] which significantly enhances it's ability to detect objects. The prompting regimen which we followed for generating the object tags is described in Figure 1.

1.5. Qualitative Results

We present additional qualitative results on the RealEstate-10K [4] and KITTI-v2 [5] datasets shown in Figure 2. As it can be seen, `UAR-Scenes` shows robust performance across a wide variety of indoor and outdoor scenes. Further, `UAR-Scenes` is able to produce meaningful explanations outside of the input image's view owing to it's strong extrapolation capabilities. Hence, when combined with an existing 3D reconstruction pipeline, we can refine the coarse Gaussians and get more realistic and plausible renderings over a wide variety of real-world scenes.

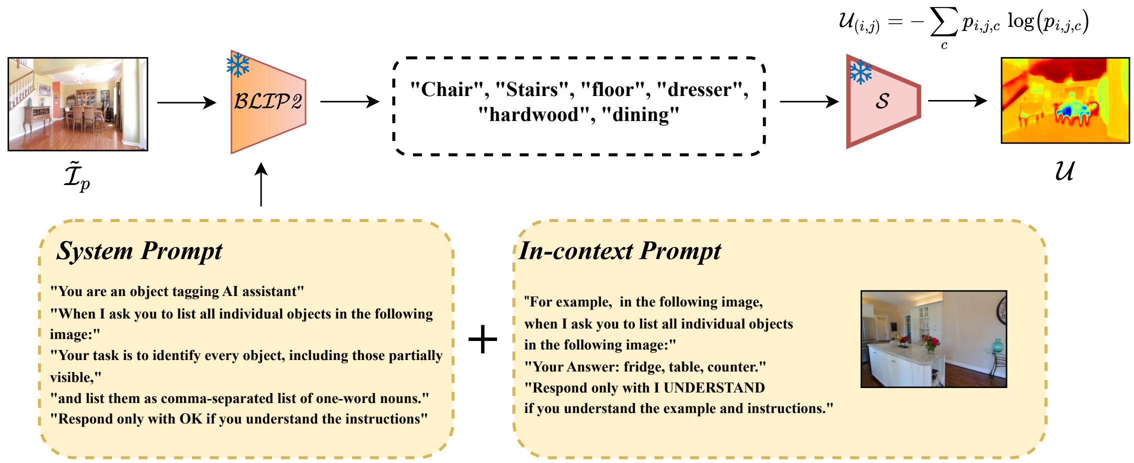

Figure 1. **Uncertainty Map Pipeline.** We pass the pseudo views to the MLLM [1] first using the one shot in-context learning setup as shown above. This gives the object tags which is then passed onto the open-vocabulary segmentation model LSeg [2]. We then compute the per-pixel entropy obtained from the segmentation maps to generate the corresponding uncertainty maps $\mathcal{U}$.

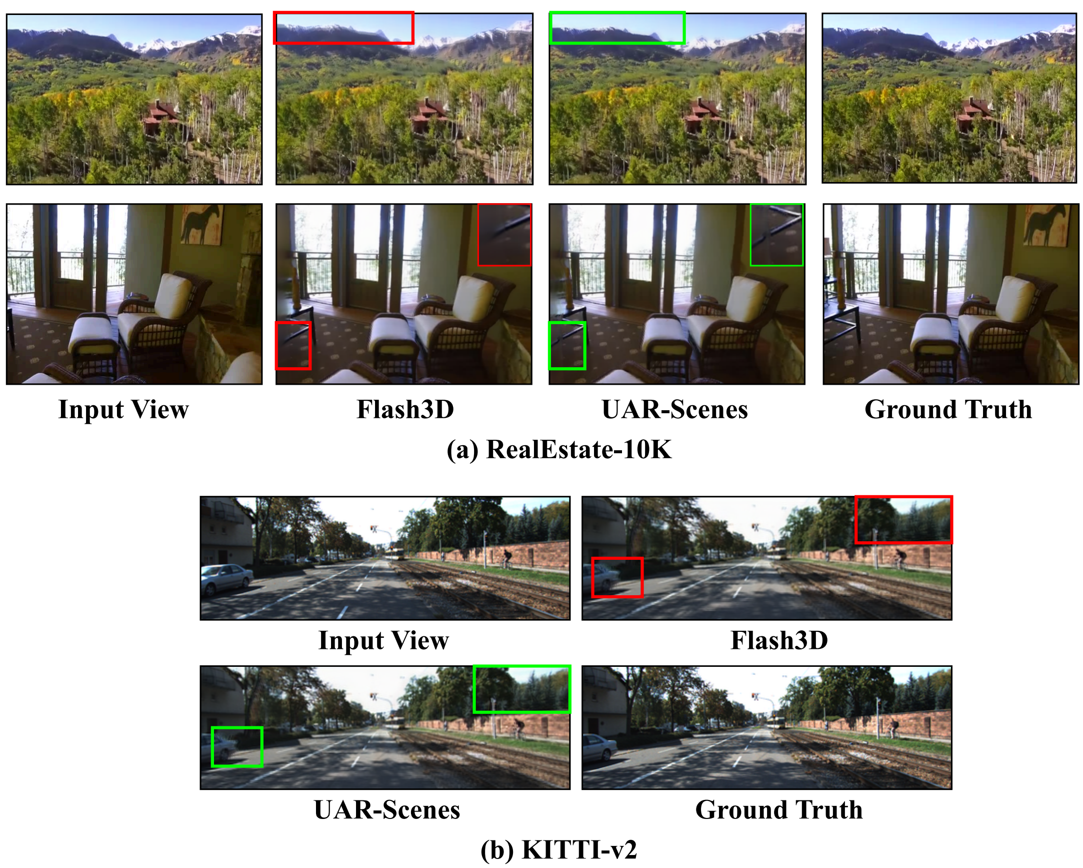

Figure 2. **Plausible Generation Results.** (a) `UAR-Scenes` is seamlessly able to adapt to indoor and outdoor scenes while preserving realistic and plausible quality in areas where Flash3D fails. In some cases as in the 2nd row, our method produces plausible explanations for regions outside of the input image's view but which may not align with the ground truth image. The FID metric is crucial to assess the effectiveness of our method in such cases.(b) `UAR-Scenes` similarly generalizes to KITTI as well showing robust performance in previously unseen scenarios.

Figure 3. **Additional Qualitative Results.** `UAR-Scenes` notably improves in those scenarios where the baseline (Flash3D) falters in all three tested cases (in-domain, out-domain & in-the-wild) as the camera moves further/rotates away from the source, i.e. unable to keep geometry & texture (rows 1, 2 & 5), with artifacts in unknown/occluded regions (row 3 & last row).

References

[1] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 1, 2

[2] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations*, 2022. 2

[3] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 1

[4] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM Transactions on Graphics (TOG)*, 37(4):1–12, 2018. 1

[5] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 1

[6] Felix Wimbauer, Nan Yang, Christian Rupprecht, and Daniel Cremers. Behind the scenes: Density fields for single view reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9076–9086, 2023. 1

[7] Stanislaw Szymanowicz, Eldar Insafutdinov, Chuanxia Zheng, Dylan Campbell, Joao F Henriques, Christian Rupprecht, and Andrea Vedaldi. Flash3d: Feed-forward generalisable 3d scene reconstruction from a single image. *arXiv preprint arXiv:2406.04343*, 2024. 1

[8] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European Conference on Computer Vision*, pages 370–386. Springer, 2024.

[9] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19457–19467, 2024.

[10] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16453–16463, 2025. 1

[11] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1