

## —Supplementary Material—

# TokenGS: Decoupling 3D Gaussian Prediction from Pixels with Learnable Tokens

### A. More Implementation Details

**Gaussian Token Details.** We use a patch size 8 for the decoder, so each Gaussian token decodes 64 Gaussians. We initialize Gaussian tokens with a standard deviation of 0.01. When increasing the number of Gaussian tokens or adding dynamic Gaussian tokens, we initialize the new tokens to existing tokens and apply a small random perturbation with standard deviation 0.01. To make sure most Gaussians are visible at initialization, we initialize the linear output projection layer with a small standard deviation of  $2e-3$  and initialize the Layer Scale layer in the decoder with  $1e-5$ .

**Training Details.** We apply a gradient clipping with a threshold of 1.0 to the gradients of the model. Weight decay of 0.05 is applied except for the bias terms and the Layer Normalization layers. A random flip augmentation is applied to the images and camera poses during training. The view sampling strategy follows DepthSplat [2]. We rescale the camera translations of RE10K, DL3DV, and Kubric by 0.25, 0.15, and 0.1 respectively to make the mean depths roughly 1. The 150K-iteration base model training takes 36 hours on 8 NVIDIA A100 GPUs and the 10K-iteration finetuning usually takes another 4 hours depending on the number of Gaussian tokens and the resolution.

**Baseline Details.** For comparison, we reproduced GS-LRM [3] and BTimer [1] following the training details provided in the original papers. We directly compare with the numbers provided in the original papers when available.

### B. More Qualitative Results

**Website.** We provide a website in the supplementary material for more qualitative results. The website includes the following sections:

- **DL3DV Results:** An interactive viewer for 6-view reconstruction results on DL3DV.
- **RE10K Results:** An interactive viewer to compare our model with GS-LRM [3] on 2-view reconstruction on RE10K.
- **Test-time training:** An interactive viewer comparing our feedforward reconstructions on 4-view DL3DV with versions tuned at token- and Gaussian-level.

- **Extrapolation:** A video that compares our model with GS-LRM [3] on scene extrapolation on RealEstate10K.
- **Dynamic Reconstruction:** A video that compares our model with BTimer [1] on dynamic reconstruction on Kubric.
- **Emergent Scene Flow:** A video that visualizes the emergent scene flow of our model on Kubric.

Due to CORS restrictions the website needs to be launched from a local server, e.g. by running `python -m http.server` in the root directory and navigating to `localhost:8000` (default).

### References

- [1] Hanxue Liang, Jiawei Ren, Ashkan Mirzaei, Antonio Torralba, Ziwei Liu, Igor Gilitschenski, Sanja Fidler, Cengiz Oztireli, Huan Ling, Zan Gojcic, et al. Feed-forward bullet-time reconstruction of dynamic scenes from monocular videos. *arXiv preprint arXiv:2412.03526*, 2024. 1
- [2] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depth-splat: Connecting gaussian splatting and depth. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16453–16463, 2025. 1
- [3] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. In *European Conference on Computer Vision*, pages 1–19. Springer, 2025. 1