GaussianDWM: 3D Gaussian Driving World Model for Scene Understanding and Multi-Modal Generation (Supplementary Material)

Tianchen Deng[1]*, Xuefeng Chen[2]*, Yi Chen[1]*, Qu Chen[3,4], Yuyao Xu[3,4], Lijin Yang[3,4], Le Xu[3,4], Yu Zhang[3,4], Bo Zhang[3,4], Wuxiong Huang[3,4], Hesheng Wang[1]
[1] Shanghai Jiao Tong University [2] Tsinghua University [3] MEGVII Technology [4] Mach Drive

A. Overview

In this supplementary material, we first provide additional details on constructing the 3DGS dataset from the large-scale nuScenes dataset, as well as the caption dataset used for trajectory prediction. We then describe the implementation and training details in Sec. C and Sec. D. Finally, in Sec. E, we present extensive experimental results across various datasets.

B. Dataset Details

To represent scenes from the NuScenes dataset [2] using 3D Gaussians, we construct the first large-scale 3D Gaussian dataset for outdoor urban environments, comprising 800 scenes derived from the original nuScenes sequences. Our goal is to build a dataset that enables generalizable 3D Gaussian–based scene understanding with LLMs, providing high-quality geometry and appearance representations suitable for downstream reasoning tasks.

Dataset Processing Many scenes contain frames with very low overlap or severe motion blur, making it difficult to obtain accurate 3DGS reconstructions. Therefore, we remove images or entire scenes whose reconstruction quality does not meet our standard. Since NuScenes dataset [2] provides high-quality LiDAR point clouds, we initialize each scene using aggregated LiDAR points, avoiding the large-scale inaccuracies that would arise from using COLMAP as in conventional 3DGS pipelines. For scenes with available depth supervision, we additionally apply a depth loss to improve geometric fidelity. After optimization, we further filter reconstructed scenes based on the PSNR metric before using them as inputs for our pre-training.

Language Label Collection and Training Our language label collection aims to establish 3D language paired data by associating each 3D Gaussian primitive with a rich language feature f_i. We build upon LangSplat [6] to construct a 3DGS language field. These embeddings are obtained from CLIP features, which inherit hierarchical semantics extracted via SAM [4]. We then follow the standard 3DGS rendering strategy, incorporating language information directly into the Gaussian primitives. As a collection of anisotropic 3D Gaussians, with each Gaussian $G(x)$ characterized by a mean $\mu \in \mathbb{R}^3$ and a covariance matrix Σ:

$$G(x) = \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right). \tag{1}$$

To optimize the parameters of 3D Gaussians, they are rendered into 2D image planes and a tile-based rasterizer is used to improve the rendering efficiency:

$$C(v) = \sum_{i\in\mathcal{N}} c_i \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j), \tag{2}$$

We also adopt the tilebased rasterizer to retain the rendering efficiency:

$$\boldsymbol{F}(v) = \sum_{i\in\mathcal{N}} f_i \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j) \tag{3}$$

where $\boldsymbol{F}(v)$ represents the language embedding rendered at pixel v, and $\alpha_i = o_i G_i^{2D}(v)$. Here o_i is the opacity of the i th Gaussian and $G_i^{2D}(\cdot)$ represents the function of the i-th Gaussian projected onto 2D.

For scene representation, each environment typically requires several million Gaussians to model its geometry and appearance. Since CLIP embeddings are high-dimensional features, directly learning the latent feature f_i for every Gaussian in the CLIP space would dramatically increase both memory consumption and computational cost—particularly because each f_i has a dimensionality of 512.

To further reduce memory consumption and improve efficiency, we introduce a scene-wise language autoencoder E, which maps the CLIP embeddings $\boldsymbol{F}(v) \in \mathbb{R}^D$ to $\boldsymbol{H}(v) = E(\boldsymbol{F}(v)) \in \mathbb{R}^d$, where $d \ll D$. We select

$d = 3, D = 512$. The autoencoder is trained with a reconstruction objective on the CLIP embeddings. We also learn a decoder Ψ to reconstruct CLIP feature. Our autoencoder can significantly decrease memory requirements while retaining semantic fidelity.

$$\mathcal{L}_{ae} = \sum_{t=1}^{T} d_{ae} \left(\Psi \left(E \left(\boldsymbol{F}(\boldsymbol{v}) \right) \right), \boldsymbol{F}(v) \right), \tag{4}$$

After training the autoencoder, we transform all CLIP embeddings into scene-specific latent features $\boldsymbol{H}(v)$. We let our 3D language Gaussians learn language embeddings in the scene-specific latent space instead of the CLIP latent space. Compared to directly modeling the 512-dimensional CLIP embeddings, we significantly reduced the memory cost by incorporating scene priors. We optimized the language embeddings with the objective:

$$\mathcal{L}_{\text{lang}} = \sum_{l \in \{s,p,w\}} \sum_{t=1}^{T} d_{\text{lang}} \left(\boldsymbol{F}_t^l(v), \boldsymbol{H}_t^l(v) \right) \tag{5}$$

where $d_{\text{lang}}()$ denotes the distance function used for our 3D Language Gaussians. To inject the 3D Gaussian tokens into the query text input, we reconstruct a dedicated QA dataset. Moreover, to align with the training paradigm of QwenVL, all 3D Gaussian token annotations are placed at the beginning of each instruction. The exact formatting protocol is illustrated as follow:


```
{
  "token": "fd8420396768425eabec9bdddf7e64b6",
  "scene_token": "e7ef871f77f44331aefdebc24ec034b7",
  "scene_idx": 2,
  "frame_idx": 0,
  "category": "2D_perception_infos",
  "task": "rdp",
  "conversations": [
    {
      "from": "human",
      "value": "based on <gauss>, Can you quantify the distance separating the <CAM_FRONT> <box>(14,219),(37,248)</box> from the ego car?"
    },
    {
      "from": "gpt",
      "value": "38.7 meters is the distance from the ego car to it"
    }
  ],
  "image": [
    "nuscenes/samples/CAM_FRONT/n015-2018-08-02-17-16-37+0800__CAM_FRONT__1533201470412460.jpg"
  ],
  "views": ["CAM_FRONT"],
  "gauss": [
    "gauss/output-full-6v/2_CAM_FRONT/langsplat_3/per_frame/00000.pth"
  ]
}
```


Trajectory Dataset Generation Our generative model supports both temporal and spatial scene generation. For spatial generation, text queries typically take the form "generate the view after shifting 2 m to the left." In this case, we first use the LLM to infer the target camera pose. Using the predicted pose, we project the point cloud to obtain a sparse condition, which serves as the low-level guidance for the diffusion model.

For temporal generation, queries are typically phrased as "predict the scene 1 second later." The LLM performs trajectory prediction and directly outputs the future camera pose. Based on this pose, we again apply point-cloud projection to generate the sparse condition used for temporal guidance.

To enable temporal scene generation, the LLM must possess the capability to predict future trajectories. Based on this requirement, we construct a trajectory-oriented QA dataset to supervise and enhance the model's trajectory reasoning ability. From each video clip, we extract a 10-frame sequence. The first 4 frames are provided to the model as the prompt input, while the remaining 6 frames are used as the prediction targets. All trajectories are transformed into the ego-coordinate system of the 5th frame. The exact formatting protocol is illustrated as follow:


```

    "conversations": [
        {
            "from": "human",
            "value": "There is last 4frames trajectory, [PT, [-7.45, 3.05, 0.16, 0.0, 0.0, -0.36, 0.93], [-5.75, 1.97, 0.11, 0.0, 0.0, -0.28, 0.96], [-4.04, 0.92, 0.09, 0.0, 0.0, -0.19, 0.98], [-2.15, 0.26, 0.04, 0.0, 0.0, -0.1, 1.0]]. Summarize the motion of the ego vehicle in this 6-frame clip"
        },
        {
            "from": "gpt",
            "value": "[PT, [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0], [2.47, 0.65, -0.06, 0.0, 0.0, 0.09, 1.0], [4.24, 2.32, -0.11, 0.0, 0.0, 0.12, 0.99], [5.5, 4.08, -0.15, 0.0, 0.0, 0.13, 0.99], [7.03, 6.45, -0.15, 0.0, 0.0, 0.13, 0.99], [8.66, 8.91, -0.17, -0.01, 0.0, 0.13, 0.99]]"
        }
    ]
```


C. Implementation Details

Our training pipeline consists of three stages. In the first stage, we train the Gaussian tokenizer, projector, and the proposed sampling strategy independently. We then inte-

grate these components with the LLM and perform joint fine-tuning. For the LLM, we adopt a LoRA-based fine-tuning strategy. All models in this stage are trained using 16 NVIDIA A100 GPUs. In the second stage, we train the multi-modal generation module. We start by training a low-resolution (224×400) RGB video generation model, extend it to low-resolution RGB-D generation, and finally refine it into a high-resolution (424×800) RGB-D video generator using a mixed-frame-length strategy. The model is optimized using simulation-free rectified flow and a v-prediction loss [3]. In the final stage, we perform end-to-end joint optimization over all components to ensure consistency between scene understanding and scene generation.

3D Gaussian Scene Loading Strategy. For each processed 3D Gaussian scene, the loading procedure depends on the type of Gaussian features associated with the given QA sample. Since each QA item focuses on a specific keyframe while the number of available camera views varies across scenes, we adopt the following unified strategy:

- **Surround-view Gaussian features.** When the QA sample requires panoramic information, we load the Gaussian features from all six surrounding views. The Gaussian tokens from all views are concatenated along the token dimension to form a unified set of scene tokens.
- **Single-view Gaussian features.** When the QA sample is associated with a specific camera view, we load only the Gaussian features of that view, without any concatenation.

This design ensures flexibility for different QA formats while maintaining consistent integration of 3D Gaussian scene information into the language model input.

Training Schedule and Parameter Efficiency. During training, the number of epochs is adapted to each task, as the difficulty and convergence behavior differ across region description and perception (RDP), 2D visual grounding (2DVG), 3D visual grounding (3DVG), and planning. Specifically, each task is trained for a task-dependent number of epochs to ensure stable convergence.

In the first-stage training, the number of trainable parameters accounts for only **0.45%** of the total model parameters, focusing mainly on the Gaussian tokenizer, projector, and sampling modules. During the LoRA-based fine-tuning stage, the trainable parameters account for merely **0.79%** of the total model parameters. This demonstrates that our framework achieves strong performance while maintaining high parameter efficiency.

World Knowledge Injection. In our framework, world knowledge is injected into the generative model through a text-conditioned encoder pathway. Specifically, we employ a pretrained language model to convert a natural-language caption into a sequence of semantic tokens, which serve as high-level conditioning signals for the diffusion UNet. Given an input caption, we first tokenize it and encode the sequence using a CLIP-based text encoder. This produces a set of contextualized text embeddings $E_{\text{text}} \in \mathbb{R}^{L \times D}$ where each token represents a semantic unit such as an object category, attribute, or action. Unlike pooled language vectors, which collapse all semantics into a single global descriptor, token-level embeddings preserve fine-grained, word-level structure and allow the diffusion model to selectively attend to relevant parts of the caption. To integrate text knowledge into the generative process, we concatenate the text embeddings with the image-embedding tokens extracted from a reference frame. We further append a learnable type embedding to each token, enabling the model to distinguish between visual and linguistic information during cross-attention. The resulting combined sequence $E_{\text{enc}} = [E_{\text{img}}; E_{\text{text}}] + E_{\text{type}}$ is passed as the encoder-hidden-states to the diffusion UNet, which uses cross-attention at multiple layers to fuse world knowledge with spatial features. Through this mechanism, the UNet learns to associate local visual structures with global textual semantics, enabling the model to leverage descriptions such as object identity, scene context, and commonsense priors to guide novel-view synthesis. This fusion pathway allows the generative model to incorporate externally grounded world knowledge—captured implicitly by large-scale text–image pre-training of the language encoder—without modifying the underlying UNet architecture. As a result, the model benefits from both low-level geometric cues (RGB/depth latents) and high-level semantic reasoning (text-conditioned cross-attention), improving structural consistency, scene understanding, and semantic controllability during view synthesis.

D. Ablation Study

We additionally provide further ablation studies in Tab. 1, including experiments on the effectiveness of the cross-attention–based sampling module. We also analyse on how world knowledge influences the performance of the generative model in Tab. 3.

E. Extensive Experiments

OmniDrive dataset [7] Scene Understanding Results. We further evaluate our model on the OmniDrive dataset, comparing against strong baselines including GPT-4o [1], LLaVA [5], OmniDrive [7], and Hermes [8]. Our GaussianDWM consistently achieves state-of-the-art performance across all evaluation metrics. These results again confirm the effectiveness of our 3D Gaussian scene representation and task-aware sampling strategy in enhancing LLM-based scene understanding under complex driving scenarios.

We also provide additional qualitative results, including more visualizations of the QA outputs and scene generation

Sampling	Gaussian Tokens	2D RD & Pre ↑			2D VG↑			3D VG↑			Plan↑	Avg.↑
		BLEU	Rouge_L	CIDEr	mAP	F1	MIoU	Pr	mAP	F1	Acc	
Top-k + Uniform+Cross	128	68.27	80.95	78.73	34.79	40.3	71.79	**51.41**	54.01	32.81	68.43	58.15
Top-k + Uniform+Cross	512	**68.37**	**81.00**	**79.22**	**34.95**	**40.39**	**71.85**	51.30	**54.07**	**32.87**	**68.52**	**58.26**
Top-k + Uniform	4096	66.17	79.07	77.06	33.89	39.31	71.37	51.16	52.87	32.05	66.27	56.92
Top-k + Uniform+Cross	4096	68.20	80.90	78.93	34.50	40.00	71.57	51.20	53.59	32.53	65.60	57.70

Table 1. **Ablation study on Sampling Strategies and Gaussian Token Counts.** We investigate the effectiveness of the proposed Cross-Attention sampling compared to the Uniform baseline. The model with 512 tokens using our strategy achieves the optimal performance, surpassing the dense baseline using 4096 tokens. This indicates that our learned sampling effectively captures critical information with significantly fewer tokens, validating its efficiency.

Model	Reference	# LLM Params	Understanding↑		
			METEOR	ROUGE_L	CIDEr
Only Understanding					
GPT-4o [1]	-	-	-	0.223	0.244
LLaVA-OV [5]	arXiv 24	7B	-	0.221	0.284
OmniDrive [7]	CVPR 25	7B	0.380	0.326	0.686
OmniDrive-2D [7]	CVPR 25	7B	0.383	0.325	0.671
OmniDrive-BEV [7]	CVPR 25	7B	0.356	0.278	0.595
Unified Understanding and Generation					
Separated unification	-	1.8B	0.384	0.327	**0.745**
HERMES [8]	ICCV 25	1.8B	0.384	0.327	0.741
GaussianDWM	-	8B	**0.395**	**0.341**	0.712

Table 2. **OmniDrive dataset [7] Scene Understanding Results.** We further evaluate our model on the OmniDrive dataset, comparing against strong baselines including GPT-4o [1], LLaVA [5], OmniDrive [7], and Hermes [8]. Our GaussianDWM consistently achieves state-of-the-art performance across all evaluation metrics. These results again confirm the effectiveness of our 3D Gaussian scene representation and task-aware sampling strategy in enhancing LLM-based scene understanding under complex driving scenarios.

results.

World Knowledge	Gaussian	shift ± 1m		shift ± 2m		shift ± 4m	
		FID ↓	FVD ↓	FID ↓	FVD ↓	FID ↓	FVD ↓
✗	✗	11.44	42.22	14.75	71.98	21.79	**137.31**
✓	✗	11.30	41.63	13.96	**70.45**	19.02	151.19
✓	✓	**11.28**	**41.62**	**13.95**	70.70	**18.91**	150.05

Table 3. **Ablation on World Knowledge and Gaussian**. We further evaluate the impact of world knowledge by comparing generation model with and without world knowledge, and by examining how Gaussian influences both world knowledge representation and final generation. The improvement brought by world knowledge becomes increasingly significant under larger view shifts. All evaluations are conducted on a subset of 30 scenes from the nuScenes dataset.

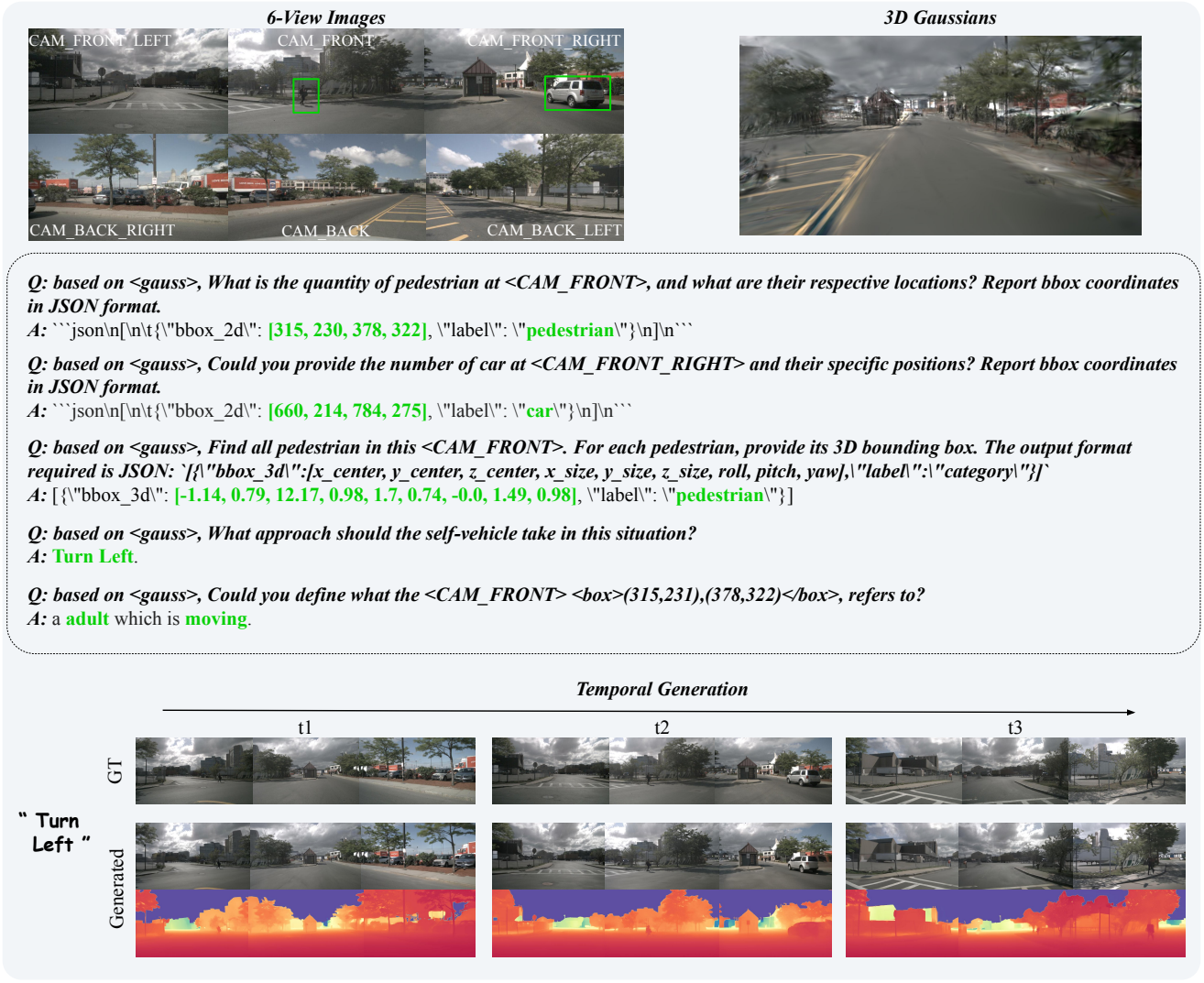

Figure 1. **Qualitative results for scene understanding and scene generation**. From top to bottom, we present the multi-view input of the current scene and the 3D Gaussian ellipsoids, the scene understanding results, and the spatial and temporal scene generation results.

Figure 2. **Qualitative results in rainy scene for understanding and generation**.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 3, 4

[2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 1

[3] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024. 3

[4] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023. 1

[5] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306, 2024. 3, 4

[6] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024. 1

[7] Shihao Wang, Zhiding Yu, Xiaohui Jiang, Shiyi Lan, Min Shi, Nadine Chang, Jan Kautz, Ying Li, and Jose M Alvarez. Omnidrive: A holistic vision-language dataset for autonomous driving with counterfactual reasoning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22442–22452, 2025. 3, 4

[8] Xin Zhou, Dingkang Liang, Sifan Tu, Xiwu Chen, Yikang Ding, Dingyuan Zhang, Feiyang Tan, Hengshuang Zhao, and Xiang Bai. Hermes: A unified self-driving world model for simultaneous 3d scene understanding and generation. *arXiv preprint arXiv:2501.14729*, 2025. 3, 4