Language Embedded 3D Gaussians for Open-Vocabulary Scene Understanding

Supplementary Material

We provide more details in this supplementary document, including load balancing loss (Sec. 7), inference strategy (Sec. 8), implementation details (Sec. 9), dataset details (Sec. 10) and more Results (Sec. 11).

7. Load Balancing Loss

To maximize utilization of the optimized feature space and avert quantization collapse, we introduce a load balancing loss, inspired by Switch Transformer [13]. When quantizing K features, the utilization ratio of each feature in $\mathcal{S}$ is calculated as:

$$m_i = \text{argmax}_j(\mathcal{D}(\mathbf{F}_i, \mathbf{f}_j)), \text{ where } \mathbf{f}_j \in \mathcal{S}, \tag{16}$$

$$\mathbf{r} = \frac{\sum_i^K \text{onehot}(m_i)}{K}, \tag{17}$$

where $\mathbf{r} \in \mathbb{R}^N$. We compute the mean selection probability for each feature over K quantizations:

$$\mathcal{D}(\mathbf{F}_i, \mathcal{S}) = [\mathcal{D}(\mathbf{F}_i, \mathbf{f}_1), \mathcal{D}(\mathbf{F}_i, \mathbf{f}_2), \ldots, \mathcal{D}(\mathbf{F}_i, \mathbf{f}_N)], \tag{18}$$

$$\mathbf{p} = \frac{\sum_i^K \text{Softmax}(\mathcal{D}(\mathbf{F}_i, \mathcal{S}))}{K}, \tag{19}$$

where $\mathcal{D}(\mathbf{F}_i, \mathcal{S}) \in \mathbb{R}^N$ and $\mathbf{p} \in \mathbb{R}^N$. The load balancing loss is then computed by the element-wise multiplication of $\mathbf{r}$ and $\mathbf{p}$, followed by their aggregation:

$$\mathcal{L}_{lb} = \sum^N (\mathbf{r} \circ \mathbf{p}), \tag{20}$$

where $\circ$ denotes the element-wise product.

8. Inference Strategy

In the inference stage, rasterization and alpha blending are employed to project the compact semantic features of 3D Gaussians into a 2D feature map. This feature map is then converted into the distribution of semantic indices using a trained MLP decoder and softmax activation, expressed as:

$$\mathcal{M}_{\text{infer}} = \text{Softmax}(D(R_s(\mathcal{G}; p_{\text{cam}}))), \tag{21}$$

where $R_s(\mathcal{G}; p_{\text{cam}}) \in \mathbb{R}^{H \times W \times d_s}$ denotes the rendered semantic features from a set of 3D Gaussians $\mathcal{G}$, as observed from the camera pose p_{cam}. Here, D symbolizes the trained MLP decoder of semantic features on 3D Gaussians. The result is a language feature index distribution $\mathcal{M}_{\text{infer}} \in \mathbb{R}^{H \times W \times N}$, where H and W represent the image's height and width, respectively. We finally acquire the language feature map, by multiplying $\mathcal{M}_{\text{infer}}$ with the quantized language features matrix $\mathbf{S} \in \mathbb{R}^{N \times d}$:

$$\mathcal{F} = \mathcal{M}_{\text{infer}}\mathbf{S}, \tag{22}$$

where $\mathcal{F} \in \mathbb{R}^{H \times W \times d}$ denotes the language feature map derived from the 3D Gaussians $\mathcal{G}$ observed from the camera pose p_{cam}. Figure 7 depicts the inference pipeline for generating language feature maps from language-embedded 3D Gaussians.

Utilizing the provided text prompt, we identify objects within the 3D scene by computing the relevance map of $\mathcal{F}$ in accordance with LERF [21].

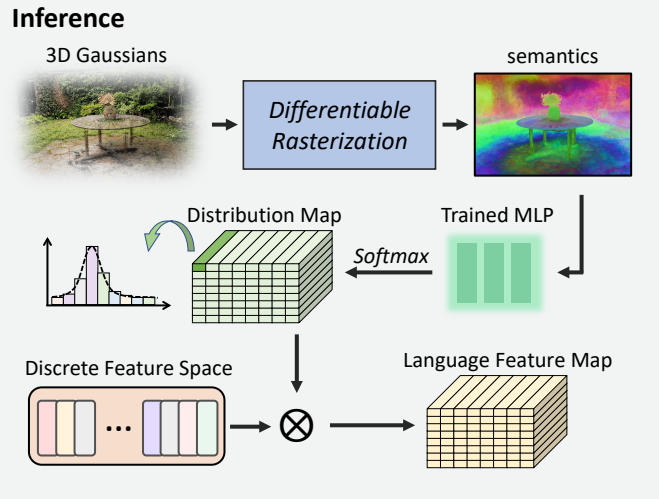

Figure 7. The inference pipeline for language feature maps from optimized 3D Gaussians. Each element in the distribution map $\mathcal{M}_{\text{infer}}$ represents a probability distribution of features in the Discrete Feature Space $\mathcal{S}$. We compute the corresponding language feature map $\mathcal{F}$ based on $\mathcal{S}$ utilizing these distributions.

	Layer	Config	Out Size
Input	-	-	$8 \times H \times W$
C1	Conv+ReLU	$128 \times 1 \times 1 / 1$	$128 \times H \times W$
C2	Conv+ReLU	$256 \times 1 \times 1 / 1$	$256 \times H \times W$
C3	Conv	$N \times 1 \times 1 / 1$	$N \times H \times W$

Table 3. Details of our semantic feature decoder D. In a layer characterized by $c \times w \times w/s$, c represents the number of filters, $w \times w$ indicates the filter size, and s denotes the stride size. The output dimensionality is expressed in terms of channel $\times$ height $\times$ width. The dimension of semantic feature on 3D Gaussians is 8 and N represents the size of discrete language feature space $\mathcal{S}$.

	Layer	Config	Out Size
PE	Positional Encoding	0	3
F1	Full-connected+ReLU	128	128
F2	Full-connected+ReLU	128	128
F3	Full-connected+ReLU	128	128
F4	Full-connected	8	8

Table 4. Details of the PE and MLP in the adaptive spatial smoothing. The input is the position of 3D Gaussian and the output is smoothed semantic feature s_{MLP}. The configure of PE layer represents the frequency of positional encoding.

9. Implementation Details

In Tab. 3 and Tab. 4, we present the implementation details of the semantic feature decoder D and the PE and MLP in the adaptive spatial smoothing, respectively. Furthermore, for the size N of the discrete language feature space $\mathcal{S}$, we set $N = 32$ for the "kitchen" scene, $N = 64$ for the "bonsai" scene, and $N = 128$ for other scenes. N, as a hyperparameter, controls the capacity of semantic information in the discrete language feature space $\mathcal{S}$ and can be adjusted according to the richness of semantic information in the scene.

To justify the performance increment, we elaborate the metrics in Tab. 1. Memory cost includes host and device memory usage for optimization programs, including 3D Gaussians, discrete feature maps and two MLPs in our method. Disk cost relates to language features, involving discrete feature space, maps and MLPs. Quantization reduces language feature cost, minimizing overall memory and disk expenses. Training time for all methods involves MLPs or 3D Gaussian optimization with language embedding. FPS measures rendering time only, while querying time remains the same across methods.

10. Datasets

To concurrently assess the quality of visual and semantic embeddings, six scenes from the Mip-NeRF360 dataset [3] are chosen for quantitative and qualitative evaluation. The 'Stump' scene is excluded due to its insufficient semantic content. The evaluation set of each scene is manually annotated with segmentation maps, which are created for the primary objects in each scene. The text prompts corresponding to these annotated objects are listed in Tab. 6. Additionally, segmentation masks for some objects in our dataset are illustrated in Fig. 8.

Although 13 scenes are included in the LeRF [21] dataset, quantitative evaluations are only carried out on five of them (waldo kitchen, bouquet, ramen, teatime, and figurines) in LeRF. We conduct an additional experiment in line with LeRF on those scenes for quantitative evaluation against LeRF. Note that the original LeRF annotations for object localization are simply rectangular boxes, which may lead to performance saturation and are insufficient for complex metrics, hence the ground truth segmentation masks are manually labeled based on the text labels from the LeRF dataset.

		PSNR↑	mPA↑	mP↑	mIoU↑	mAP↑	LA ↑
LeRF	bouquet	21.374	0.832	0.298	0.282	0.595	0.500
	figurines	19.641	0.933	0.370	0.328	0.721	0.866
	kitchen	18.740	0.717	0.259	0.231	0.592	0.676
	ramen	21.793	0.569	0.193	0.182	0.508	0.552
	teatime	21.196	0.785	0.288	0.282	0.662	0.683
	Overall	20.549	0.768	0.282	0.262	0.582	0.618
Ours	bouquet	23.175	0.911	0.555	0.396	0.628	0.673
	figurines	21.939	0.950	0.517	0.317	0.567	0.767
	kitchen	23.205	0.868	0.441	0.251	0.541	0.523
	ramen	24.804	0.923	0.434	0.385	0.736	0.737
	teatime	25.522	0.889	0.504	0.309	0.621	0.683
	Overall	**23.812**	**0.909**	**0.490**	**0.332**	**0.619**	**0.677**

Table 5. Quantitative comparisons of LeRF and ours on the LeRF dataset. LA is the localization accuracy in LeRF.

11. More Results

Further qualitative results on the Mip-NeRF360 dataset [3] are presented to illustrate the comparison of visual quality (Fig. 10), the evaluation of novel view synthesis and query accuracy (Fig. 9), and the exploration of open-vocabulary queries (Fig. 11). The quantitative evaluations on the LeRF dataset [21] demonstrate the superiority of our method over the LeRF. Full results are presented in the Tab. 5.

Scene	Positive Words
bicycle	green grass, white bicycle, tire, bench, asphalt ground, silver oak tree
bonsai	piano keyboard, bicycle, purple table cloth, black stool, plastic bonsai tree, dark grey patterned carpet
counter	jar of coconut oil, fruit oranges, onions, plants, blue oven gloves, wood rolling pin, free range eggs box, stable bread, garofalo pasta, napolina tomatoes, gold ripple baking pan
garden	football, wood round table, green grass, wood pot, elderflower, green plant, bricks wall, windows, stone ground
kitchen	LEGO Technic 856 Bulldozer, basket weave cloth, wood plat, old pink striped cloth, red oven gloves
room	blue grey chair, curtain, brown shoes, books, windows, door, piano keyboard, wood floor, wine glasses and bottles, yucca plant, deep green carpets

Table 6. Text prompts used for evaluating quality and accuracy of open-vocabulary query.

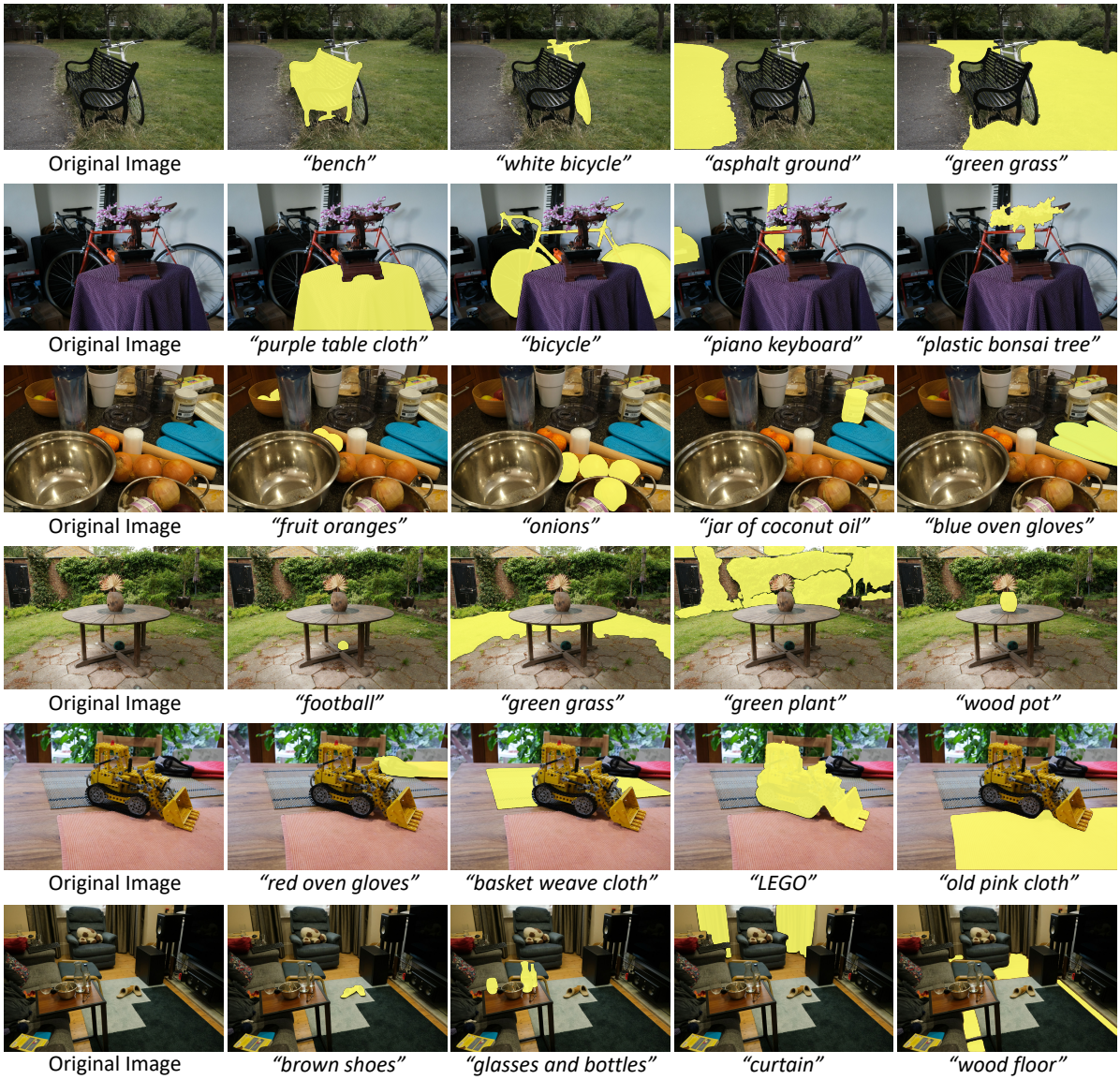

Figure 8. Ground truth segmentation masks for some objects in our dataset. In each scene, we select primary, unambiguous objects for semantic annotation. This includes both large and small objects, as well as challenging entities with complex geometric structures or transparent and translucent properties, such as bicycles, windows, and water glasses.

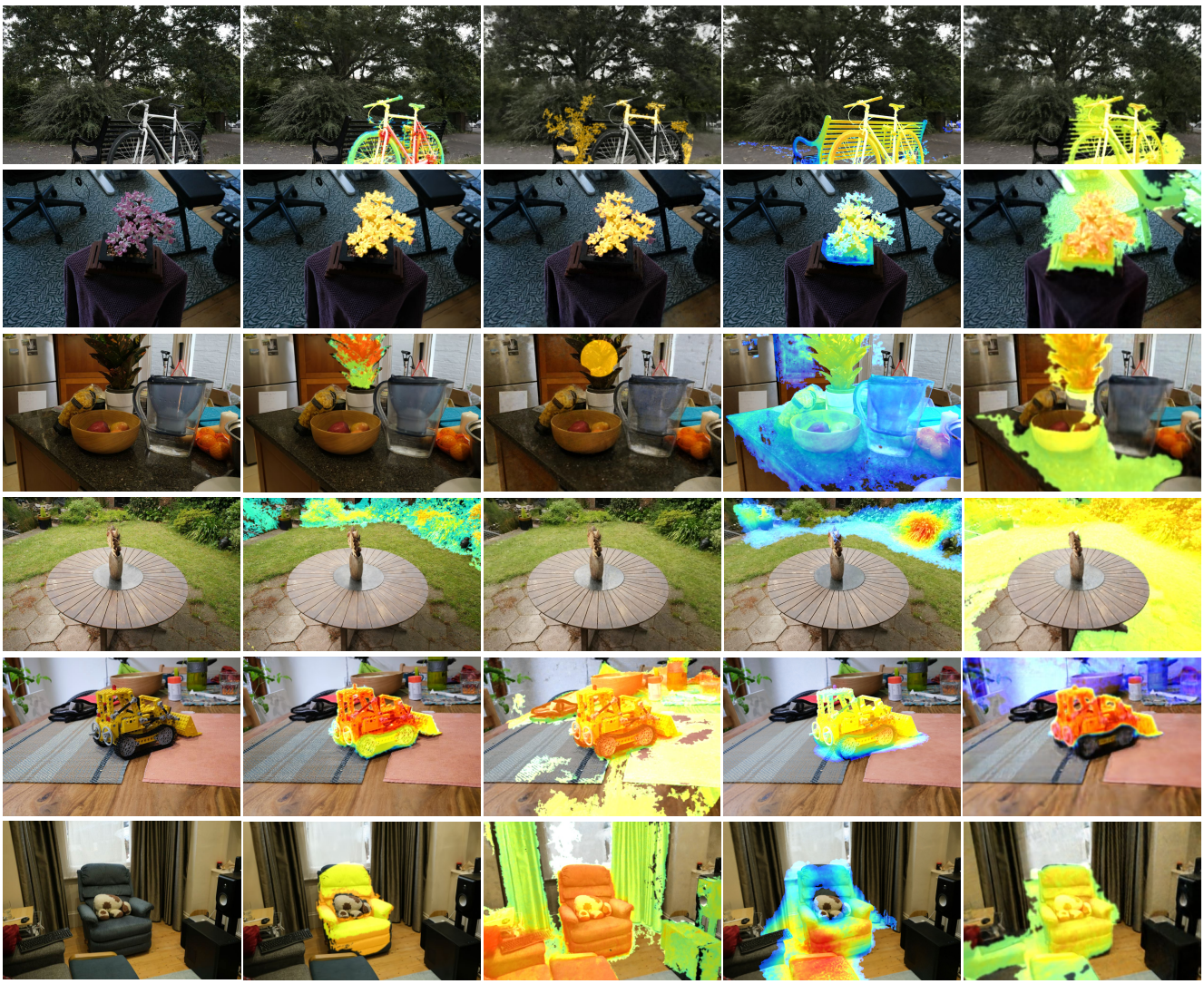

Figure 9. Comparison of novel view synthesis quality and open-vocabulary query accuracy. Left to right: Ground truth novel view synthesis, novel view images with relevance visualization from our method, DFF [22], LeRF [21], and 3DOVS [26]. Top to bottom: Query words “white bicycle”, “bonsai”, “plants”, “green plant”, “LEGO Technic 856 Bulldozer”, and “blue grey chair”.

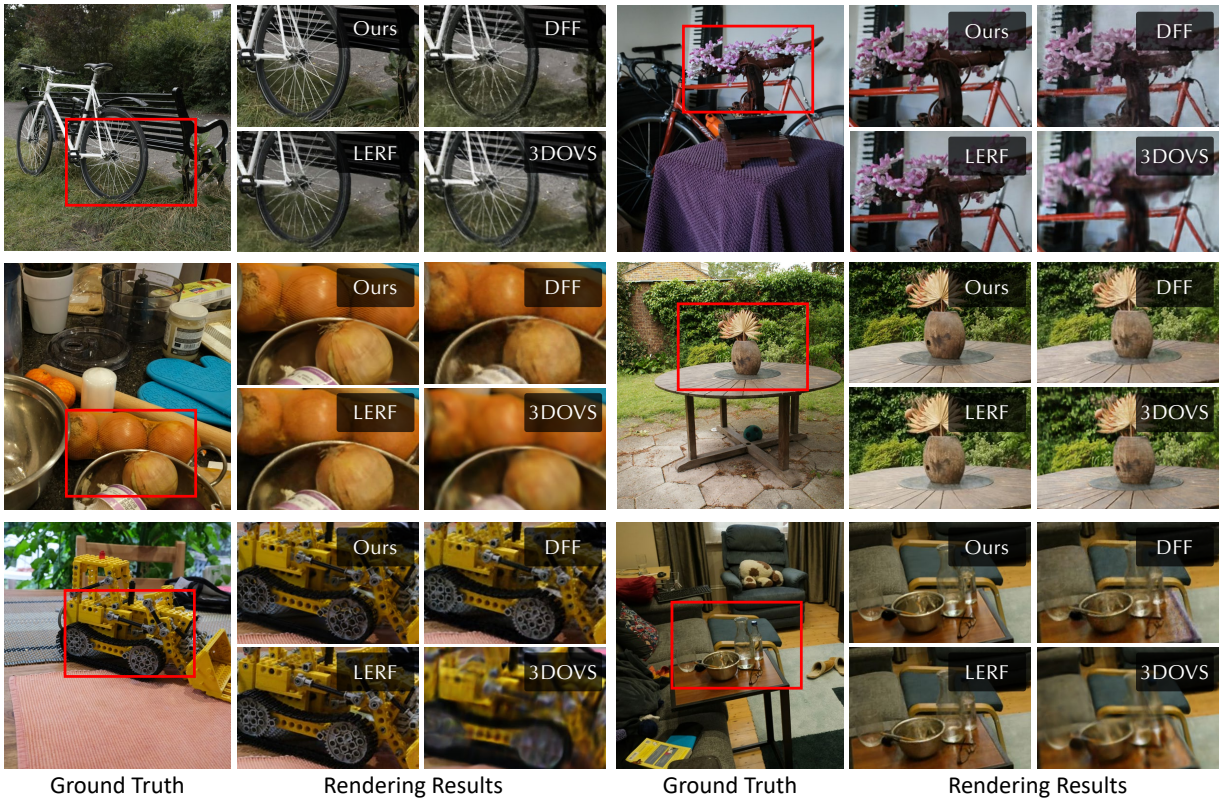

Figure 10. Comparison of the quality of novel view synthesis. Even with dense language features embedded into the 3D Gaussians, our method still only requires a reasonable amount of memory, thus allowing a massive amount of points to be rendered and optimized at the same time, achieving the best visual quality with more details compared to other methods.

Figure 11. Examples of various open-vocabulary queries. Our approach enables accurate open-vocabulary queries using a diverse class of word types, including but not limited to, visual attributes, general terms, materials, olfactory properties, and related actions.