The Visual Computer (2026) 42:38
https://doi.org/10.1007/s00371-025-04219-7

RESEARCH

Enhancing open-vocabulary scene understanding via push–pull alignment in gaussian splatting

Tong Chen[1] · Shengjia Liang[1] · Yuan Xiong[2] · Qiang Zhou[1] · Qichuan Geng[3] · Zhong Zhou[1,4]

Received: 23 April 2025 / Accepted: 7 November 2025 / Published online: 10 December 2025

Abstract

Open-vocabulary scene understanding based on 3D Gaussian Splatting (3DGS) has shown promising potential for applications such as embodied agents and object localization. By integrating open-vocabulary embeddings into spatial 3D gaussians, these models enable a more comprehensive understanding of scenes. However, existing methods often suffer from misalignment due to the gap between RGB and language modalities, leading to incorrect interpretations of similar-looking objects. To address this issue, we propose a cross-modal integration approach that aligns multiple representations through spatial gaussian positioning. We introduce Push-Pull alignment in Gaussian Splatting(PPGS), a novel bimodal framework that bridges RGB and language modalities through cohesive representation fields. Leveraging the illumination-invariant properties of language embeddings, we design the bridge module, which uses the geometrically-grounded positions for the gaussians as a direct bridge between the two modalities. This module significantly enhances cross-modal alignment, improves high-fidelity rendering, and ensures accurate language feature embeddings. Furthermore, our framework dynamically adjusts gradients based on the distinct optimization requirements of RGB and language during joint learning, ensuring stable and efficient convergence. Comprehensive experiments demonstrate that PPGS achieves superior language query accuracy and enhanced visual quality compared to existing language-embedded representations, with Intersection over Union (mIoU) increasing by 6% and Peak Signal-to-Noise Ratio (PSNR) showing gains over mainstream methods, all within only 50% of the training time.

Keywords Gaussian splatting · Open-vocabulary · Scene understanding · Novel view synthesis · Neural rendering

1 Introduction

Reconstructing 3D scenes and open-vocabulary scene understanding are challenging tasks in computer vision [1–6], with massive applications in embodied intelligence [7–9] and augmented/virtual reality [10–12]. Recent advancements in 3D Gaussian Splatting (3DGS) [13] have greatly enhanced 3D scene reconstruction and novel view synthesis from multi-view images. By optimizing the positions, rotations, scales, and spherical harmonic coefficients of gaussians, 3DGS achieves fast training time and rendering speeds. Considering the multi-view inconsistencies and the severe noise in reconstruction caused by spatial misplacement in volumetric 3D Gaussian representations, 2D Gaussian Splatting (2DGS) [14] replaces 3D gaussians with 2D Gaussians. This approach ensures precise spatial positioning of gaussians, leading to a more accurate reconstruction of the scene and significantly reduced noise.

Recent advances have explored leveraging language as a powerful modality for efficient understanding and interaction with 3D scenes. Language inherent ability to capture high-level semantics enables applications like 3D scene understanding and editing[15]. However, a primary challenge is the scarcity of datasets that provide diverse language annotations for 3D scenes at a large scale. To overcome this, recent methods[16–19] create language-embedded representations by distilling latent features from pre-trained Vision Foundation Models (VFMs) such as CLIP, SAM, and DINO[20–23]. These distilled features from multi-view 2D images are then integrated into the 3D scene for open-vocabulary queries.

✉ Zhong Zhou
zz@buaa.edu.cn

Tong Chen
tchen@buaa.edu.cn

1 State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China

2 SUN Yat-Sen University, Shenzhen 518107, China

3 Capital Normal University, Beijing 100048, China

4 Zhongguancun Laboratory, Beijing, China

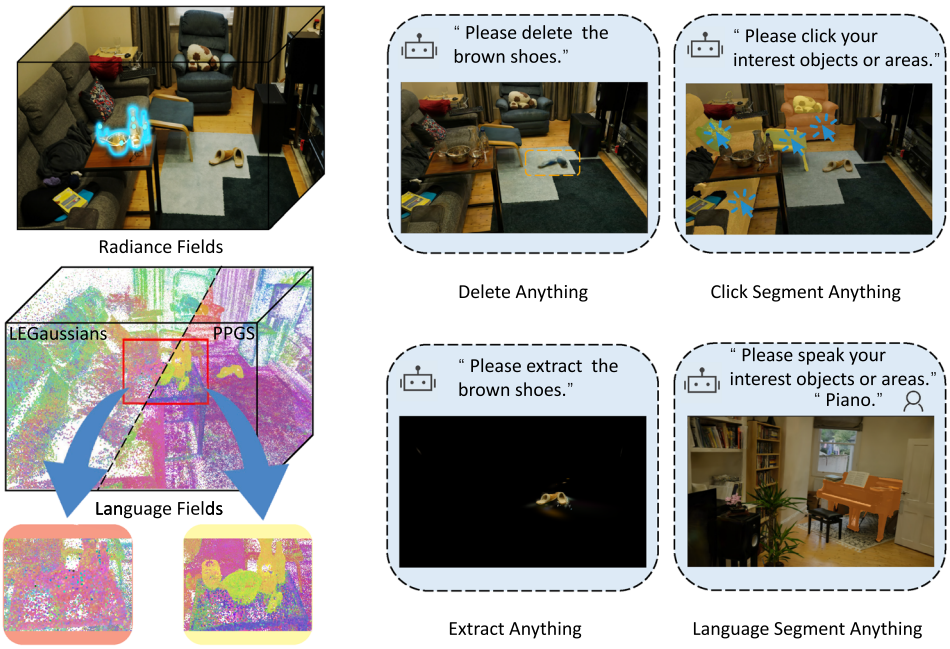

Fig. 1 Visualizing the challenge of cross-moda alignment and the interactive applications by the Push–Pull Gaussian Splatting (PPGS). The left panel illustrates the problem of noisy and semantically misaligned feature fields. The right panel demonstrates the versatile downstream tasks, such as including language-guided object deletion and extraction, and prompt-based segmentation. The implementation details for these interactive tasks are discussed in Section 4.2.5

Previous methods mainly perform open-vocabulary scene understanding in a two-stage manner. In the first stage, the positions, shapes, and opacities of the 3D gaussians are optimized solely for photometric consistency. In the second stage, language features are embedded onto these gaussian positions, which are fixed after being optimized purely for visual fidelity. This process incurs a fundamental misalignment, as the semantic boundaries demanded by language cannot conform to a spatial layout optimized purely for visual fidelity. This problem is exemplified in Fig. 1, when reconstructing visually ambiguous regions, the optimizer might place Gaussians in ambiguous positions, blurring the boundary between them (the highlights shared by 'a stainless steel bowl' and 'a glass cup'). For instance, Feature-3dgs [24] attributes the issue to inherent 'merging noisy or inconsistent 2D labels' between the RGB and feature domains. An alternative paradigm, exemplified by LEGaussians [18], reconstructs both fields simultaneously. This approach allows the Gaussian positions to be influenced by both modalities, offering the potential to mitigate the misalignment issues of two-stage methods. However, this simultaneous approach introduces a cross-modal conflict in the optimization of individual Gaussians. During joint optimization, a single gaussian receives competing gradients: the RGB loss might pull it toward a position to better reconstruct a visual texture, while the language loss simultaneously pulls it toward a different position to better fit a semantic category. This optimization conflict often results in the formation of "noisy", which are individual Gaussians settling in spatial locations unable to satisfy either objective. Furthermore, this simultaneous training approach introduces the additional challenge of an imbalance problem. Previous methods struggle to achieve satisfactory reconstruction quality when concurrently learning RGB reconstruction (regression) and language reconstruction (classification). Simply combining the loss functions of these two tasks often degrades performance in both areas. However, as they inherently share the same spatial distribution, it is crucial to design a loss function with adaptive weights that dynamically balance both objectives while considering the overall optimization landscape.

In this paper, we propose PPGS, a bimodal framework that reconstructs RGB and language fields through cross-modal alignment, enabling precise open-vocabulary scene understanding. To address the misalignment overlooked by previous methods, we propose the bridge module. To establish a robust interface for cross-modal integration, we adopt 2D gaussians and extend their primitives with latent language parameters, enabling the simultaneous optimization of both fields. This foundational design provides the precise spatial positions that our bridge module leverages, which in turn introduces a push–pull mechanism to effectively improve alignment, reduce noise, and enhance reconstruction quality and query performance. Additionally, the imbalance in weight optimization between the two fields hinders optimal performance, as the bridging process requires maintaining consistency across both modalities. We adopt a joint training strategy that dynamically balances the learning of RGB and language fields, enhancing their coordination and overall performance.

In summary, our contributions include:

- We propose PPGS, a bimodal framework that leverages the geometric prior from 2D gaussian to enable robust cross-modal alignment, and adopts a joint training strategy with dynamic gradient balancing to handle RGB and language modality integration.
- We propose the bridge module, a push–pull mechanism that synchronizes RGB and language modalities through shared Gaussian spatial positions, enabling bidirectional optimization to enhance language query accuracy and maintain rendering quality.
- Extensive experiments demonstrate that our method sets a new state-of-the-art in open-vocabulary scene understanding. It achieves significant improvements in both visual reconstruction quality and open-vocabulary query performance, which is supported by enhanced feature alignment and spatial consistency.

2 Related work

2.1 Neural rendering and Gaussian splatting

NeRF [25] has demonstrated superior performance in novel view synthesis compared to traditional methods [26–30]. However, NeRF and other related methods [31, 32] require massive computational resources and long inference time. 3D Gaussian Splatting (3DGS) [13] has been proposed to improve reconstruction quality and reduce training time. Different from NeRF, 3DGS reconstructs with a collection of 3D Gaussians in place of differentiable volume rendering. Hence, faster rasterization is realized compared to NeRF-based [33–37] methods. Subsequent works [38–43] have been dedicated to achieving even higher fidelity and shorter training time on the task of novel view synthesis. For instance, SuGaR [44] compresses 3D Gaussians from meshes and 3D point clouds. The strategy of constraining 3DGS with surface reconstruction improves the accuracy and efficiency of the approximation, enabling better handling of surface details. However, the absence of geometric constraints in SuGaR produces much gaussians noise. Therefore, 2D Gaussian is proposed to replace 3D Gaussian, and this method is named “2D Gaussian Splatting" [14]. 2D Gaussian Splatting (2DGS) aims to overcome the limitations posed by noise. It is capable of providing more accurate geometric positions, which can be used for a wider range of downstream tasks [45, 46].

2.2 Open-vocabulary scene understanding

Open-vocabulary 3D scene understanding tasks have been increasingly explored by transferring 2D Vision Foundation Models (VFM) such as DINO [23], and SAM [22] into 3D RGB fields. These approaches [15, 47] learn 3D-consistent features by lifting 2D representations into 3D. The 2D Vision-Language Model (VLM) such as LSeg [20], CLIP [21] has exhibited impressive performance in zero-shot image understanding tasks. Several methods [16, 48] leverage models like SAM [22] or DINO [23] to extract clear boundaries, which are then fed into CLIP to obtain high-quality language features. These features are subsequently integrated into RGB and language fields to enable open-vocabulary 3D scene understanding.

Given the success of 3D Gaussian Splatting (3DGS) in common 3D scene reconstruction tasks, it has been extended to open-vocabulary 3D scene understanding. This advancement has improved the efficiency and controllability of integrating 3D scenes with language fields. We categorize the approaches for open-vocabulary 3D scene understanding into two types: two-stage and end-to-end methods. Two-stage methods, such as LangSplat [17], GOI [49], and OpenGaussian [50], achieve relatively satisfactory results by separately training RGB and language fields, but struggle to effectively balance the distinct optimization demands of each. Furthermore, due to the time-consuming nature of training in two-stage methods, end-to-end approaches have gained prominence. LEGaussians [18] is one such end-to-end method that simultaneously trains the RGB and language fields. By enabling joint optimization, it fosters a tighter integration between the geometric layout and semantic content.

3 Methodology

Leveraging the 2D Gaussian Splatting technique, we reconstruct a unified field integrating both RGB and language features from a given set of 2D images. This approach enables rendering RGB images from arbitrary viewpoints while supporting precise understanding through open-vocabulary language queries. An overview of our method is illustrated in Fig. 2.

3.1 Recap: 3D Gaussian splatting

3DGS [13] involves projecting and rendering of 3D gaussians into 2D image space. It initializes gaussians using Structure-from-Motion (SfM) [51] and constructs RGB fields from multi-view images to achieve high rendering quality. 3DGS explicitly parameterizes gaussian via 3D covariance matrix $\mathbf{\Sigma}$ and 3D position $p_i \in \mathbb{R}^3$:

$$\mathcal{G}^{3D}(x) = \exp(-\frac{1}{2}(x - p_i)^{\mathrm{T}}\mathbf{\Sigma}^{-1}(x - p_i)). \quad (1)$$

The 3D covariance matrix $\mathbf{\Sigma}$ can be decomposed into a rotation matrix $\mathbf{R}$ and a scaling matrix $\mathbf{S}$. When 3D Gaussians are projected into 2D image space, this matrix is transformed into $\mathbf{\Sigma}'$ as follows:

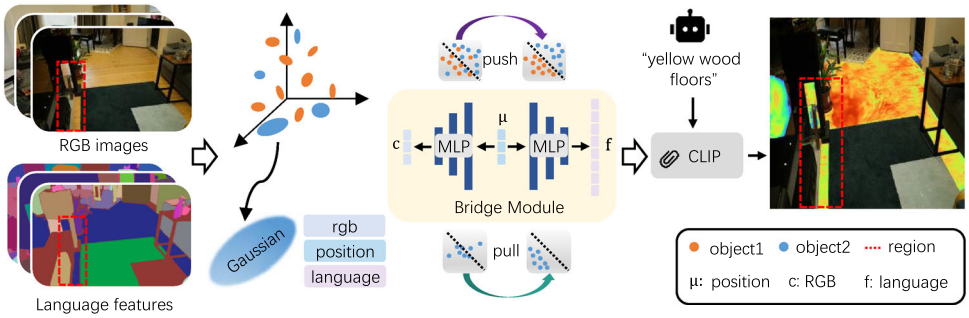

Fig. 2 Overview of PPGS framework. The core bridge module processes each gaussian based on its shared spatial position. Through a push–pull alignment mechanism, it actively corrects feature misalignment. The 'push' visualized by the purple arrow, enhances distinctiveness by pushing apart the features of visually similar but semantically different objects. Conversely, the 'pull' shown by the green arrow, encourages semantic consistency by pulling together the features of gaussians that belong to the same object (floor). The resulting aligned feature fields enable precise open-vocabulary querying. A text prompt ("yellow wood floors") is encoded via CLIP to generate a query vector, which is then used to produce a semantic heatmap on any novel rendered view

$$\mathbf{\Sigma} = \mathbf{R}\mathbf{S}\mathbf{S}^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}, \tag{2}$$

$$\mathbf{\Sigma}' = \mathbf{J}\mathbf{W}\mathbf{\Sigma}\mathbf{W}^{\mathrm{T}}\mathbf{J}^{\mathrm{T}}, \tag{3}$$

where $\mathbf{W}$ is the transformation matrix from world coordinates to camera coordinates, and $\mathbf{J}$ is the Jacobian matrix for the affine approximation of the projective transformation. Thus, the i-th ($i \in \mathcal{N}$) 3D gaussian is described by a group of learnable parameters $\{p_i, \mathbf{r}_i, \mathbf{s}_i, \alpha_i, sh_i\}$, which combines 3D position p, rotation matrix $\mathbf{r}$, scale matrix $\mathbf{s}$, opacity value α and spherical harmonics sh. To integrate the alpha-weighted appearance from front to back, volumetric alpha blending is applied with respect to index j:

$$C = \sum_{i=1}^{\mathcal{N}} c_i \alpha_i \mathcal{G}_i^{3\mathrm{D}} \prod_{j=1}^{i-1} (1 - \alpha_j \mathcal{G}_j^{3\mathrm{D}}), \tag{4}$$

3.2 2D Gaussian splatting for language fields

Different from 3D Gaussian Splatting, 2D Gaussian Splatting (2DGS) [14] represents scenes using flat 2D gaussians within the 3D field. This approach distributes densities of 2D gaussians along surfaces, improving positions with geometry and significantly reducing erroneous gaussians.

In 2DGS, each 2D gaussian is defined by a central point $p_i \in \mathbb{R}^3$ as in 3DGS, two principal tangential vectors $(\boldsymbol{t}_u, \boldsymbol{t}_v) \in \mathbb{R}^3$ and two scaling factors $(s_u, s_v) \in \mathbb{R}$. The 2D gaussian is then parameterized using a local tangent plane as follows, where $\mathbf{H}$ is a homogeneous transformation matrix representing the geometry of the 2D gaussian:

$$P(u, v) = p_i + s_u \boldsymbol{t}_u u + s_v \boldsymbol{t}_v v = \mathbf{H}(u, v, 1, 1)^{\mathrm{T}}, \tag{5}$$

$$\mathbf{H}(u, v, 1, 1)^{\mathrm{T}} = [\mathbf{R}_i \mathbf{S}_i, p_i], \tag{6}$$

where $\mathbf{R}_i = [\boldsymbol{t}_u, \boldsymbol{t}_v, \boldsymbol{t}_u \times \boldsymbol{t}_v] \in SO(3)$ represents rotation, and $\boldsymbol{t}_u \times \boldsymbol{t}_v$ denotes normal orientation. The scale is from diagonal matrix $\mathbf{S}_k = \mathrm{diag}(s_u, s_v, 0) \in \mathbb{R}^{3\times3}$. For each point $\mathbf{u} = (u, v)$ in uv space, the corresponding 2D gaussian value can be computed as:

$$\mathcal{G}^{2\mathrm{D}}(\mathbf{u}) = \exp(-\frac{u^2 + v^2}{2}). \tag{7}$$

Each 2D gaussian also carries an opacity value α and a view-dependent appearance c same as 3DGS. The color C at pixel x is obtained through volumetric rendering and is expressed as:

$$C(x) = \sum_{i=1}^{\mathcal{N}} c_i \alpha_i \mathcal{G}_i^{2\mathrm{D}} \prod_{j=1}^{i-1} (1 - \alpha_j \mathcal{G}_j^{2\mathrm{D}}). \tag{8}$$

In our work, we extend each 2D Gaussian with language embedding f. To generate language features, we follow the same initial process as LangSplat [17], leveraging the SAM [22] to produce features at three hierarchical levels: "Whole", "Part", and "Subpart". However, LangSplat is a two-stage method that trains the language features independently after an initial RGB field is pre-trained. Adopting such a multi-level, staged training scheme would disrupt the end-to-end nature of our joint optimization. Consequently, we employ a strategy that utilizes only the features from the "Whole" semantic level. This approach provides a consistent and stable semantic signal, which is beneficial for a joint optimization model. A similar strategy is also adopted by concurrent works like OpenGaussian [50]. The i-th 2D gaussian is represented by a set of learnable parameters $\{f_i, p_i, \mathbf{r}_i, \mathbf{s}_i, \alpha_i, sh_i, \epsilon_i\}$. Where $\mathcal{N}$ denotes the dynamically changing number of 2D gaussians during training. Directly embedding high-dimensional language features from models

like CLIP (512 dimensions) into a large number of gaussian primitives presents significant challenges in terms of memory consumption and computational efficiency. To address this, we adopt an effective strategy inspired by prior work such as LangSplat: a lightweight, scene-specific autoencoder [52] is pre-trained to compress the high-dimensional CLIP features into a low-dimensional latent space (8 dimensions). This dimensionality reduction is a critical step for achieving a memory-efficient representation, the autoencoder is trained using the following reconstruction loss L_{ae}:

$$L_{ae} = \sum_{x=1}^{y} d_{ae}(\Psi(E(\text{CLIP}(x)), \text{CLIP}(x))), \tag{9}$$

where Ψ and E denote the decoder and encoder, and $d_{ae}()$ is the distance function of the autoencoder using both $L1$ loss and cosine distance loss, y represents all pixels in a set of 2D images. Subsequently, the same tile-based rasterization pipeline used in 3DGS is applied to language field to preserve rendering efficiency:

$$F(x) = \sum_{i=1}^{\mathcal{N}} f_i \alpha_i \mathcal{G}_i^{\text{2D}} \prod_{j=1}^{i-1} (1 - \alpha_j \mathcal{G}_j^{\text{2D}}). \tag{10}$$

where $F(x)$ represents the language embedding rendered at pixel x. By embedding the language representation directly into 2D gaussians, the language field effectively responds to language-based queries.

3.3 Bridge module

The design of our bridge module is founded on a core principle in neural rendering. NeRF [25] establishes that volume density depends only on that position and is independent of the viewing direction. Similarly, the semantic identity of a spatial location, as represented by language embeddings, is also an intrinsic property of its position and remains independent of complex lighting conditions. This property makes the precise structural position p a stable and fundamental attribute to serve as a shared bridge between the RGB (visual appearance) and language (semantic meaning) fields. By treating the spatial position as a shared anchor, we enable supervision from both modalities to propagate back into 3D space, where alignment is ultimately expressed as changes in the distribution of gaussian centers.

Leveraging this principle, we design the bridge module as a dual-branch network structure (MLP_c for RGB and MLP_f for language features) that operates on a per-gaussian basis. The sole input to the module is the central position p_i of an individual 2D Gaussian. This design choice inherently accommodates the dynamic nature of Gaussian splatting, where primitives are changed during optimization. The dual-branch architecture enables a bidirectional information flow and facilitates a "push–pull" optimization dynamic: the total loss, with both visual and language components, is back propagated to the position p_i, since the language features are generated as a differentiable function of this position. This joint optimization corrects spatial misalignments, encouraging gaussians to settle in positions that are coherent across both modalities. Each input position p_i undergoes positional encoding (PE) as in NeRF, before being processed by the two parallel branches:

1. RGB Branch (MLP_c)**.** The encoded gaussian position p_i is passed through four fully connected ReLU layers, each with 96 channels. A skip connection concatenates the position encoding with the activation of the fifth layer, yielding a 159-dimensional feature vector. A final fully connected layer then outputs a three-dimensional RGB color representation:

$$c_i = \text{MLP}_c(\text{PE}(p_i)), \tag{11}$$

2. Language Branch (MLP_f)**.** The same position encoding is processed through an identical architecture but optimized for language features. A final fully connected layer outputs an d-dimensional latent language embedding:

$$f_i = \text{MLP}_f(\text{PE}(p_i)). \tag{12}$$

A key design choice is that the bridge module operates on view-independent features. This is a deliberate strategy to focus on achieving robust cross-modal alignment directly in the 3D domain. View-dependent effects, such as specularities, are handled at the rendering stage. Higher-order spherical harmonic (SH) coefficients, supervised by the final L_1 rendering loss, remain responsible for capturing view-dependent effects at the rendering stage. MLP_c and MLP_f share parameters in the first five layers to reduce memory consumption while maintaining symmetry. The final layers independently produce RGB color $c \in \mathbb{R}^3$ and the latent language embedding $f \in \mathbb{R}^d$. This symmetric dual-branch structure effectively balances computational efficiency and feature expressiveness. Additionally, we introduce an optimizable uncertainty ϵ for each 2D Gaussian following the approach in LEGaussians:

$$\beta = \beta_{uncert}(1 - \epsilon) + \epsilon, \tag{13}$$

where $\epsilon \in [0, 1]$ represents the confidence of 2D Gaussians. A higher ϵ indicates instability, leading to frequent updates during optimization.3 We also set $\beta_{uncert} = 0.1$, which acts as a threshold to control the effective weights of 2D Gaussians input to the MLP. To train the bridge module, two loss functions are designed:

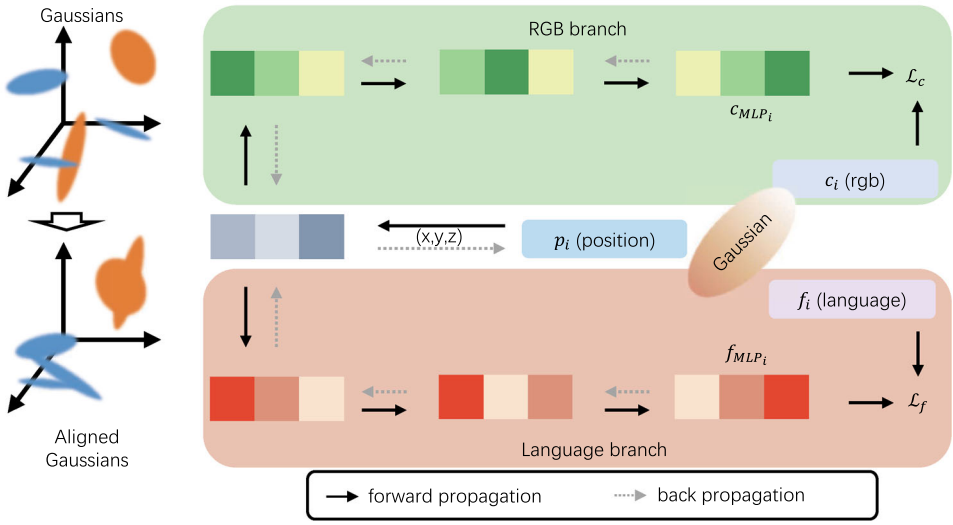

Fig. 3 The bridge module's push–pull alignment mechanism. A Gaussian's position (xyz) is input to parallel branches to predict RGB and language features (solid arrows). Gradients from losses L_c and L_f backpropagate (dotted arrows) to update both the branches and the position itself, correcting misalignment to produce the "Aligned Gaussians"

$$\mathcal{L}_c = \sum_{i=0}^{\mathcal{N}} (\| c_{\text{MLP}_i} - c_i^* \|_2 + \beta \| c_{\text{MLP}_i}^* - c_i \|_2), \tag{14}$$

$$\mathcal{L}_f = \sum_{i=0}^{\mathcal{N}} (\| f_{\text{MLP}_i} - f_i^* \|_2 + \beta \| f_{\text{MLP}_i}^* - f_i \|_2), \tag{15}$$

where c_i and f_i represent the RGB and language parameters of each 2D Gaussian, respectively. We adapt the bidirectional loss structure from spatial smoothness regularizer for our task of multi-modal fusion. To ensure stable cross-modal learning, we employ an iterative optimization schedule where one branch is updated while the other is frozen (*). For instance, the RGB branch is trained for several iterations using L_c while the language branch's parameters are frozen, after which the roles are reversed. This alternating process prevents destructive interference between the two tasks. This creates a push–pull dynamic where the bridge module and the gaussian parameters are co-trained, effectively aligning and fusing the two modalities. We introduce an adaptive weighting mechanism to balance RGB and language reconstruction tasks:

$$\mathcal{L}_{rec} = \mathcal{L}_1 + \mathcal{L}_c, \tag{16}$$

$$\mathcal{L}_{rf} = \mathcal{L}_r + \mathcal{L}_f, \tag{17}$$

where $\mathcal{L}_1$ represents the difference between the rendered RGB images and the original images, and $\mathcal{L}_r$ denotes the difference between the rendered language features and the latent language features. With this loss function, both tasks benefit from improvements, further enhancing the performance of PPGS.

Following [53], this approach formulates homoscedastic uncertainty, which can be learned using probabilistic deep learning to estimate task-specific confidence levels. It measures the inherent uncertainty of the task itself, independent of the input data:

$$\mathcal{L} = \mathcal{L}_{rec} \hat{\sigma}_{rec}^{-2} + \log \hat{\sigma}_{rec}^2 + \mathcal{L}_{rf} \hat{\sigma}_{rf}^{-2} + \log \hat{\sigma}_{rf}^2, \tag{18}$$

where $\hat{\sigma}_{rec}^2$ and $\hat{\sigma}_{rf}^2$ dynamically adjust during backpropagation. These uncertainties are free scalar values, not model outputs, and represent the task noise. This loss $\mathcal{L}$ consists of two components: the residual regression term $\log \hat{\sigma}^2$ and the uncertainty regularization term $\hat{\sigma}^2$. Once the variance of $\hat{\sigma}^2$ becomes larger, it exerts a tempering effect on the residual regression term. Larger variances will result in a smaller residual loss. The regularization term prevents the network from predicting infinite uncertainty, which would lead to zero loss. This mechanism allows PPGS to adaptively balance RGB and language tasks.

4 Experiments

4.1 Setting

Dataset. To assess the effectiveness of our approach, we conduct experiments on the Mip-NeRF360 dataset [55], adopting the same settings as those used in LEGaussians [18]. The Mip-NeRF360 dataset is a high-quality, real-world collection for 3D reconstruction, featuring a diverse array of everyday objects and intricate background textures. It comprises six distinct scenes (bicycle, bonsai, counter, garden, kitchen, and room) encompassing both indoor and outdoor settings. Each scene comprises approximately 200 images captured from diverse viewpoints and distances, thereby enhancing data richness and increasing the complexity of the reconstruction task. The dataset includes manually annotated segmentation masks, using descriptive labels (e.g., "purple tablecloth") to

Table 1 Quantitative comparison of our method with DFF [15], LERF [16], 3DOVS [54], Feature-3dgs [24], LEGaussians [18] and LangSplat [17] on the Mip-NeRF360 dataset [55]. To ensure a fair comparison, results for Feature-3dgs, LEGaussians, and LangSplat are re-trained. The results for LEGaussians* are the original scores reported in their paper. For broader reference, the results for DFF, LERF, and 3DOVS are cited from the LEGaussians paper. The first , second , and third performance are highlighted. Our method shows overall superior performance over baseline methods

Method	PSNR↑	SSIM↑	LPIPS↓	mIoU↑	mAP↑	FPS↑	Training Time↓
DFF [15]	25.378	0.712	0.312	0.091	0.199	0.20	184min
LERF [16]	25.749	0.811	0.317	0.403	0.688	0.04	54min
3DOVS [54]	25.782	0.733	0.295	0.458	0.550	0.17	158min
Feature-3dgs [24]	29.794	0.900	0.103	0.486	0.817	13.00	340min
LEGaussians [18]	29.495	0.860	0.146	0.565	0.822	89.00	68min
LEGaussians* [18]	29.826	0.901	0.112	0.578	0.815	89.00	68min
LangSplat [17]	29.759	0.897	0.103	0.619	0.830	92.00	145min
Ours	29.798	0.903	0.101	0.623	0.832	90.00	72min

identify the primary object or background in each scene. The test set comprises novel view images, randomly selected to constitute approximately 10% of each corresponding training set.

Baseline Methods and Metrics. We compare our method through a comparative analysis against several mainstream approaches: DFF [15], LERF [25], 3DOVS [54], Feature-3dgs [24], LangSplat [17] and LEGaussians [18]. To establish a fair and rigorous comparison against key contemporary methods, we retrained Feature-3dgs, LangSplat, and our core baseline, LEGaussian, within a unified experimental setup. This ensures all comparisons are based on a consistent benchmark. For a broader context, results for earlier methods (DFF, LERF, 3DOVS) are cited from the LEGaussians paper [18]. The results for LEGaussians* are the original scores reported in their paper. Among these, LangSplat exemplifies the leading two-stage approach, whereas LEGaussians represents the leading end-to-end method. We evaluate these methods using the following criteria. Visual quality is quantified by PSNR, SSIM, and LPIPS [56], measuring the fidelity of novel view renderings. Language query accuracy is measured by mean Intersection over Union (mIoU) and mean Average Precision (mAP), based on annotated labels in the test set. Rendering speed is reported in frames per second (FPS), evaluated for images incorporating language features at a fixed resolution.

Implementation Details. We implement our method with PyTorch [57] and incorporate the CUDA kernel from 2D Gaussian Splatting [14] to accelerate the rasterization rendering process. Our approach employs joint learning to optimize both the RGB and language fields. For extracting latent language features, we follow the same process as LangSplat [17], but we utilize only the whole level embedding, and omit the "tire" label in the scene "bicycle" as it represents a hierarchical structure. The same operation is also performed in the method proposed by [50]. The model is trained on a single RTX3090 GPU for approximately 1 h, with 30,000 iterations. We use the Adam optimizer with a learning rate of 0.001 and betas of (0.9, 0.999). The bridge module and joint learning were optimized with a weight of 0.001. The reported results are the averages of five independent trials, the standard deviations were consistently low across all metrics (typically within ± 0.05 for PSNR), confirming the stability of our findings and are thus omitted for brevity.

4.2 Comparison

4.2.1 Quantitative results

Our method surpasses other methods in both rendering quality and language query accuracy as shown in Table 1. It achieves a notable mIoU of "0.632", although this performance is inherently constrained by the dataset's limitations. Although query objects may be accurately recognized, they still exhibit lower mIoU scores, reflecting an upper bound on accuracy due to ambiguities in human labeling. 3DOVS and DFF require significant memory and storage due to the use of raw language features during training. LERF exhibits the slowest rendering speed, attributable to its MLP architecture, and shows poor language query accuracy. Feature-3dgs [24] employs LSeg [20] as a 2D semantic segmentation model for feature extraction. However, it struggles with open-vocabulary queries, often retrieving all objects related to the prompt and failing to distinguish complex cases. Feature-3dgs incurs significant resource costs due to the construction of a speed-module network and the dimensionality compression of LSeg. This results in a fivefold increase in training time and a reduction to one-seventh of the original FPS, rendering it impractical. Furthermore, end-to-end frameworks like LEGaussians introduce an imbalance issue between the RGB and language modalities. The "garden" and "kitchen" scenes suffer the most from imbalance. LangSplat halts RGB reconstruction during language embedding, resulting in a PSNR equivalent to the original 3DGS reconstruction. In contrast, our method enhances precision through two key

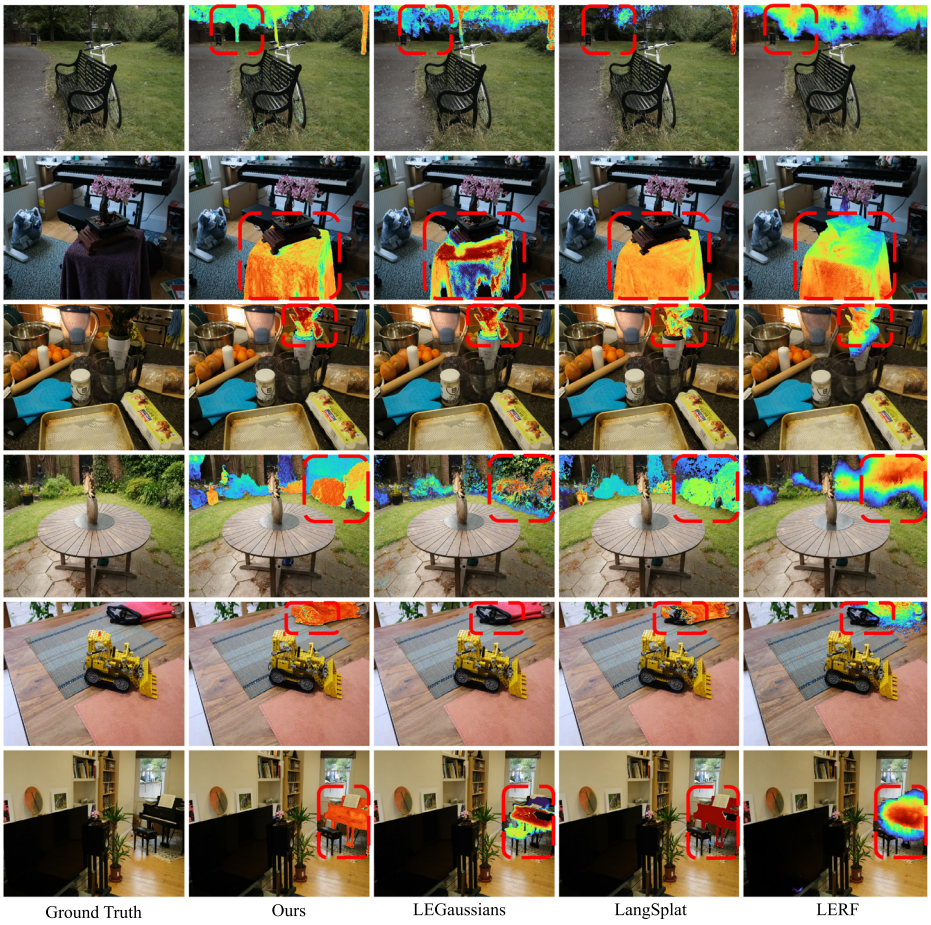

Fig. 4 Comparison of novel view synthesis and query relevance visualization. From left to right: Ground truth novel view synthesis, novel view images with relevance visualization from our method, LEGaussians [18], LangSplat [17], and LERF [16]. From top to bottom: Query words “silver oak tree" in the “bicycle" scene, “purple table cloth" in “bonsai" scene, “plants" in the “counter" scene, “green plant" in the “garden" scene, “red oven gloves" in the “kitchen" scene, and “piano keyboard" in the “room" scene

components. First, 2DGS provides a robust geometric foundation. Second, the bridge module facilitates field integration and joint learning, which in turn mitigates the imbalance issue. Experiments clearly demonstrate the effectiveness of our approach, with an increase in PSNR and improvements in language query accuracy, with detailed results for each individual scene provided in Table 5.

4.2.2 Qualitative results

To illustrate the effectiveness of our method in language-based scene understanding, we present a qualitative comparison in Fig. 4, where our approach successfully identifies complex objects in diverse environments. LERF successfully locates queried objects due to its grid-based representation but struggles with defining precise boundaries, as seen with the “green plants" in the “garden" scene. LEGaussians fails to identify the “red oven gloves" in the “kitchen" scene due to its reliance on DINO [23], which struggles with detecting small objects in sparse views and complex backgrounds. LangSplat suffers from noise in the reconstruction due to the lack of geometric constraints, as demonstrated by the hollow section of the “piano keyboard" in the “room" scene. In contrast, our approach benefits from 2D gaussians, which provide geometrically constrained positions that minimize noise. The bridge module further enhances both RGB and language by combining information from both fields, thereby suppressing noise generation. For example, the “silver oak tree" in the “bicycle" scene shows almost no noise at the edges, as the RGB and language constraints work together effectively. Similarly, a more compact language distribution for “plants" is obtained in the “counter" scene. Further qualitative results are shown, including the mask visualization of open-vocabulary queries (Fig. 6) and a visual quality comparison with other methods (Fig. 5). These results demonstrate that our method reconstructs higher-quality RGB and significantly enhances the performance of open-vocabulary scene understanding.

4.2.3 Ablation study

We conducted ablation experiments to analyze variability across different scenarios. Similar to its performance in the “room" scenario, our method demonstrates excellent results in the “kitchen" scenario. The results of the ablation studies for the language query are presented in Tab .2. 2DGS with its inherent multi-view geometric consistency reduces

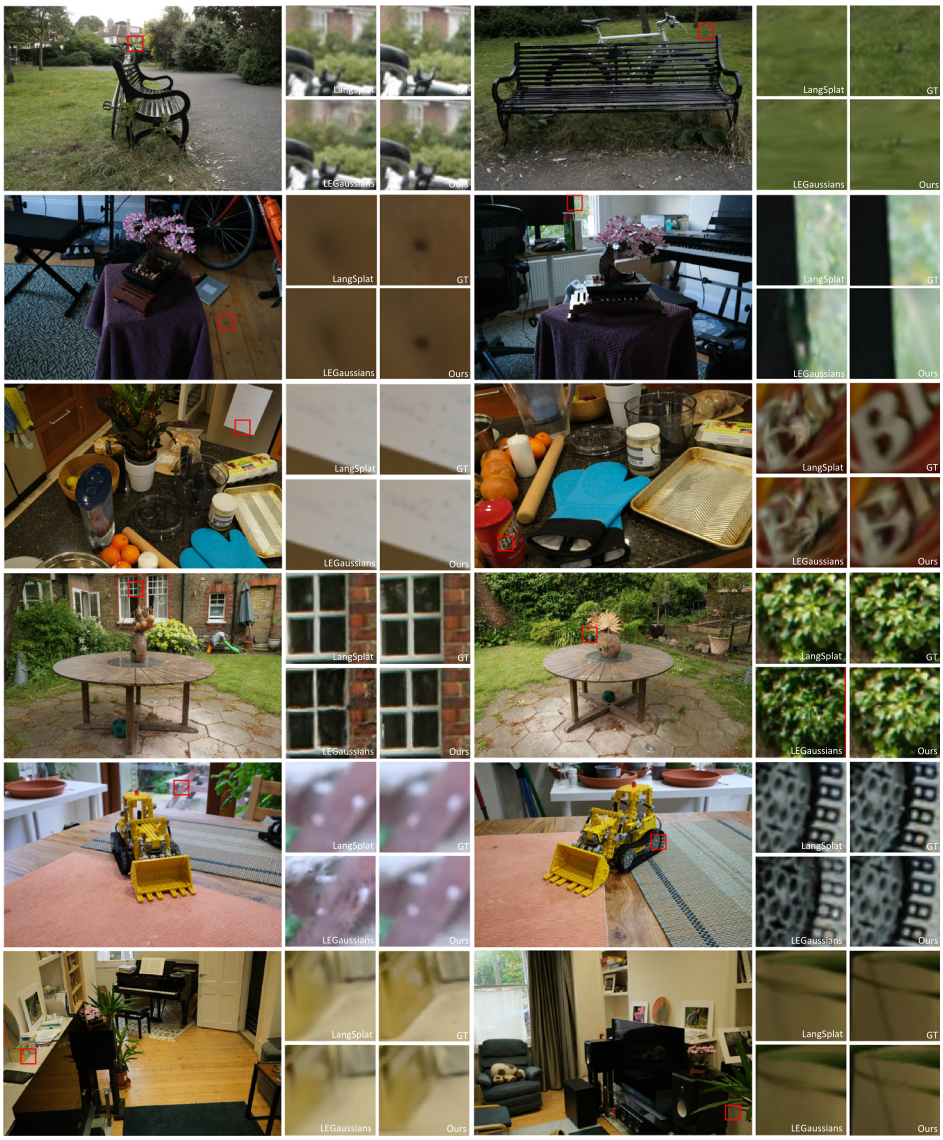

Fig. 5 Visual quality comparison of novel view synthesis results. Our method is able to recover more detailed texture and appearance compared to LEGaussians [18] and LangSplat [17]. From top to bottom: “bicycle" scene, “bonsai" scene, “counter" scene, “garden" scene, “kitchen" scene, and “room" scene

the noise in the field and mitigates the impact of erroneous language embeddings. 2DGS experiences a slight drop in mIoU with “kitchen" for complex scenes, while maintaining high mAP performance. This disparity arises because the “kitchen" scene contains more occlusions and complex objects, which can degrade performance when overly rigid structural constraints are applied during reconstruction. By leveraging the bridge module’s dual-branch architecture, the system adapts to the complexities of the scene, ensuring that occlusions and intricate object structures are better handled. The bridge module facilitates mutual learning between the RGB field and gaussian positions, as well as between the language field and gaussian positions. This mutual interaction allows the gaussian representations to positively influence the reconstruction of both fields. Our results demonstrate that the bridge module is crucial for language query performance, as it improves the mIoU and mAP. We further validate the effectiveness of the bridge module, experimental results show that in the “room" scenario, introducing the bridge module improves the mIoU from “0.501" to “0.563", demonstrating a significant performance gain. Similarly, a comparable trend is observed in the “kitchen" scenario, further confirming that the bridge module effectively facilitates mutual learning between different fields and enhances feature fusion. The PSNR metrics in Table 1 clearly demonstrate the imbalance problem. The introduction of joint learning in our framework effectively mitigates this imbalance, resulting in better performance .

4.2.4 Discussions

Bridge Module. We delve deeper into the functional impact of the bridge module. As shown in Table 3, the model without

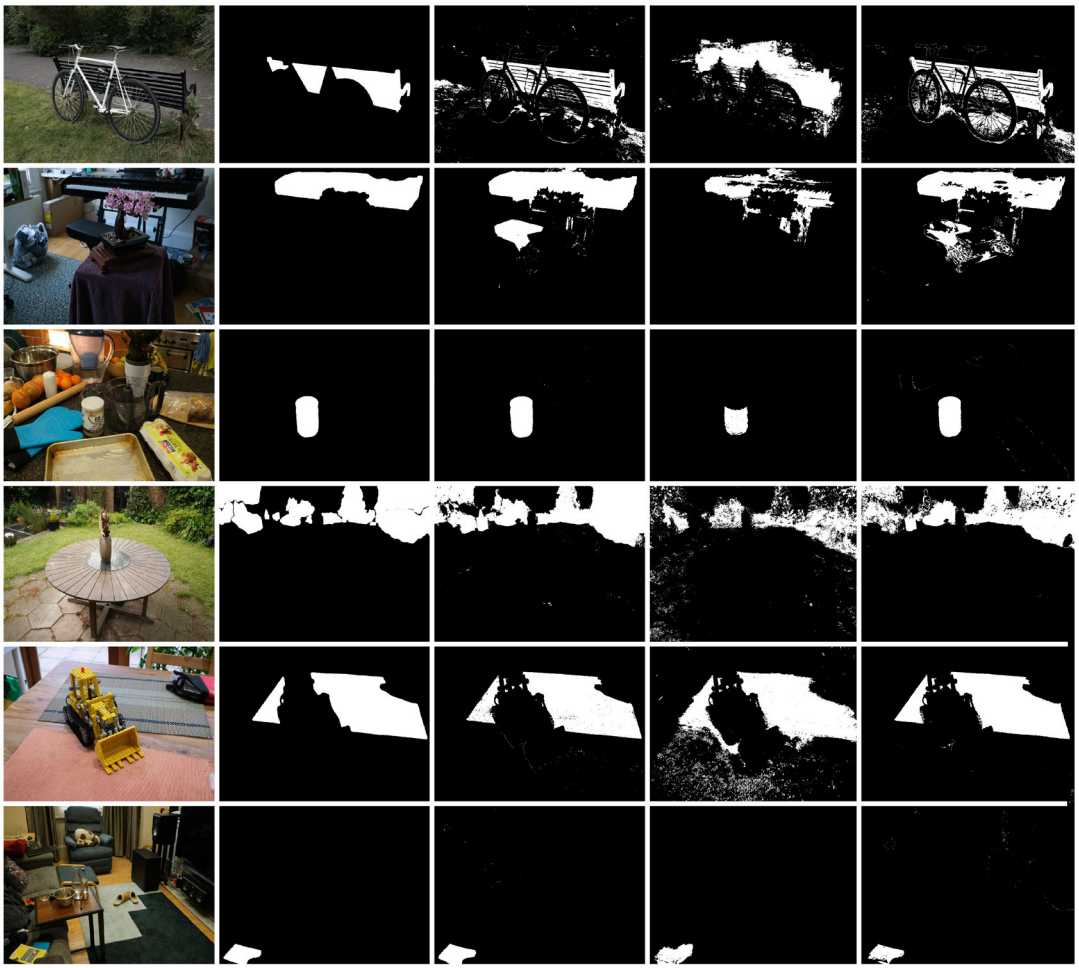

Fig. 6 A comparison of novel view synthesis and the mask visualization, which determines the value of mIoU. Left to right: Ground truth, GT mask, mask from our method, LEGaussians [18], LangSplat [17]. Our method produces clearer and more complete boundaries, improving the overall quality of open-vocabulary queries. From top to bottom: Query words “bench" in scene “bicycle", “piano keyboard" in scene “bonsai", “jar of coconut oil" in scene “counter", “green plant" in scene “garden", “basket weave cloth" in scene “kitchen", and “yellow books" in scene “room"

Table 2 Quantitative results of ablation experiments (“room" and “kitchen" scenes). The first, second, and third performance are highlighted

scene	3DGS	joint learning	Bridge Module	2DGS	mIoU↑	mAP↑
room	✓				0.461	0.747
	✓		✓		0.501	0.757
				✓	0.532	0.734
			✓	✓	0.563	0.768
		✓	✓	✓	0.572	0.770
kitchen	✓				0.716	0.886
	✓		✓		0.721	0.916
				✓	0.673	0.910
			✓	✓	0.723	0.938
		✓	✓	✓	0.744	0.946

the bridge module results in the lowest performance for both RGB reconstruction and language query tasks.

Retaining only the language branch in bridge significantly improves language query performance. Similarly, keeping only the RGB branch in bridge enhances RGB reconstruction performance. However, this single-branch configuration shifts the loss function, adversely affecting the other task and reducing overall performance. The dual-branch architecture is able to merge the information from two different fields in a way that improves the performance of each task at the same time and outperforms the single-branch architecture. The dual-branch structure effectively reduces ambiguity and significantly enhances open-vocabulary scene understanding mIoU from “0.532" to “0.572".

We continue our analysis of the network structure of the bridge module. The position encoding layer serves a similar purpose to that in NeRF [25], ensuring the preservation of high-frequency details to prevent information loss. As shown in Table 4, the absence of the position encoding layer results in a noticeable degradation in performance. Similarly, removing the skip connection negatively impacts performance, as it weakens the transmission of critical information, requiring additional reinforcement. Performance exhibits consistent improvement with an increase in the number of layers, as deeper architectures enhance the network’s capacity to learn complex patterns and capture finer details. Additionally, increasing the dimensionality of the MLP from “96" to “128" enhances performance but comes with a higher parameter count. To maintain a balance between performance and efficiency, we adopt a dimensionality of “96" in this study.

Visualization Results. As shown in Fig. 5, language significantly boosts RGB visual quality. Objects with similar language features pull cluster gaussians closer, improving RGB reconstruction, while dissimilar features push them apart. Our method performs best in the “bonsai" scene, where the floor is clearly reconstructed. LEGaussians suffers the most from the imbalance between language and RGB. In contrast, the LangSplat method experiences no imbalance interference, but its performance still lags behind ours in terms of visual quality due to the lack of language assistance. The mask visualization of open-vocabulary queries (Fig. 6) showcases our model’s ability to generate precise and semantically meaningful segmentations, highlighting its robustness in handling diverse and unseen categories. Furthermore, a visual quality comparison with other methods underscores our method’s superior RGB reconstruction, which significantly enhances the performance of open-vocabulary scene understanding (Table 5).

4.2.5 Downstream tasks

Our PPGS framework supports a range of powerful downstream tasks for interactive 3D scene editing and understanding. These applications all stem from the model’s ability to accurately ground language and prompts within its learned feature fields. Fig. 1 provides a visual summary of these key applications, demonstrating object deletion, extraction, and segmentation via both click and language prompts. The implementation for each is as follows:

Delete Anything: The ‘Delete Anything’ panel shows object removal via language. For the command “Please delete the brown shoes”, the system generates a 3D selection mask for all gaussians whose language features exceed a similarity threshold with the text query. During rendering, gaussians within this mask are skipped, resulting in their seamless removal as shown.

Extract Anything: The ‘Extract Anything’ panel visualizes object isolation. Using the prompt “the brown shoes”, a similar 3D mask is generated based on a feature simi-

Table 3 Discuss the impact of each branch of bridge in the “room" scene. The first, second, and third performance are highlighted

Methods	PSNR↑	SSIM↑	mIoU↑	mAP↑
w/o Bridge Module	32.493	0.954	0.532	0.734
w/o RGB branch	32.405	0.954	0.565	0.768
w/o language branch	32.624	0.955	0.535	0.739
ours full	32.695	0.956	0.572	0.770

Table 4 A comprehensive analysis of the network structure of the bridge module in the “room" scene. The first, second, and third performance are highlighted

	Layer	PSNR↑	SSIM↑	mIoU↑	mAP↑
Bridge module	w/o pe	32.660	0.954	0.570	0.766
	w/o skip	32.704	0.955	0.564	0.765
	3 layer, 96 dims	32.703	0.955	0.564	0.764
	2 layer, 96 dims	32.518	0.954	0.569	0.767
	1 layer, 96 dims	32.661	0.955	0.568	0.766
	1 layer, 128 dims	32.660	0.955	0.571	0.761
Ours	4 layer, 96 dims	32.695	0.956	0.572	0.770

Table 5 The specificity values of our approach in each scene are provided

Scene	PSNR↑	SSIM↑	LPIPS↓	mIoU↑	mAP↑
bicycle	24.186	0.732	0.245	0.535	0.848
bonsai	31.981	0.957	0.077	0.715	0.921
counter	29.776	0.936	0.074	0.651	0.808
garden	27.792	0.867	0.104	0.486	0.685
kitchen	31.768	0.961	0.043	0.761	0.953
room	33.285	0.964	0.062	0.589	0.776
Average	29.798	0.903	0.101	0.623	0.832

Specifically, PSNR, SSIM, and LPIPS are related to RGB reconstruction, while mIoU and mAP are associated with open-vocabulary query performance

larity threshold. The rendering logic is then inverted: only gaussians included in the mask are processed, presenting the object against a black background.

Click Segment Anything: The 'Click Segment Anything' panel showcases interactive segmentation from a spatial prompt. A user's click, for instance on the chair, is used to predict segmentation scores across the chair scene. A final selection mask is then generated by applying a threshold to these scores, which instructs the renderer to highlight the entire object instance with the overlay shown.

Language Segment Anything: The 'Language Segment Anything' panel shows direct segmentation from a text command. For the prompt "Piano", the system generates a selection mask based on a feature similarity threshold. This mask then instructs the renderer to visually distinguish the selected Gaussians, resulting in the highlighted piano in the image.

5 Conclusion

In this paper, we propose a novel language-embedded framework named PPGS, which enhances open-vocabulary scene understanding by simultaneously reconstructing RGB and language fields. Our method addresses the critical issue of misalignment between RGB and language representations through a push–pull mechanism within the bridge module. This mechanism improves cross-modal alignment and promotes effective fusion between RGB and language fields. Furthermore, to ensure stable and efficient convergence of these distinct fields during joint learning, we optimize joint learning by modulating gradients to accommodate the distinct optimization requirements of RGB and language fields, thereby ensuring efficient and stable convergence. These results demonstrate the effectiveness of our approach in achieving high-quality RGB reconstruction and improved language query accuracy.

Acknowledgements This paper is supported by the Science and Technology Project of Hainan Provincial Department of Transportation (Grant No. HNJTTKXC- 2024-3-22-02), the National Natural Science Foundation of China (Grant No. 62272018, 62206184).

Author Contributions Author 1 (First Author): Conceptualization, Methodology, Software, Investigation,Data Curation, Formal Analysis, Formal Analysis, Writing - Original Draft; Author 2 : Software, Visualization, Validation; Author 3 : Resources, Validation; Author 4 : Resources, Project Administration; Author 5 : Resources, Supervision; Author 6(Corresponding Author) : Conceptualization, Funding Acquisition, Resources, Supervision, Writing - Review & Editing;.

Data Availability Code repository: https://gitee.com/VR_NAVE/ppgs.git

Declarations

Conflict of interest We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work; there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled.

References

1. Jiang, L., Cai, L., Wu, W., Zhou, Z.: Mirror world: creating digital twins of the space and persons from video streamings. Vis. Comput. **40**(9), 6689–6704 (2024)
2. Zhou, Y., Yan, F., Zhou, Z.: Handling pure camera rotation in semi-dense monocular slam. Vis. Comput. **35**, 123–132 (2019)
3. Wu, Z., Zhou, Z., Tian, D., Wu, W.: Reconstruction of three-dimensional flame with color temperature. Vis. Comput. **31**, 613–625 (2015)
4. Yao, J., Chen, J., Niu, L., Sheng, B.: Scene-aware human pose generation using transformer. In: Proceedings of the 31st ACM International Conference on Multimedia, pp. 2847–2855 (2023)
5. Zhu, X., Yao, X., Zhang, J., Zhu, M., You, L., Yang, X., Zhang, J., Zhao, H., Zeng, D.: Tmsdnet: Transformer with multi-scale dense network for single and multi-view 3d reconstruction. Comput. Anim. Virt. Worlds **35**(1), 2201 (2024)
6. Liu, Y., Huang, E., Zhou, Z., Wang, K., Liu, S.: 3d facial attractiveness prediction based on deep feature fusion. Comput. Anim. Virt. Worlds **35**(1), 2203 (2024)
7. Gordon, D., Kembhavi, A., Rastegari, M., Redmon, J., Fox, D., Farhadi, A.: Iqa: Visual question answering in interactive environments. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4089–4098 (2018)
8. Azuma, D., Miyanishi, T., Kurita, S., Kawanabe, M.: Scanqa: 3d question answering for spatial scene understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19129–19139 (2022)
9. Shi, W., Gao, D., Xiong, Y., Zhou, Z.: Qr-clip: Introducing explicit knowledge for location and time reasoning. ACM Trans. Multimed. Comput. Commun. Appl. **20**(11), 1–22 (2024)
10. Liu, K., Zhan, F., Zhang, J., Xu, M., Yu, Y., El Saddik, A., Theobalt, C., Xing, E., Lu, S.: Weakly supervised 3d open-vocabulary segmentation. Adv. Neural. Inf. Process. Syst. **36**, 53433–53456 (2023)
11. Xiong, Y., Wang, J., Zhou, Z.: Virtualloc: large-scale visual localization using virtual images. ACM Trans. Multimed. Comput. Commun. Appl. **20**(3), 1–19 (2023)
12. Qin, Y., Zhao, N., Sheng, B., Lau, R.W.: Text2city: one-stage text-driven urban layout regeneration. Proc. AAAI Conf. Artif. Intell. **38**, 4578–4586 (2024)
13. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139 (2023)
14. Huang, B., Yu, Z., Chen, A., Geiger, A., Gao, S.: 2d gaussian splatting for geometrically accurate radiance fields. In: ACM SIGGRAPH 2024 Conference Papers, pp. 1–11 (2024)
15. Kobayashi, S., Matsumoto, E., Sitzmann, V.: Decomposing nerf for editing via feature field distillation. Adv. Neural. Inf. Process. Syst. **35**, 23311–23330 (2022)

16. Kerr, J., Kim, C.M., Goldberg, K., Kanazawa, A., Tancik, M.: Lerf: Language embedded radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 19729–19739 (2023)
17. Qin, M., Li, W., Zhou, J., Wang, H., Pfister, H.: Langsplat: 3d language gaussian splatting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 20051–20060 (2024)
18. Shi, J.-C., Wang, M., Duan, H.-B., Guan, S.-H.: Language embedded 3d gaussians for open-vocabulary scene understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5333–5343 (2024)
19. Zhang, H., Wei, Z., Liu, G., Wang, R., Mu, R., Liu, C., Yuan, A., Cao, G., Hu, N.: Mkeah: Multimodal knowledge extraction and accumulation based on hyperplane embedding for knowledge-based visual question answering. Virtual Real. Intell. Hardw. **6**(4), 280–291 (2024)
20. Li, B., Weinberger, K.Q., Belongie, S., Koltun, V., Ranftl, R.: Language-driven semantic segmentation. arXiv preprint arXiv:2201.03546 (2022)
21. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763 (2021). PMLR
22. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.-Y., Dollár, P., Girshick, R.: Segment anything. arXiv:2304.02643 (2023)
23. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9650–9660 (2021)
24. Zhou, S., Chang, H., Jiang, S., Fan, Z., Zhu, Z., Xu, D., Chari, P., You, S., Wang, Z., Kadambi, A.: Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 21676–21685 (2024)
25. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Commun. ACM **65**(1), 99–106 (2021)
26. Chaurasia, G., Duchene, S., Sorkine-Hornung, O., Drettakis, G.: Depth synthesis and local warps for plausible image-based navigation. ACM Trans. Graph. (TOG) **32**(3), 1–12 (2013)
27. Eisemann, M., De Decker, B., Magnor, M., Bekaert, P., De Aguiar, E., Ahmed, N., Theobalt, C., Sellent, A.: Floating textures. In: Computer Graphics Forum, vol. 27, pp. 409–418 (2008). Wiley Online Library
28. Goesele, M., Snavely, N., Curless, B., Hoppe, H., Seitz, S.M.: Multi-view stereo for community photo collections. In: 2007 IEEE 11th International Conference on Computer Vision, pp. 1–8 (2007). IEEE
29. Hedman, P., Philip, J., Price, T., Frahm, J.-M., Drettakis, G., Brostow, G.: Deep blending for free-viewpoint image-based rendering. ACM Trans. Graph. (ToG) **37**(6), 1–15 (2018)
30. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3d. In: ACM Siggraph 2006 Papers, pp. 835–846 (2006)
31. Wang, Y., Wang, J., Qu, Y., Qi, Y.: Rip-nerf: learning rotation-invariant point-based neural radiance field for fine-grained editing and compositing. In: Proceedings of the 2023 ACM International Conference on Multimedia Retrieval, pp. 125–134 (2023)
32. Wang, Y., Wang, J., Qi, Y.: We-gs: An in-the-wild efficient 3d gaussian representation for unconstrained photo collections. arXiv preprint arXiv:2406.02407 (2024)
33. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: European Conference on Computer Vision, pp. 333–350 (2022). Springer
34. Garbin, S.J., Kowalski, M., Johnson, M., Shotton, J., Valentin, J.: Fastnerf: High-fidelity neural rendering at 200fps. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 14346–14355 (2021)
35. Huang, C., Li, X., Zhang, S., Cao, L., Ji, R.: Nerf-dets: Enhancing multi-view 3d object detection with sampling-adaptive network of continuous nerf-based representation. arXiv preprint arXiv:2404.13921 (2024)
36. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Trans. Graph. (TOG) **41**(4), 1–15 (2022)
37. Wang, Y., Wang, J., Wang, C., Duan, W., Bao, Y., Qi, Y.: Scarf: Scalable continual learning framework for memory-efficient multiple neural radiance fields. In: Computer Graphics Forum, p. 15255 (2024). Wiley Online Library
38. Yu, Z., Chen, A., Huang, B., Sattler, T., Geiger, A.: Mip-splatting: Alias-free 3d gaussian splatting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19447–19456 (2024)
39. Lu, T., Yu, M., Xu, L., Xiangli, Y., Wang, L., Lin, D., Dai, B.: Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 20654–20664 (2024)
40. Chen, D., Li, H., Ye, W., Wang, Y., Xie, W., Zhai, S., Wang, N., Liu, H., Bao, H., Zhang, G.: Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. arXiv preprint arXiv:2406.06521 (2024)
41. Qu, Z., Vengurlekar, O., Qadri, M., Zhang, K., Kaess, M., Metzler, C., Jayasuriya, S., Pediredla, A.: Z-splat: Z-axis gaussian splatting for camera-sonar fusion. arXiv preprint arXiv:2404.04687 (2024)
42. Yu, Z., Sattler, T., Geiger, A.: Gaussian opacity fields: Efficient high-quality compact surface reconstruction in unbounded scenes. arXiv:2404.10772 (2024)
43. Fan, Z., Wang, K., Wen, K., Zhu, Z., Xu, D., Wang, Z.: Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. arXiv preprint arXiv:2311.17245 (2023)
44. Guédon, A., Lepetit, V.: Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5354–5363 (2024)
45. Zhao, Y., Wu, C., Huang, B., Zhi, Y., Zhao, C., Wang, J., Gao, S.: Surfel-based gaussian inverse rendering for fast and relightable dynamic human reconstruction from monocular video. arXiv preprint arXiv:2407.15212 (2024)
46. Xu, J., Wang, Y., Zhao, Y., Fu, Y., Gao, S.: 3d streetunveiler with semantic-aware 2dgs. arXiv preprint arXiv:2405.18416 (2024)
47. Tschernezki, V., Laina, I., Larlus, D., Vedaldi, A.: Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. In: 2022 International Conference on 3D Vision (3DV), pp. 443–453 (2022). IEEE
48. Ye, J., Wang, N., Wang, X.: Featurenerf: Learning generalizable nerfs by distilling foundation models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8962–8973 (2023)
49. Qu, Y., Dai, S., Li, X., Lin, J., Cao, L., Zhang, S., Ji, R.: Goi: Find 3d gaussians of interest with an optimizable open-vocabulary semantic-space hyperplane. arXiv preprint arXiv:2405.17596 (2024)
50. Wu, Y., Meng, J., Li, H., Wu, C., Shi, Y., Cheng, X., Zhao, C., Feng, H., Ding, E., Wang, J., et al.: Opengaussian: Towards point-level 3d gaussian-based open vocabulary understanding. arXiv preprint arXiv:2406.02058 (2024)

51. Schonberger, J.L., Frahm, J.-M.: Structure-from-motion revisited. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4104–4113 (2016)
52. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
53. Kendall, A., Gal, Y., Cipolla, R.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7482–7491 (2018)
54. Liu, K., Zhan, F., Zhang, J., Xu, M., Yu, Y., El Saddik, A., Theobalt, C., Xing, E., Lu, S.: 3d open-vocabulary segmentation with foundation models. arXiv preprint arXiv:2305.14093 2(3), 6 (2023)
55. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5470–5479 (2022)
56. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–595 (2018)
57. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. In: NIPS-W (2017)

Tong Chen received the M.S. degree from North China Electric Power University, Beijing, China, in 2022. He is currently pursuing the Ph.D.degree with State Key Lab of Virtual Reality Technology and Systems, Beihang University, Beijing, China. His research interests concern on computer graphics, virtual reality, 3D Reconstruction and localization.

Shengjia Liang is a Ph.D. student at State Key Lab of Virtual Reality Technology and Systems, Beihang University, Beijing, China. He received the B.S. degree in computer science and technology from Harbin Engineering University in 2024. His main research interests are three-dimensional reconstruction and open vocabulary scene understanding.

Yuan Xiong received the M.S. degree in computer science from Clemson University in 2014 and the Ph.D. degree from Beihang University, Beijing, China, in 2024. He is currently pursuing a postdoctoral research position at the School of Cyber Science and Technology, SUN Yat-Sen University · Shenzhen. His research interest includes machine learning, computer vision and virtual reality.

Qiang Zhou is a Ph.D. candidate at State Key Lab of Virtual Reality Technology and Systems, Beihang University, Beijing, China. He received M.S. degree from Beihang University, Beijing, China, in 2007. His research interests include computer vision and video understanding.

Qichuan Geng received the B.S. degree in Automation Science in 2012 and the Ph.D. degree in Technology of Computer Application in 2021 from Beihang University, Beijing, China. He is currently a Lecturer and the Master's Instructor with the Information Engineering College, Capital Normal University. His main research interests include computer vision, artificial intelligence, and scene geometry recovery.

Zhong Zhou received the B.S. degree in material physics from Nanjing University in 1999 and the Ph.D. degree in computer science and engineering from Beihang University, Beijing, China, in 2005. He is currently a Professor and Ph.D. Adviser with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University. His main research interests include virtual reality, augmented reality, computer vision, and artificial intelligence.