Online Language Splatting

Saimouli Katragadda[1], Cho-Ying Wu[2], Yuliang Guo[2*], Xinyu Huang[2], Guoquan Huang[1], Liu Ren[2]

[1]University of Delaware, [2]Bosch Research North America & Bosch Center for Artificial Intelligence (BCAI)

{saimouli,ghuang}@udel.edu

{Cho-Ying.Wu,yuliang.guo2,xinyu.huang,liu.ren}@us.bosch.edu

Abstract

*To enable AI agents to interact seamlessly with humans and 3D environments, they must accurately perceive 3D spaces and align language with spatial representations. While prior work has made significant progress by integrating language features into geometrically detailed 3D scene representations using 3D Gaussian Splatting (GS), these approaches rely on computationally intensive offline preprocessing of language features for each input image, limiting adaptability to new environments. In this work, we introduce **Online Language Splatting**, the first framework enabling near real-time, open-vocabulary language mapping within a 3DGS-SLAM system without requiring pre-generated language features. The key challenge lies in efficiently fusing high-dimensional language features into 3D representations while balancing the computation speed, memory usage, rendering quality and open-vocabulary capability. To this end, we innovatively design: (1) a high-resolution CLIP embedding module capable of generating detailed language feature maps in 18ms per frame, and (2) a two-stage online auto-encoder that compresses 768-dimensional CLIP features to 15 dimensions while preserving open-vocabulary capabilities. Experiments show our online method not only surpasses the state-of-the-art offline methods in accuracy but also achieves more than $40\times$ efficiency boost, demonstrating the potential for dynamic and interactive AI applications.*

1. Introduction

Radiance Fields [7, 13, 16] have transformed 3D scene representation, with 3D Gaussian Splatting (GS) [7] emerging as a leading method for its efficiency and high-quality rendering. However, while radiance fields excel at photorealistic rendering, they lack the semantic information essential for human interaction.

The integration of language features into 3D scene representations has recently enabled open-vocabulary language queries, improving interpretability and interactivity in human-computer interaction [8, 14, 17, 20]. For example, LangSplat [14] embeds CLIP-based language features [15] into 3DGS, including both RGB and language channels per Gaussian. However, existing Lang-GS methods typically rely on computationally intensive preprocessing to generate pixel-wise language features using multimodal foundation models like SAM+CLIP, which can require minutes per frame, limiting their use to offline scenarios where language features must be precomputed.

While offline language mapping is sufficient for static, predefined environments, many real-world applications demand immediate scene understanding. For instance, a service robot entering a new environment must quickly perceive the 3D surroundings to follow commands, and augmented reality (AR) systems need to deliver instant, interactive feedback as users explore new spaces. Recent advancements in combining Gaussian Splatting with online mapping [1, 4, 6, 12, 19] have enabled detailed geometric and textured maps to be created in near real-time. However, these approaches do not incorporate language features, focusing solely on geometry and texture. Alternatively, methods that use pre-annotated ground-truth semantic maps [5, 9, 10] simplify the problem but are limited to closed-vocabulary settings, lacking the flexibility required for open-vocabulary commanding.

The key challenge in online 3D language mapping is efficiently integrating language features while preserving open-vocabulary capabilities. To address this, we introduce **Online Language Splatting**, the first framework to achieve near real-time, open-vocabulary 3D language mapping within a SLAM-GS system, eliminating the need for pre-generated language maps. Fig. 1 illustrates the proposed framework. In particular, our method addresses three core sub-challenges: (1) **Real-time High-Resolution CLIP Embedding:** Since offline, segment-centric CLIP feature preparation is a major runtime bottleneck, we replace it with a single-stage CLIP embedding and a Super-Resolution Decoder (SRD) module, enabling the generation of detailed, pixel-aligned CLIP maps in 18 ms per frame (Sec. 2.1). (2)

*Corresponding Author.

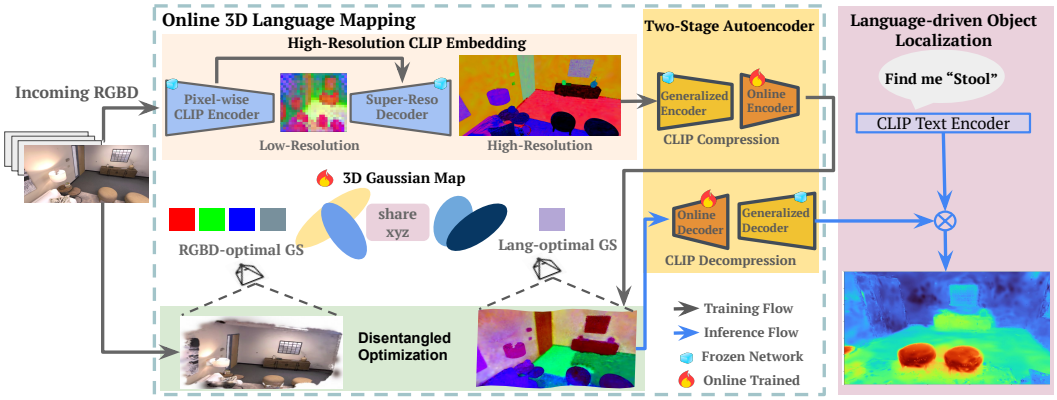

Figure 1. **Online Language Splatting Pipeline.** Our pipeline integrates 3D Gaussian Splatting with SLAM, using 3D Gaussians as the sole mapping elements. **Left**(Training): Raw images pass through a High-Resolution (HR) CLIP embedding module, which generates real-time HR language features. These are compressed via a two-stage CLIP compression module into low-dimensional maps for efficient optimization while preserving open-vocabulary capabilities. **Right**(Inference): The rendered low-dimensional language map is decoded in two-stages to reconstruct the full-resolution CLIP feature map, enabling open-vocabulary object queries.

Open-Vocabulary-Preserving Feature Compression in Novel Scenes: Unlike offline methods, which allow feature compression modules to be trained on the test scene, online methods must generalize to unseen data. However, due to domain gaps, a single pre-trained autoencoder may struggle to maintain open-vocabulary capabilities when compressing CLIP features for online mapping. To address this online-specific generalization challenge, we introduce a two-stage autoencoder, where the second stage, an Online-Learned AutoEncoder (OLAE), dynamically adapts to the dominant data variance of the current scene. This further reduces feature dimensions while preserving critical information (Sec. 2.2).Building on these designs, our extensive experiments demonstrate that our approach not only surpasses prior state-of-the-art (SoTA) offline Lang-GS methods in text-queried 2D and 3D object localization and segmentation but also delivers a $40\times$ to $200\times$ efficiency boost.

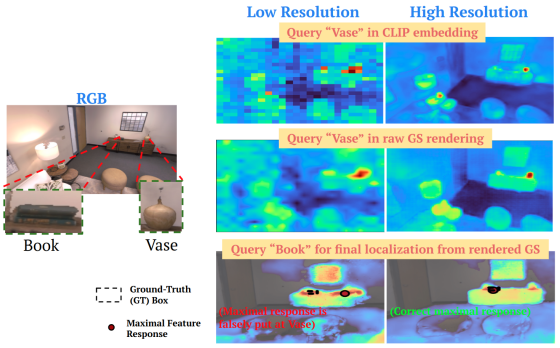

Figure 2. **Importance of Feature Map Resolution. Top:** Compared to the Low Resolution (LR) query heatmap from the pixel-wise encoder output (left), the High Resolution (HR) heatmap from SRD output (right) improves localization and differentiation. **Middle & Bottom:** Query heatmaps from rendered maps after GS mapping. GS mapping from LR exhibits feature bleeding, while mapping from HR preserves structural details, better localization.

2. Online Language Splatting

Our approach enables near real-time, high-resolution, Open-Vocabulary (OV) language mapping within a 3DGS framework, facilitating language-driven spatial understanding for robotics and AR applications. As shown in Fig. 1, our pipeline consists of three main components during the training and optimization phase, addressing the key challenges outlined in Sec. 1.

The pipeline begins with standard RGB-D SLAM input streams. Color images are processed through a pixel-wise CLIP encoder to generate low-resolution language features. These features, combined with hierarchical encoder outputs, are then refined by a Super-Resolution Decoder (SRD) to produce pixel-aligned, high-resolution language maps. Next, the CLIP Compression module, implemented as a Two-Stage Autoencoder, significantly reduces the dimensionality of CLIP features for efficient online mapping while preserving essential information for OV queries. The second stage, an Online-Learned Autoencoder (OLAE), further enhances generalization to novel scenes. Finally, Disentangled Optimization separates gradient flows for color and language, enabling independent optimization of Gaussian parameters. This improves rendering quality across both modalities. During inference, the rendered low-dimensional language map can be passed through the Two-Stage Autoencoder to reconstruct full CLIP features, allowing OV queries for locating target objects.

2.1. High-Resolution CLIP Embedding

Unlike offline methods that require multiple passes and complex mask generation, our approach leverages a ConvNeXt-based pixel-wise CLIP Encoder [18] to generate a coarse CLIP embedding map, which is then refined

by a lightweight **Super-Resolution Decoder (SRD)** to produce dense, high-quality language maps. This design preserves conceptual integrity while enabling real-time operation. The SRD takes a coarse CLIP map along with the intermediate outputs from layers 1 and 2 of the pixel-wise encoder as inputs, progressively enhancing the CLIP feature map resolution through two convolutional upsampling blocks that align with hierarchical encoder features.

The SRD is trained using high-resolution CLIP feature maps as supervision, following [14]. It is dataset-agnostic, requiring only diverse images—without annotations—to learn generalizable upsampling. On rich datasets like COCO [11], this preserves CLIP's open-vocabulary capability. Our CLIP embedding module (pixel-wise encoder + SRD) is efficient, running in 18 ms and using 1.6 GB of memory on an RTX-3090, with SRD contributing just 2 ms while enhancing feature quality. This improves both accuracy and IoU (see Table 1, Fig.2). While similar to FeatUp[3], our SRD uses supervised hierarchical guidance, achieving better performance with greater efficiency.

2.2. Two-Stage Online CLIP Compression

Since CLIP features are high-dimensional (768) vectors, a key challenge is how to effectively compress them to enable real-time integration while preserving OV capabilities.

To address this, we first develop a generalized language compressor that leverages the redundancy in language embeddings. Using diverse datasets (e.g., COCO), we train a simple autoencoder baseline with a multi-layer MLP to compress the dimensionality from 768 to a 32-dimensional code balancing semantic preservation and data compression. Due to the domain gap between the pretraining dataset and the test scenes, the output dimension cannot be too low, as excessive compression may compromise OV capabilities when applied to new domains.

While the generalized language compressor effectively reduces dimensionality, the resulting code size remains too large for efficient integration into an online Lang-GS framework. To further compress the CLIP feature while preserving their OV capability, we introduce an **Online-Learned AutoEncoder (OLAE)** as a second-stage compressor, which adapts dynamically to testing scenes by compressing features into a smaller 15-dimensional code. This leverages the observation that scene-specific variance can often be captured in fewer dimensions, discarding less relevant directions. The OLAE begins with 200 iterations (6 ms/iter) and incrementally updates using selected keyframes. For each iteration, two additional random keyframes are incorporated, ensuring retention of previously learned features and preventing catastrophic forgetting. By combining a generalized compressor (for broad vocabulary preservation) and an online-learned compressor (for scene adaptability), our approach maintains OV capabilities while significantly reducing memory cost, making real-time applications feasible.

Table 1. **Comparison to Lang-GS SoTA on Replica**. Our method is compared to the SoTA Lang-GS methods on the Replica dataset in terms of image-based localization accuracy and per-frame running time. We also analyze the impact of key introduced modules, including Super-Resolution Decoder (SRD) in CLIP Embedding and Online Learning of AutoEncoder (OLAE) in feature compression. The variants without OLAE train a single AE from other scenes in Replica Dataset.

Method	Modules		Query Loc.		Time
	SRD	OLAE	mIOU	Loc	
LangSplat [14]	–	–	0.417	0.720	2.8 min/fr
Feature3DGS [20]	–	–	0.359	0.755	2.3 min/fr
LEGaussian [17]	–	–	0.245	0.682	32 s/fr
Ours	✗	✗	0.400	0.754	**0.8 s/fr**
	COCO	✗	0.475	0.782	
	COCO	✓	0.479	0.759	
	Omni	✗	0.485	0.802	
	Omni	✓	**0.487**	**0.826**	

Table 2. **Comparison to Lang-GS SoTA on TUM RGB-D.** Our method is compared to the Lang-GS SoTA method LangSplat on image-based localization accuracy and running time.

TUM RGB-D	Scene1		Scene2		Time
	mIOU	Loc	mIOU	Loc	
LangSplat [14]	**0.646**	0.850	**0.538**	0.7825	2.1 min/fr
Ours	0.599	**0.917**	0.535	**0.7905**	**0.6 s/fr**

3. Experiments

Baselines. Since we introduce the first online Language Gaussian Splatting (Lang-GS) method, we primarily compare our approach to state-of-the-art (SoTA) offline Lang-GS methods, including LangSplat [14], Feature3DGS[20], and LEGaussian[17], in text-based object localization. Since LEGaussian uses re-annotated ground truth for Replica, we re-evaluate it for fairness. **Datasets.** We evaluate on Replica (synthetic) and TUM RGB-D (real-world) datasets, conducting both qualitative and quantitative analysis. For Replica, we test the 10 most frequent classes, sampling 21 frames per sequence. In TUM RGB-D, we manually annotate test frames to generate ground-truth masks for language queries. Training leverages COCO[11] and Omnidata[2] for broad generalization across diverse scenes and objects. **Evaluation Metrics.** For text-based object localization, we follow LangSplat and use mIoU and localization accuracy (Loc), considering localization successful if the highest-relevancy pixel falls within the ground-truth bounding box. Runtime is measured per-frame.

Implementation Details. We use a pre-trained CLIP ViT-L model [15] with a ConvNeXt-L hierarchical encoder [18] to extract 768D features from 640×640 RGBD images, producing a 24×24×768 feature map. The SRD then enhances

it to 192×192×768, preserving semantic context. We train two SRD models: one on 7% COCO[11] and another on 30% Omnidata-Tiny[2], using four A5000 GPUs (batch size 12 per GPU). For feature compression, we use an 8-layer MLP autoencoder to reduce language features to 32D. The online compressor uses a 2-layer MLP, further compressing to 15D, trained online with Adam (LR: 1×10^{-3}). It initializes with 200 iterations on 10 keyframes and updates with 1 iteration per frame. For fair comparison, we upgrade LangSplat's OpenCLIP to 768D (from 512D), increase code size to 15 (from 3), and train offline on the full Replica and TUM RGBD datasets, sampling every 10th image per sequence.

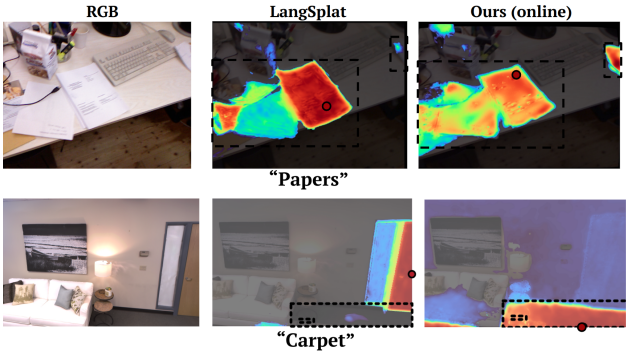

Figure 3. **Qualitative comparison with offline SoTA: Top**: On the TUM RGB-D dataset, our method successfully segments the paper in the top-right corner, which LangSplat fails to detect. **Bottom**: On the Replica dataset, we accurately localize the carpet, whereas LangSplat misidentifies a different object. Black box: ground-truth box; red dot: maximal feature response as the predicted localization.

3.1. Comparison with the State of the Art

Comparison to Lang-GS SoTA Methods. The comparison between our method and previous SoTA offline Lang-GS methods is presented in Table 1 and Table 2. As observed, our method establishes a new SoTA performance on the Replica dataset, significantly surpassing offline methods, regardless of whether SRD is trained on COCO or Omnidata datasets. It also leads to improved localization accuracy and competitive mIoU scores on the TUM-RGBD dataset upon LangSplat. As an online method, our approach is $40\times$ to $200\times$ more efficient than SoTA offline methods. Qualitatively, Fig. 3 shows that our method correctly identifies objects that LangSplat either misses or misidentifies. The ablation study of key modules are further discussed in Sec 3.2. We attribute the performance gains to pixel-aligned features, which are more robust for large objects—where SAM-based segmentation often fails to capture contextual cues that pixel-aligned feature maps preserve.

On the other hand, our performance advantage on the TUM-RGBD dataset is less pronounced. This is primarily due to challenges such as motion blur and lower image quality, which complicate online camera tracking. These conditions favor offline approaches that rely on extensive global optimization (e.g., 30k iterations) of both 3D Gaussian parameters and camera poses.

Runtime Analysis. Our entire network module runs at 23ms per frame on an RTX-3090 GPU, including 15ms for CLIP encoding, 2ms for super-resolution decoding, and 6ms for online compression with online training. Although the overall pipeline speed is bottlenecked by the MonoGS baseline, leading to 0.6–0.8s per frame, a much higher speed can be achieved with advancements in the SLAM-GS system. In contrast, the offline method LangSplat requires approximately 168s per frame (2.8 minutes), including 35s for SAM, 10s for post-processing, and an additional 123s per frame (amortized) for training the dense CLIP autoencoder on the testing scene. This total runtime underscores the significant computational cost of an offline approach.

3.2. Ablation Study

Super-Reso Decoder (SRD) in CLIP Embedding We analyze SRD's impact on the Replica dataset Table 1. SRD significantly improves both mIoU and Loc metrics from our basic online baseline. The underlying reasons for these improvements are evident through visual comparisons in Fig. 2 and Fig. 3. From Fig. 2, we can see that high-resolution language maps greatly enhance localization of small or distant objects.

Online Learning of AutoEncoding (OLAE). Table 1 summarizes OLAE's effect in CLIP compression. To assess the impact of removing online encoding, we train a single autoencoder using 4-fold cross-validation on Replica, holding out two sequences per fold for testing. This ensures exposure to the Replica domain while keeping test scenes unseen. OLAE outperforms even in-domain fine-tuned autoencoders, preserving semantic concepts more effectively.

4. Conclusion

In this work, we introduce Online Language Splatting, a framework that enables online language-aware 3D mapping through key innovations. First, a real-time Super-Resolution Decoder (SRD) enhances CLIP embeddings, generating detailed language maps. Second, an highly effective and efficient two-stage CLIP compression preserving open-vocabulary capabilities. Our experimental results demonstrate that our online approach not only outperforms offline SoTA Lang-GS methods, but also leads to orders of magnitude efficiency improvement.

References

[1] Tianchen Deng, Yaohui Chen, Leyan Zhang, Jianfei Yang, Shenghai Yuan, Jiuming Liu, Danwei Wang, Hesheng Wang, and Weidong Chen. Compact 3d gaussian splatting for dense visual slam. *arXiv preprint arXiv:2403.11247*, 2024. 1

[2] Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10786–10796, 2021. 3, 4

[3] Stephanie Fu, Mark Hamilton, Laura E. Brandt, Axel Feldmann, Zhoutong Zhang, and William T. Freeman. Featup: A model-agnostic framework for features at any resolution. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. 3

[4] Xinli Guo, Peng Han, Weidong Zhang, and Hongtian Chen. Motiongs: Compact gaussian splatting slam by motion filter. *arXiv preprint arXiv:2405.11129*, 2024. 1

[5] Yiming Ji, Yang Liu, Guanghu Xie, Boyu Ma, Zongwu Xie, and Hong Liu. Neds-slam: A neural explicit dense semantic slam framework using 3d gaussian splatting. *IEEE Robotics and Automation Letters*, 2024. 1

[6] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat track & map 3d gaussians for dense rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21357–21366, 2024. 1

[7] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 2023. 1

[8] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19729–19739, 2023. 1

[9] Linfei Li, Lin Zhang, Zhong Wang, and Ying Shen. Gs3lam: Gaussian semantic splatting slam. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 3019–3027, 2024. 1

[10] Mingrui Li, Shuhong Liu, Heng Zhou, Guohao Zhu, Na Cheng, Tianchen Deng, and Hongyu Wang. Sgs-slam: Semantic gaussian splatting for neural dense slam. In *European Conference on Computer Vision (ECCV)*, pages 163–179. Springer, 2024. 1

[11] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014. 3, 4

[12] Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. Gaussian splatting slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18039–18048, 2024. 1

[13] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, pages 102:1–102:15, 2022. 1

[14] 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 (CVPR)*, pages 20051–20060, 2024. 1, 3

[15] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning (ICML)*, pages 8748–8763. PMLR, 2021. 1, 3

[16] Antoni Rosinol, John J Leonard, and Luca Carlone. Nerf-slam: Real-time dense monocular slam with neural radiance fields. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3437–3444. IEEE, 2023. 1

[17] Jin-Chuan Shi, Miao Wang, Hao-Bin Duan, and Shao-Hua Guan. Language embedded 3d gaussians for open-vocabulary scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5333–5343, 2024. 1, 3

[18] Bin Xie, Jiale Cao, Jin Xie, Fahad Shahbaz Khan, and Yanwei Pang. Sed: A simple encoder-decoder for open-vocabulary semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3426–3436, 2024. 2, 3

[19] Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. Gs-slam: Dense visual slam with 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19595–19604, 2024. 1

[20] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21676–21685, 2024. 1, 3