Langsplat study and 4D Langsplat report

Yingwei Song, 2024 Fall

1. 3D query, Arbitrary semantic object removal and inpainting:

1.1. Introduction

LangSplat's original approach to semantic querying relies on rendering 3D scenes into 2D images and performing cosine similarity-based queries in the 2D space. While effective for basic semantic exploration, this approach faces significant challenges for semantic object removal in the 3D Gaussian field. Specifically, LangSplat's semantic supervision is primarily applied to the 2D projections, leaving many internal Gaussian points either unsupervised or sparsely supervised. Consequently, naively deleting Gaussian points based on their semantic similarity to a query word often removes only surface-level points, leaving residual fragments within the object.

To address this, we propose a direct querying mechanism that operates on the language features of Gaussian points, combined with a robust removal strategy that ensures the structural and semantic coherence of object removal in the 3D scene.

1.2. Semantic Querying on Gaussian Points

1.2.1. Cosine Similarity Definition

Instead of relying on rendered 2D images, we directly compute the cosine similarity between the language feature of each Gaussian point and the CLIP embedding of the query word. Given:

$$\mathbf{s}_i \in \mathbb{R}^d \quad \text{(the language feature of the } i\text{-th Gaussian point), and}$$

$$\mathbf{q} \in \mathbb{R}^d \quad \text{(the CLIP embedding of the query word, e.g., "blue elephant"),}$$

the semantic similarity is defined as:

$$\mathcal{S}_i = \frac{\mathbf{s}_i \cdot \mathbf{q}}{\|\mathbf{s}_i\| \|\mathbf{q}\|}.$$

Threshold-Based Filtering

Gaussian points with:

$$\mathcal{S}_i > \tau,$$

where τ is a user-defined threshold, are identified as semantically relevant to the query.

1.3. Convex Hull-Based Removal

1.3.1. Convex Hull Construction

To address the challenges of sparse or incomplete supervision, we leverage convex hull-based grouping inspired by Grouping Gaussian methods. A convex hull is constructed over the Gaussian points satisfying:

$$\mathcal{S}_i > \tau,$$

using standard algorithms (e.g., Quickhull). The convex hull defines the boundary of the object as a closed 3D surface.

Limitations of Convex Hull

While the convex hull effectively captures the core structure of the object, it may fail to include peripheral Gaussian points that:

Figure 1. Left - original Mid- old method Right - novel method with Mask Refinement Process

- Are located near the edges of the object (e.g., thin extensions such as ears or tails).
- Have semantic similarity scores just below τ, despite being visually and semantically connected to the object.

For example, when querying for "blue elephant," the convex hull may exclude points representing the top of the elephant's ears or the edges of its body, leaving residual semantic fragments in the scene.

1.4. Symmetry-Based Sampling for Enhanced Removal

1.4.1. Centroid Calculation

Compute the centroid $\mathbf{c}$ of the convex hull:

$$\mathbf{c} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i,$$

where $\mathbf{x}_i$ are the 3D coordinates of Gaussian points on the convex hull.

1.4.2. Symmetric Location Generation

For each Gaussian point $\mathbf{x}_i$ on the convex hull, generate a symmetric counterpart $\mathbf{x}_i'$ with respect to the centroid:

$$\mathbf{x}_i' = 2\mathbf{c} - \mathbf{x}_i.$$

1.4.3. Peripheral Point Identification

For each $\mathbf{x}_i'$, search within a small radius r for Gaussian points with:

$$\mathcal{S}_j > \tau.$$

These points are likely to belong to the peripheral structure of the object and are marked for removal.

1.4.4. Iterative Expansion

Repeat the process iteratively by treating newly identified points as part of the boundary until no additional points meet the similarity threshold. This approach ensures that both the core and peripheral Gaussian points associated with the queried semantics are removed, leaving minimal residual artifacts.

1.4.5. Inpainting with LaMa

After removing Gaussian points, the resulting scene often contains structural gaps where the object was previously represented. To fill these gaps, we apply LaMa, a state-of-the-art inpainting algorithm. LaMa operates on the remaining Gaussian field to generate plausible replacements for the removed points, ensuring smooth transitions between the removed regions and the surrounding environment.

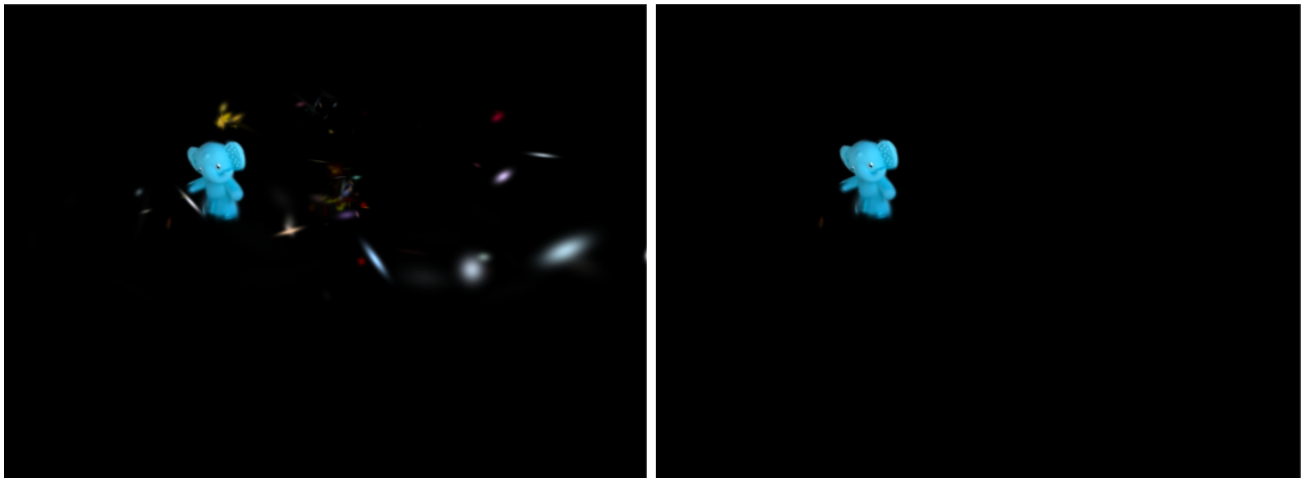

Figure 2. Left: naive mask; Right: centroid-based noise removal

Figure 3. From left to right: original image; without LaMa; with LaMa

1.5. Mask Refinement Process

1.5.1. Centroid-Based Noise Removal

Using the centroid $\mathbf{c}$ of the queried Gaussian points, compute the distances of all candidate Gaussian points from $\mathbf{c}$. Gaussian points that are far away from the centroid and exhibit semantic similarity to the query are considered outliers and removed:

$$\|\mathbf{x}_i - \mathbf{c}\| > r_{\text{threshold}},$$

where $r_{\text{threshold}}$ is a predefined radius.

1.5.2. Ellipsoid-Based Noise Filtering

Recognizing that Gaussian points are modeled as ellipsoids, we further refine the mask by focusing on 2D projections of the scene. Specifically, we remove points that are:

- Far away from the centroid in the 2D projection.
- Similar in semantics but lack coherent spatial relationships with the queried object.

By leveraging the ellipsoidal nature of Gaussian points, this operation ensures that spurious points in the mask are eliminated.

By applying the above refinements, we create a clean, well-defined mask that effectively isolates the target object and removes surrounding noise. This refined mask significantly enhances the performance of LaMa, allowing it to seamlessly reconstruct the removed region with high visual fidelity. Our approach achieved remarkable results in extracting objects

with arbitrary semantics from complex 3D scenes. For example, objects queried with specific semantics, such as "blue elephant," could be almost perfectly removed from the scene without leaving residual artifacts. This success is largely attributed to the robust semantic field constructed by LangSplat, which ensures precise semantic querying, and the noise reduction techniques that refine the inpainting process. Inpainting quality heavily depends on the precision of the mask used to guide the inpainting process. A poorly defined mask with noise or irrelevant regions can lead to unnatural or inconsistent inpainting results. To ensure high-quality inpainting, we build upon techniques from Feature3DGS and Gaussian Grouping to refine the mask generated during the semantic querying process. These refinements help in removing unwanted Gaussian points that could interfere with the inpainting process.

2. Vector Quantization:

2.1. Vector Quantization-Based Dimensionality Reduction

LangSplat leverages an autoencoder to reduce the dimensionality of CLIP-generated semantic features from 512-dimensional to 3-dimensional for each Gaussian point, resulting in an $N \times 3$ matrix of language features, where N is the total number of Gaussian points. While this reduction is efficient for storage, the inference phase requires decoding the $N \times 3$ matrix back into $N \times 512$, which incurs a significant memory overhead due to the large value of N (often in the range of millions). This limitation is particularly pronounced during rendering tasks, where CUDA memory is a critical resource.

To alleviate this bottleneck, we introduce a Vector Quantization (VQ)-based approach inspired by VQ-VAE. This method replaces continuous $N \times 3$ feature storage with a compact codebook representation, reducing memory consumption without significant loss of semantic fidelity.

2.2. Codebook Definition and Initialization

We define a codebook $\mathbf{C} \in \mathbb{R}^{K \times 3}$, where K is the number of discrete entries in the codebook. Each entry represents a cluster center in the 3-dimensional language feature space. The primary goal is to approximate the original $N \times 3$ feature matrix using K representative features, thereby compressing the storage requirements. The codebook is initialized randomly and updated during training to minimize the reconstruction error of the original language features.

2.3. Feature Assignment

For each Gaussian point i with language feature $\mathbf{s}_i \in \mathbb{R}^3$, we find the closest codebook entry using:

$$j^* = \arg\min_j \|\mathbf{s}_i - \mathbf{C}_j\|^2, \quad j \in \{1, \ldots, K\}.$$

The index j^* is stored as the quantized representation of $\mathbf{s}_i$.

2.4. Reconstruction

During inference, the original feature $\mathbf{s}_i$ is reconstructed from the codebook as:

$$\hat{\mathbf{s}}_i = \mathbf{C}_{j^*}.$$

The reconstructed features are then decoded into $N \times 512$ using the pre-trained decoder of the original autoencoder.

3. 4D Langsplat

3.1. Introduction

We propose 4D LangSplat, a novel framework for learning 4D language fields in dynamic scenes, enabling precise, time-sensitive, and open-vocabulary queries. Unlike LangSplat, which handles static 3D scenes, 4D LangSplat leverages Multimodal Large Language Models (MLLMs) to generate temporally consistent, object-wise captions, encoded into sentence embeddings for feature supervision. A status deformable network captures smooth object transitions over time, achieving state-of-the-art performance on dynamic scene benchmarks.

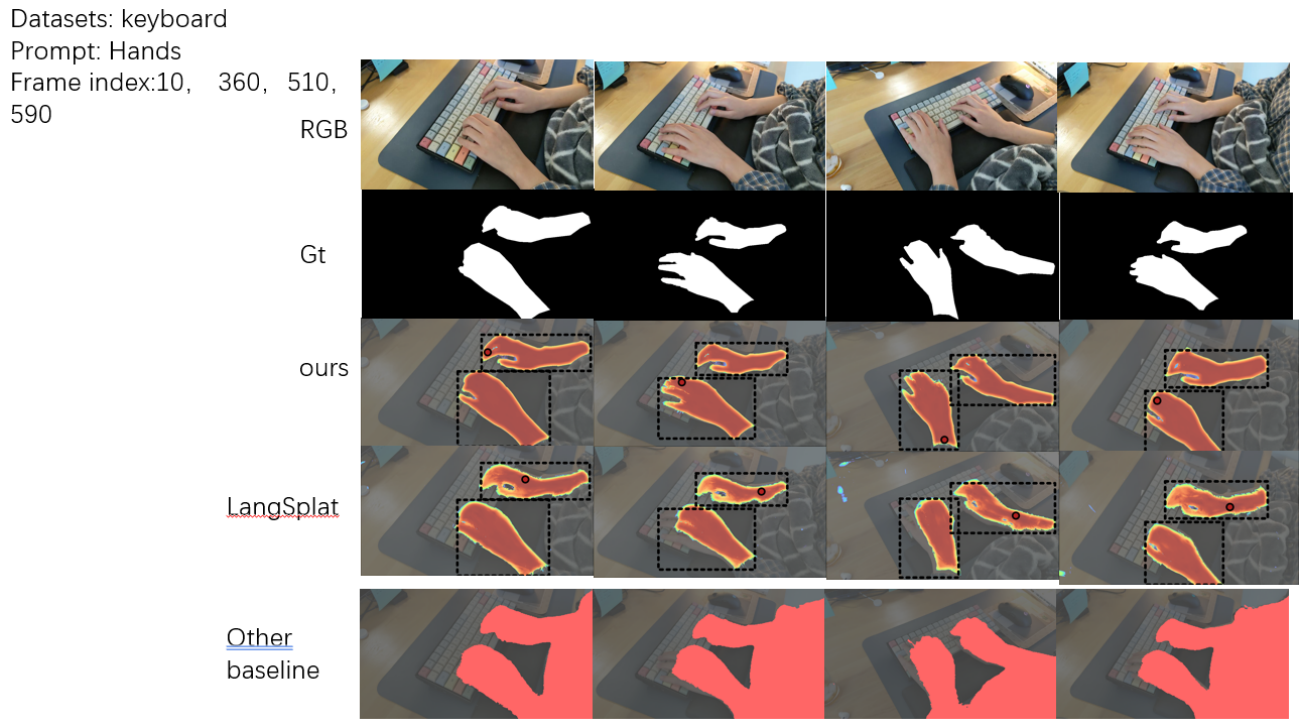

Figure 4. 4D Langsplat Comparison

3.2. Contribution

In this project, I was primarily responsible for extensive dataset annotation, conducting ablation studies on Feature3DGS for comparisons in different dynamic datasets including dynerf and hypernerf, as well as contributing to the writing and revision of the final paper.

4. Thanks and Words

I want to express my heartfelt gratitude to Prof. James Tompkin and Post-Doc Wanhua Li for their invaluable support during my year at Brown and Harvard VCG. This year has been a transformative journey, one that has not only shaped my understanding of research but also profoundly influenced my personal growth.

I've always struggled with procrastination, and yet, both Prof. Tompkin and Wanhua showed immense patience and understanding. I cannot thank Prof. Tompkin enough for introducing me to the fascinating worlds of NeRF and 3DGS. Before this, I had no exposure to these fields and little understanding of how research truly works. His guidance opened up a new horizon for me, and for that, I will always be grateful.

Equally, meeting Wanhua has been a turning point in my academic life. His unwavering support, patience, and willingness to mentor me through every step of this journey have left an indelible mark. He has not only taught me the technical aspects but also helped me navigate the challenges and uncertainties of research. Wanhua is, without a doubt, one of the most important mentors I've ever had.

I am deeply appreciative of their kindness in writing recommendation letters for my 2025 Fall PhD applications. I know how tedious and time-consuming this process can be, and their effort means the world to me. Their belief in me is a source of strength and motivation as I take the next steps.

As I look to the future, I find myself at a crossroads. Will I continue to explore the depths of computer vision and 3D reconstruction, or will I pivot towards medical imaging, perhaps working with CT scans to make a tangible difference in healthcare? I'm still figuring it out, and this uncertainty is both daunting and exhilarating.

What I do know is that the experiences I've had at Brown and Harvard this past year have been nothing short of life-changing.

They've given me a glimpse of what's possible and a foundation to build upon. From the bottom of my heart, I thank Prof. James Tompkin and Post-Doc Wanhua Li for their unwavering guidance, their patience, and their belief in me. Without them, this journey wouldn't have been the same.