LangSplat

> LangSplat: 3D Language Gaussian Splatting

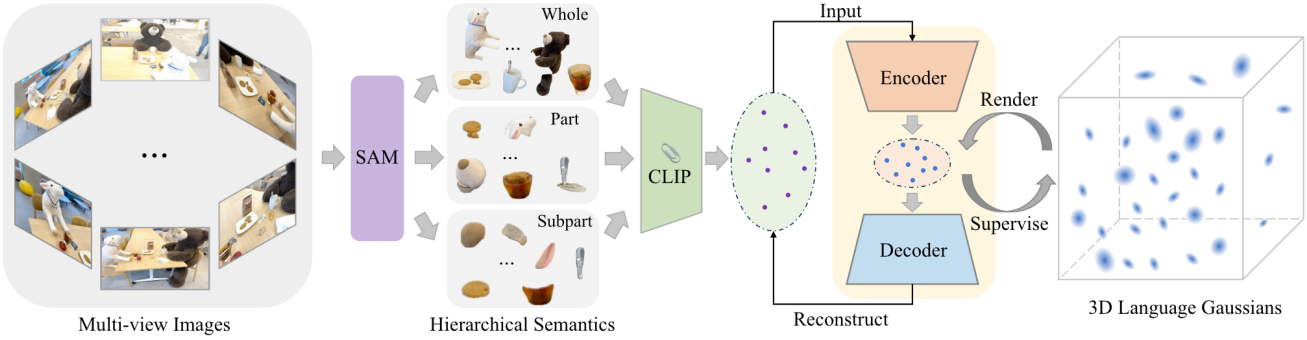

Figure 2. The framework of our LangSplat. Our LangSplat leverages SAM to learn hierarchical semantics to address the point ambiguity issue. Then segment masks are sent to the CLIP image encoder to extract the corresponding CLIP embeddings. We learn an autoencoder with these obtained CLIP embeddings. Our 3D language Gaussian learn language features on the scene-specific latent space to reduce the memory cost. During querying, the rendered language embeddings are sent to the decoder to recover the features on the CLIP space.

这篇文章的主要流程包括以下几个步骤，涵盖了每个公式和思路。

1. 3D高斯点分布表示

在这篇文章中，**3D高斯点分布**被用来表示场景中的对象或区域。每个3D高斯点表示场景中的一个体积元素，拥有以下几个特性：

- **位置**：表示该点在3D空间中的位置。
- **协方差矩阵**：控制该点的分布范围和形状，从而表示点在空间中的扩展。

- **特征嵌入**：每个高斯点包含附加的语言特征嵌入，用于表达对象的语义信息。

这些3D高斯点的分布可用公式表示为：

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

其中，μ 是高斯点的中心位置，Σ 是协方差矩阵，表示该点在空间中的分布。

---

2. 自动编码器压缩（公式5）

为了降低计算成本和内存占用，文章使用了自动编码器来将高维语言特征压缩到低维空间。这部分的损失函数被定义为：

$$\mathcal{L}_{ae} = \sum_{l\in\{s,p,w\}} \sum_{t=1}^{T} d_{ae}(\Psi(E(L_t^l(v))), L_t^l(v)) \quad (2)$$

- $L_t^l(v)$：表示在时间 t 和层次 l 下像素 v 的原始 CLIP 嵌入特征。
- E 和 Ψ** 分别是编码器和解码器函数，将高维的 CLIP 嵌入特征映射到低维潜在空间，再映射回原始空间。
- d_{ae}：距离函数，用于计算重构误差。

这一步确保了在减少存储成本的同时，保持语言嵌入的语义完整性。

---

3. 语言嵌入相关性得分计算（公式见图片）

为了在3D空间中找到与查询文本最相关的区域，文章定义了一个相关性得分，通过以下公式来计算每个渲染的语言嵌入 ϕ_{img} 与查询嵌入 ϕ_{qry} 的相关性：

$$\min_i \frac{\exp(\phi_{\text{img}} \cdot \phi_{\text{qry}})}{\exp(\phi_{\text{img}} \cdot \phi_{\text{qry}}) + \exp(\phi_{\text{img}} \cdot \phi^i_{\text{canon}})} \tag{3}$$

- ϕ^i_{canon}：CLIP嵌入的预定义标准短语，取自"object"、"things"、"stuff" 和 "texture"。
- $\min$：用于在多个语义类别中选择最具区分性的匹配度。

这种得分计算方法有助于过滤掉无关的类别，从而更精确地找到符合查询语义的3D区域。

---

4. 分层的3D相关性图

对每个文本查询，生成三个不同语义层次的相关性图，分别对应子部分、部分和整体。通过LERF的方法，选择产生最高相关性得分的语义层次。每个层次的得分图用来进一步精确定位或分割目标。

---

5. 语言损失（公式6）

为了让模型更好地理解语言特征，文章还定义了语言损失函数 $\mathcal{L}_{lang}$，用于优化嵌入特征，使其符合目标语义。公式如下：

$$\mathcal{L}_{lang} = \sum_{l \in \{s,p,w\}} \sum_{t=1}^{T} d_{lang}(F_t^l(v), H_t^l(v)) \qquad (4)$$

- $F_t^l(v)$：渲染的3D语言场在像素 v 处的嵌入特征。
- $H_t^l(v)$：通过自动编码器压缩后的低维语言嵌入特征。
- d_{lang}：用于计算语言特征之间差异的距离函数。

该损失函数确保模型生成的语言嵌入在不同的语义层次上与目标语言特征一致。

---

6. 应用于3D对象定位和语义分割

- **3D对象定位**：选择具有最高相关性得分的点作为目标对象的位置。

- **3D语义分割**：对每个点的相关性得分进行筛选，过滤掉得分低于阈值的点，生成3D语义分割掩码。

总结

1. **3D高斯点分布**：通过各向异性3D高斯表示场景中的点。
2. **自动编码器压缩**：压缩高维语言嵌入特征，减小计算和存储开销。
3. **相关性得分计算**：计算渲染嵌入和查询嵌入的相关性，以定位目标区域。
4. **分层相关性图**：在不同语义层次上生成相关性图，找到最相关的层次。
5. **语言损失优化**：通过语言损失函数，优化嵌入特征的语义一致性。
6. **3D对象定位和分割**：根据相关性得分，精确定位或分割3D空间中的目标区域。