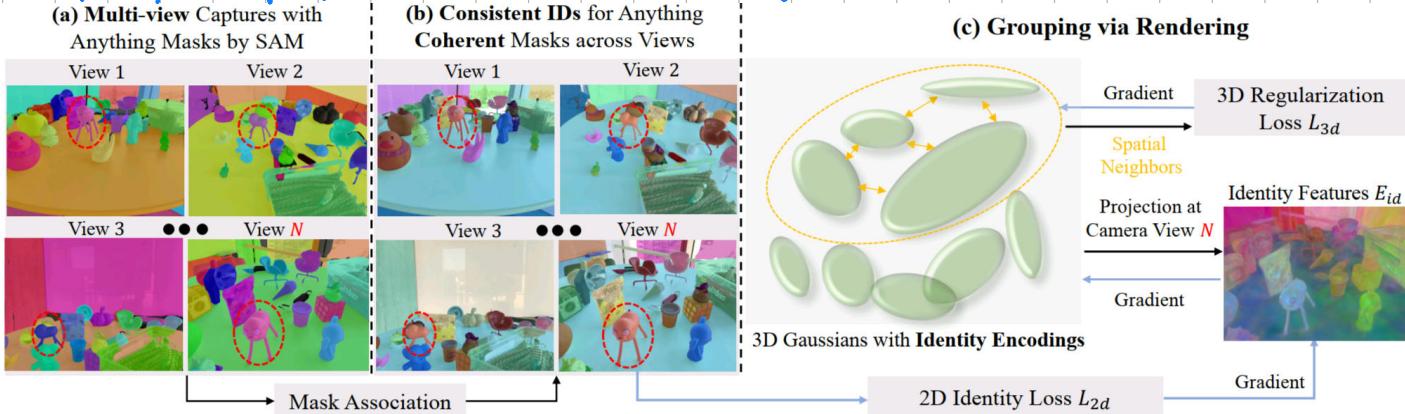


3D Gaussian Grouping

为每个高斯添加身份标识，通过2D+3D双监督实现分组，无需人工标注3D掩码



(a) 输入预处理：多视角RGB图像 + SAM自动生成的2D“任意目标”掩码

(b) 跨视角掩码关联：生成全局一致的目标ID

(c) 联合训练：通过可微渲染优化高斯的几何、外观参数及新增的“身份编码”实现重建与分组

输入数据预处理：

Step 1. 生成2D掩码：

对每个视角的RGB图像，用SAM的“everything mode”自动生成密集的2D掩码。

Step 2. 跨视角掩码关联：

问题：SAM单独为每个视角生成掩码，同一物体在不同视角的掩码ID不一致，无法支撑3D分组。

将多视角图像视为视角渐变的视频序列，用零样本跟踪模型(DEVAN)进行掩码传播与关联，为同一物体分配唯一ID，生成全局一致的多视角分割标签

身份编码 (Identity Encoding)

让每个高斯具备“语义身份”，能够被归类到对应的物体/类别组

关键属性：

维度：16维，实验验证下16维平衡了分割结果与计算效率

表示形式：与颜色(SH)类似，作为高斯的可学习参数。

优化目标：同一物体高斯具有相似编码，不同物体高斯编码差异显著

视角无关性：将SH度数设为0，仅建模直流量，保证同一物体身份编码在不同视角下一致

3D高斯的渲染与分组

通过可微渲染将3D高斯的身份编码投影到2D图像，实现“3D分组”的监督信号对齐

(1) 沿用3DGGS的可微渲染框架，将每个高斯的身份编码 e_i 与几何、外观参数一起投影到2D平面

(2) 计算每个高斯对2D像素的影响权重 α_i' ($\sum \rightarrow \sum'$, 结合 α_i 不透明度)

(3) 每个像素最终的身份特征 E_{id} 是该像素覆盖的所有高斯编码的加权和

$$E_{id} = \sum_{i \in N} e_i \alpha_i' \prod_{j=1}^{N-1} (1 - \alpha_j')$$

分组损失函数 L_{id}

1. 2D Identity Loss

基于渲染的2D像素身份特征 E_{id} ，对齐跨视角关联后的2D掩码标签

添加一个线性层 f ，将16维的 E_{id} 映射到 K 维 (K 为场景目标总数)，再通过 softmax 做身份分类，采用交叉熵损失 L_{2d} 监督

利用SAM掩码的强2D分割能力，为3D分组提供核心监督

2. 3D Regularization Loss (3D正则化损失)

问题：仅用2D损失时，物体内部或被严重遮挡的高斯（未被2D视图覆盖），无法得到有效监督。

会出现透明物体的问题(如图7)

利用3D空间的一致性，强制“空间上邻近的高斯属于同一物体”

$$\mathcal{L}_{3d} = \frac{1}{m} \sum_{j=1}^m D_{kl}(P||Q) = \frac{1}{mk} \sum_{j=1}^m \sum_{i=1}^k F(e_j) \log \left(\frac{F(e_j)}{F(e'_i)} \right)$$

e_j —采样高斯的编码， e'_i —3D高斯中 k 个最邻近高斯的编码
 F —与2D损失共享的“线性层”+“softmax”操作

总损失函数：

$$\mathcal{L}_{\text{render}} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{id}} = \mathcal{L}_{\text{rec}} + \lambda_{2d} \mathcal{L}_{2d} + \lambda_{3d} \mathcal{L}_{3d}$$

1.6 2.0

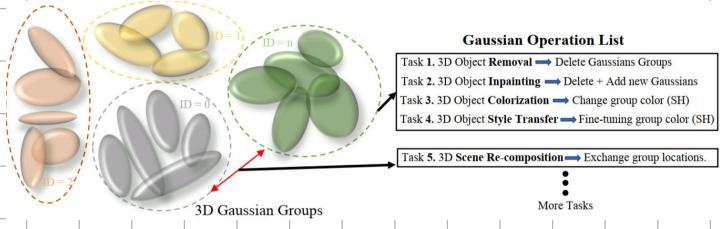
算法流程

Algorithm 1 Gaussian Grouping

```

 $p \leftarrow \text{SfM Points}$                                  $\triangleright 3D \text{ Positions}$ 
 $m = (m_1, m_2, \dots, m_K) \leftarrow \text{SAM}$            $\triangleright \text{SAM's Masks at Various } K \text{ Views}$ 
 $(\hat{M}_1, \hat{M}_2, \dots, \hat{M}_K) \leftarrow \text{Zero-shot Tracking}(m)$   $\triangleright \text{Multi-view Associated Masks}$ 
 $s, \alpha, c, e \leftarrow \text{InitAttributes}()$             $\triangleright \text{Covariances, Opacities, Colors, Identity Encodings}$ 
 $i \leftarrow 0$                                           $\triangleright \text{Iteration Count}$ 
while not converged do
     $V, I, M \leftarrow \text{SampleTrainingView}()$ 
     $I, E_{\text{id}} \leftarrow \text{Rasterize}(p, s, a, c, e, V)$ 
     $\mathcal{L}_{\text{image}} \leftarrow \mathcal{L}(I, I)$ 
     $\mathcal{L}_{\text{id}} \leftarrow \lambda_{2d} \mathcal{L}_{\text{id}}(\mathcal{E}_{\text{id}}, \hat{M}) + \lambda_{3d} \mathcal{L}_{\text{3d}}(e)$ 
     $\mathcal{L} \leftarrow \mathcal{L}_{\text{image}} + \mathcal{L}_{\text{id}}$ 
     $p, s, a, c, e \leftarrow \text{Adam}(\nabla \mathcal{L})$ 
    if IsRefinementIteration(i) then
        for all  $J$  Gaussians  $(p_j, s_j, \alpha_j, c_j, e_j)$  in  $(p, s, a, c, e)$  do
            if  $\alpha_j < \epsilon$  or IsTooLarge( $p_j, s_j$ ) then
                RemoveGaussian()
            end if
            if  $\nabla_{p_j} > \tau_p$  then
                if  $|S| > r_S$  then
                    SplitGaussian( $p_j, s_j, \alpha_j, c_j, e_j$ )
                else
                    CloneGaussian( $p_j, s_j, \alpha_j, c_j, e_j$ )
                end if
            end if
        end for
         $i \leftarrow i + 1$ 
    end while

```



初始化：输入SfM得3D点，SAM生成多视角2D掩码，高斯参数随视角掩码关联：用采样点跟蹤得一致ID的多视角掩码前， \hat{M}_K

训练：

采样训练视角，图像和掩码

可微渲染得图像 I 和身份特征 E_{id}

计算损失

反向传播更新参数

精炼迭代：修剪，分裂，克隆

收敛后输出

Limitation

仅支持静态场景

依赖外部模型精度：开放词汇分割依赖 SAM, DEFA