

ChildSplat: Anchor-Child Semantic Gaussian Representation for 3D Object Segmentation

Haochuan Dang¹, Junke Zhu¹, Jinshuo Wang¹, Siyuan Zhang¹, Boyan Du¹, Zhangjin Huang^{†1}, Lu Zou²,

¹University of Science and Technology of China, Hefei, China, ²Wuhan Institute of Technology, Wuhan, China,

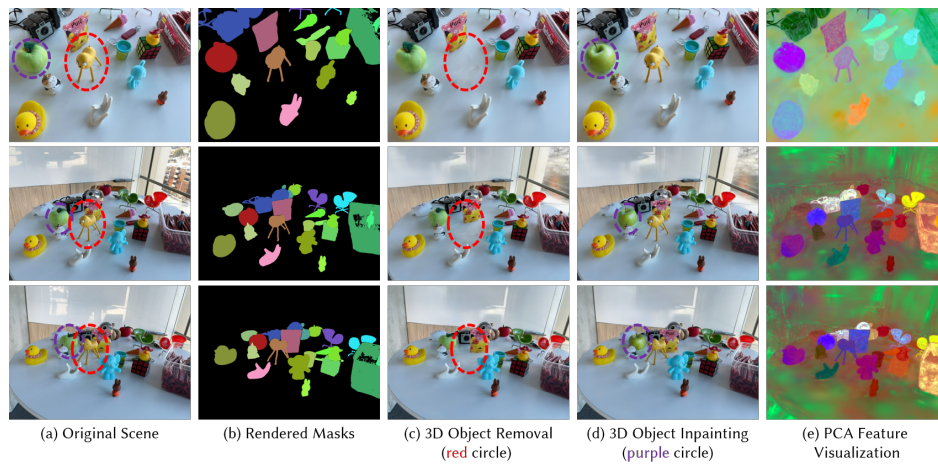


Figure 1: Overview of ChildSplat. Given multi-view images and 2D masks, our method learns view-consistent 3D semantic groups for mask rendering, feature visualization, and downstream local editing.

Abstract

Object-level understanding of reconstructed 3D scenes remains challenging without costly 3D annotations. Recent mask-lifting methods use 2D foundation-model masks to supervise 3D segmentation, while structured Gaussian representations such as Scaffold-GS provide a useful prior for balancing compact semantics and detailed appearance. However, existing Scaffold-GS-based semantic methods often rely on anchor- or region-level shared features, which improve compactness but can be too coarse for object boundaries and fine-grained parts. We propose ChildSplat, an anchor-child semantic Gaussian representation for 3D object segmentation with downstream editing handles. Starting from a pretrained Scaffold-GS scene, we freeze the geometry and appearance fields and learn a lightweight semantic branch on its anchor-child structure. Anchors encode compact semantic context, while child Gaussians obtain geometry-aware residual semantic embeddings and are rendered with the frozen splatting weights. We further derive child-level ownership supervision from multi-view mask contributions, explicitly linking 2D mask evidence to individual child Gaussians to improve fine-grained semantic ownership and reduce cross-object leakage in child-level selection. When hierarchical annotations are available, residualized leaf supervision and lightweight hierarchy-aware constraints enable the same semantic field to support both object-level and part-level queries. Experiments show that our method learns compact, view-consistent 3D semantic groups, which can serve as semantic handles for downstream local editing.

CCS Concepts

• Computing methodologies → Computer graphics; Rendering; Computer vision tasks;

1. Introduction

Learning editable and semantically meaningful 3D scene representations from posed images is a fundamental problem in computer

[†] Corresponding author. Email: zhuang@ustc.edu.cn

graphics and vision, with broad applications in immersive content creation, robotic interaction, and scene editing. Recent neural scene representations have achieved impressive progress in photorealistic novel-view synthesis. In particular, 3D Gaussian Splatting (3DGS) represents a scene with explicit anisotropic Gaussians and renders them through efficient differentiable splatting, enabling high-quality real-time rendering [KKLD23]. Scaffold-GS further improves the compactness and robustness of Gaussian representations by organizing local neural Gaussians with a structured anchor-child design, where sparse anchors spawn view-adaptive child Gaussians [LYX*24]. However, these representations are mainly optimized for appearance reconstruction. For interactive scene understanding and editing, the reconstructed scene should also provide object-level and part-level structures that can be queried, segmented, and manipulated.

Open-world 3D object segmentation addresses this need by decomposing a reconstructed scene into meaningful object or group regions without relying on a fixed semantic taxonomy. A practical way to obtain supervision is to lift 2D foundation-model predictions, such as SAM masks and tracked multi-view object masks, into 3D representations [KMR*23, YDYK24]. Recent methods augment NeRF or Gaussian fields with language features, instance embeddings, or identity encodings, enabling 3D semantic querying and mask prediction from 2D supervision [KKG*23, KMS22, QLZ*24, YDYK24, WML*24]. However, most Gaussian-based approaches either operate on unstructured 3DGS primitives or attach semantic attributes directly to individual Gaussians. As a result, they do not fully exploit the structural prior provided by the anchor-child organization of Scaffold-GS. How to leverage this structure for compact yet fine-grained semantic learning remains underexplored.

The key difficulty lies in the granularity mismatch between appearance and semantics. A scene often requires many local Gaussians to model texture, opacity, and view-dependent effects, while the semantic identity should remain stable within the same object or part [LWM*25]. In Scaffold-GS, anchors provide compact region-level support, but assigning a single semantic embedding to an anchor can be too coarse around object boundaries and small parts. In contrast, assigning independent semantic parameters to every child Gaussian improves local granularity, but introduces redundancy and weakens the compact scaffold structure. An effective semantic Scaffold-GS representation should therefore preserve compact anchor-level context while enabling child-level semantic prediction and supervision. This motivates us to decouple compact semantic context from fine-grained child ownership within the structured Gaussian scaffold: anchors provide stable semantic context, while child Gaussians receive local semantic ownership cues for boundary-aware prediction.

To address these challenges, we propose a semantic Scaffold-GS framework for mask-supervised open-world 3D object segmentation. Our method attaches a lightweight semantic branch to a pretrained Scaffold-GS scene while keeping its geometry and appearance fields fixed. Anchors encode compact semantic context, and child Gaussians receive locally refined semantic embeddings for fine-grained mask prediction. Multi-view 2D mask supervision is further consolidated into child-level semantic learn-

ing, encouraging view-consistent object grouping while retaining the compact structured representation of Scaffold-GS. To support hierarchical queries, we extend the same semantic child field with hierarchy-aware supervision, enabling unified object-level and part-level mask prediction within the same structured 3D representation [QLZ*24, YYZ*23, ZZWW26]. Our code will be released at github.com/Anonymous5i/semantic-scaffold-gs.

Our contributions are summarized as follows:

- We propose an anchor-child semantic factorization for Scaffold-GS, where anchors store compact semantic context and child Gaussians obtain geometry-aware residual embeddings for local semantic ownership.
- We introduce multi-view child ownership supervision that converts frozen splatting contributions and 2D masks into reliable child-level pseudo labels, providing direct 3D semantic supervision beyond rendered 2D mask losses.
- We demonstrate a preliminary hierarchy-aware extension of the same semantic child field for object/part queries when hierarchical annotations are available, and provide transparent mask-supervision protocols including a SAM3-assisted LERF-Obj setting and 3D-OVS supervision details.

2. Related Work

3D Gaussian Splatting. 3D Gaussian Splatting (3DGS) represents a scene with explicit anisotropic Gaussians and renders them through differentiable rasterization, achieving real-time novel-view synthesis with high visual fidelity [KKLD23]. Its explicit primitive-based formulation also makes it attractive for downstream 3D tasks, as each Gaussian can carry additional identity, language, or semantic attributes [YDYK24, QLZ*24, WML*24, CFY*25, CSK*25, LWM*25]. However, vanilla 3DGS optimizes freely distributed Gaussians, which provides limited structural organization for compact semantic modeling. Scaffold-GS addresses this issue with a structured anchor-child representation, where sparse anchors generate local neural Gaussians in a view-adaptive manner [LYX*24]. InstanceGaussian further extends this structure with a Semantic-Scaffold-GS representation, progressive appearance-semantic joint training, and bottom-up instance aggregation, where children associated with the same anchor can model different appearances while sharing semantics [LWM*25]. In contrast, ChildSplat freezes the pretrained appearance and geometry fields and refines semantics at the child level, allowing children from the same anchor to receive distinct semantic assignments under explicit child-ownership supervision. Our work builds on Scaffold-GS and leverages its anchor-child structure for compact yet fine-grained mask-supervised semantic learning.

2D-Supervised 3D Scene Understanding. Learning 3D semantics from 2D supervision has become a practical paradigm for open-world scene understanding, as it avoids costly 3D annotations while benefiting from strong 2D foundation models [KMR*23, CTM*21, RKH*21]. Early methods distill semantic or vision-language features into NeRF-like representations for language-guided interaction, open-vocabulary localization, and 3D segmentation [KMS22, KKG*23, LZZ*23, PGJ*23, JKG*23]. These methods demonstrate the potential of 2D-supervised 3D understanding,

but NeRF-like representations can be computationally expensive and often produce over-smoothed semantic responses near object boundaries.

With the emergence of 3DGS, recent methods learn semantic, language, or instance features on explicit Gaussian primitives. LangSplat learns compact language features on Gaussians using SAM-derived masks and a scene-wise autoencoder, OpenGaussian associates 2D masks and language features with 3D Gaussians, and Gaussian Grouping uses SAM-generated and tracked multi-view masks to supervise open-world Gaussian grouping [QLZ*24, WML*24, YDYK24]. Recent approaches further target cross-view consistency and object-level grouping: ILGS aligns language embeddings using identity cues and progressively expands masks to refine semantic boundaries; Unified-Lift introduces an object-aware Gaussian representation with Gaussian-level features and a learnable object codebook for end-to-end 2D-to-3D segmentation; and Gaga associates inconsistent class-agnostic 2D masks across diverse viewpoints through a 3D-aware memory bank [JK25, ZQL*25, LLK*24]. These works show that 2D foundation-model predictions, especially masks, provide strong and scalable supervision for 3D semantic learning. In a related but different setting, IGGT [LZL*26] jointly models geometric reconstruction and instance-grounded scene understanding using a unified transformer with 3D-consistent contrastive supervision. Unlike methods that mainly operate on NeRF fields or unstructured 3DGS primitives, our work exploits the structured anchor-child representation of Scaffold-GS to decouple compact semantic context from fine-grained child-level ownership.

Hierarchical 3D Segmentation and Scene Editing. Beyond flat object or group segmentation, real scenes often require hierarchical understanding, where objects and parts should be queried at different semantic levels. Recent works explore hierarchical or multi-level 3D semantics using multi-scale masks, scale-aware grouping, open-vocabulary features, or contrastive learning [KWK*24, YYZ*23, QLZ*24, ZZWW26]. These methods demonstrate the importance of multi-level 3D semantics, but most focus on language-driven querying, feature hierarchy, or scale-aware grouping. In contrast, our hierarchy extension builds on a mask-supervised semantic child field and uses residualized leaf labels to handle overlapping parent-part annotations within the structured Scaffold-GS representation.

Semantic 3D groups also provide explicit handles for local scene editing. Prior NeRF- and Gaussian-based methods use learned groups or traced Gaussians for removal, inpainting, recoloring, and recomposition [MAAD*23, CCZ*24, YDYK24]. Similarly, our learned semantic groups serve as object-level selection handles for the Gaussian Grouping-style 2D-inpainting-and-3D-lifting editing pipeline, while our contribution focuses on learning compact and fine-grained semantic groups on Scaffold-GS.

3. Overview

Given multi-view images and 2D object masks, our goal is to learn a compact yet fine-grained semantic field on top of a pretrained Scaffold-GS scene. We freeze its geometry and appearance fields and optimize only a lightweight semantic branch. For a frozen

scene $\mathcal{S}$ and camera view π , the semantic branch produces a semantic logit map

$$\Phi_{\text{sem}}(\mathcal{S}, \pi) = \mathbf{Z}\pi \in \mathbb{R}^{H \times W \times C}, \quad (1)$$

where H , W , and C denote the image height, image width, and number of semantic labels, respectively.

Following Scaffold-GS [LYX*24], for anchor position $\mathbf{x}_i$, scaling vector $\mathbf{l}_i$, and pretrained child offset $\mathbf{o}_{i,k}$, the child center is

$$\mu_{i,k} = \mathbf{x}_i + \mathbf{l}_i \odot \mathbf{o}_{i,k}, \quad (2)$$

where $\odot$ denotes element-wise multiplication.

Anchors store compact semantic context, while geometry-aware residuals refine semantics at the child level. As shown in Fig. 2, Stage 1 learns the anchor semantic scaffold from rendered 2D mask supervision; Stage 2 freezes it and learns child residuals using rendered masks and child-level ownership cues. When hierarchical annotations are available, hierarchy-aware supervision further extends the same child field to object-part queries.

4. Method

4.1. Anchor-level Semantic Scaffold Representation

We first augment each anchor with a learnable semantic feature $\mathbf{f}_i^s \in \mathbb{R}^{d_s}$, which is independent of the appearance feature used for RGB reconstruction. This decoupling allows semantic features to remain stable across views and local appearance variations, while leaving the pretrained appearance branch unchanged.

For each anchor, we construct a semantic input by combining its semantic feature with the normalized anchor position:

$$\bar{\mathbf{x}}_i = \gamma(\mathbf{x}_i), \quad \mathbf{u}_i = [\mathbf{f}_i^s, \bar{\mathbf{x}}_i], \quad \mathbf{e}_i = M_a(\mathbf{u}_i), \quad (3)$$

where $\gamma(\cdot)$ denotes fixed coordinate normalization applied to the frozen anchor positions, $M_a: \mathbb{R}^{d_s+3} \rightarrow \mathbb{R}^{d_e}$ is the anchor semantic MLP, and $\mathbf{e}_i \in \mathbb{R}^{d_e}$ is the anchor semantic embedding. Thus, the anchor position is used only as a fixed geometric condition for semantic learning and is not updated by semantic losses. The resulting anchor-level semantic scaffold stores compact semantic context on top of the frozen Scaffold-GS structure and provides the semantic basis for subsequent child-level refinement.

All semantic embeddings are mapped to class logits by a shared linear classifier:

$$\mathcal{C}(\mathbf{e}) = \mathbf{W}\mathbf{e} + \mathbf{b}, \quad \mathcal{C}: \mathbb{R}^{d_e} \rightarrow \mathbb{R}^C, \quad \mathbf{W} \in \mathbb{R}^{C \times d_e}, \quad \mathbf{b} \in \mathbb{R}^C, \quad (4)$$

where C denotes the number of semantic labels or object groups. The same classifier is used for anchor embeddings, refined child embeddings, and rendered embedding maps. When applied to a rendered embedding map, it is implemented as a 1×1 convolution, which is equivalent to pixel-wise linear classification. Sharing this classifier aligns anchor-level, child-level, and rendered 2D predictions in a common semantic label space.

4.2. Geometry-aware Child Semantic Refinement

The anchor-level semantic scaffold provides a compact semantic representation, but assigning the same semantic embedding to all

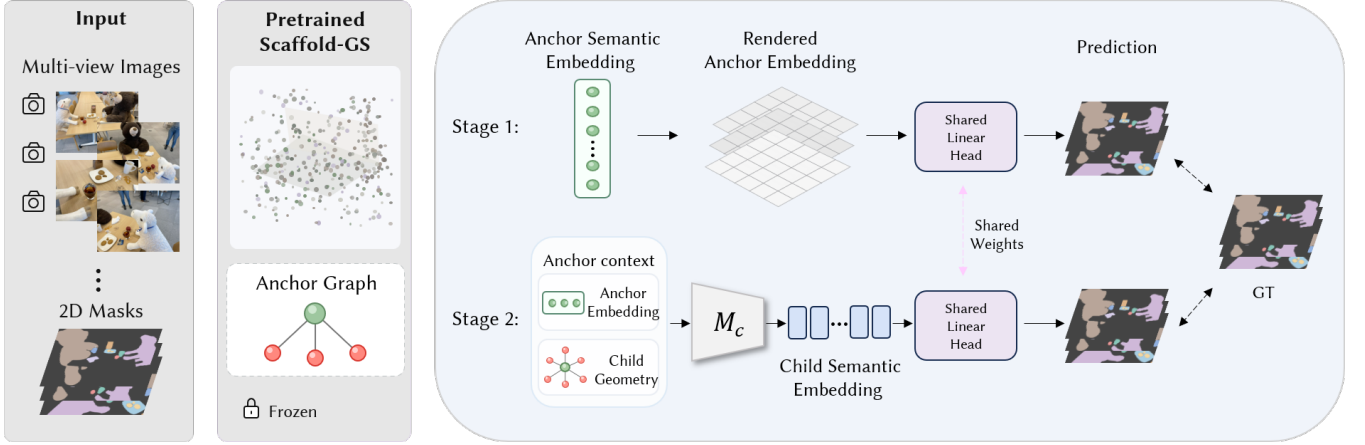


Figure 2: Overview of our ChildSplat framework. Starting from a pretrained and frozen Scaffold-GS representation, Stage 1 learns anchor-level semantic embeddings from 2D mask supervision. Stage 2 refines child semantics from anchor context and child geometry, producing fine-grained child semantic embeddings. Both anchor and child predictions are classified by a shared linear head, aligning rendered 2D predictions and 3D child-level semantics in a common semantic space.

children of an anchor is too coarse for fine-grained segmentation. Children spawned from the same anchor may contribute to different object boundaries or part regions. We therefore refine child semantics with a geometry-aware residual module.

For the k -th child of anchor i , we first define its scaled offset and world-space center as

$$\tilde{\mathbf{o}}_{i,k} = \mathbf{l}_i \odot \mathbf{o}_{i,k}, \quad \mu_{i,k} = \mathbf{x}_i + \tilde{\mathbf{o}}_{i,k}, \quad \bar{\mu}_{i,k} = \gamma(\mu_{i,k}), \quad (5)$$

where $\mathbf{l}_i$ is the anchor scaling vector, $\mathbf{o}_{i,k}$ is the pretrained child offset, and $\gamma(\cdot)$ denotes the same fixed coordinate normalization used for anchor positions. These geometry terms are inherited from the frozen Scaffold-GS representation and are used only as fixed geometric conditions for semantic refinement.

The child semantic residual is conditioned on both the anchor semantic context and the local child geometry:

$$\Delta \mathbf{e}_{i,k} = M_c([\mathbf{u}_i, \mathbf{e}_i, \mathbf{o}_{i,k}, \tilde{\mathbf{o}}_{i,k}, \bar{\mu}_{i,k}, r_k]), \quad (6)$$

where M_c denotes the geometry-aware child refinement MLP and r_k denotes the normalized child slot index. The final child semantic embedding is obtained by residual refinement:

$$\mathbf{e}_{i,k} = \mathbf{e}_i + \beta_r \Delta \mathbf{e}_{i,k}, \quad (7)$$

where $\beta_r \in [0, 1]$ controls the strength of child-level residuals. It is set to zero during anchor-level warm-up and is gradually increased after child refinement starts. This residual formulation regularizes child semantics around the parent anchor semantics while allowing local semantic deviations near object boundaries and fine-grained parts.

The child-level logits are computed by the shared classifier:

$$\mathbf{z}_{i,k} = \mathcal{C}(\mathbf{e}_{i,k}) = \mathbf{W}\mathbf{e}_{i,k} + \mathbf{b}. \quad (8)$$

4.3. Semantic Splatting and Two-stage Optimization

Differentiable semantic splatting. We render semantic embeddings by reusing the differentiable splatting weights of the frozen Scaffold-GS renderer [KKLD23, LYX*24]. The semantic embedding of each child Gaussian is treated as an additional per-Gaussian attribute, analogous to color but defined in a learned semantic space. For a pixel $\mathbf{p}$, we first compute the accumulated opacity

$$A(\mathbf{p}) = \sum_{(i,k) \in \mathcal{N}(\mathbf{p})} w_{i,k}(\mathbf{p}), \quad w_{i,k}(\mathbf{p}) = T_{i,k}(\mathbf{p})\alpha_{i,k}(\mathbf{p}), \quad (9)$$

where $\mathcal{N}(\mathbf{p})$ denotes the child Gaussians contributing to pixel $\mathbf{p}$, $\alpha_{i,k}$ is the opacity contribution, and $T_{i,k}$ is the accumulated transmittance. Both $\alpha_{i,k}$ and $T_{i,k}$ are inherited from the frozen Scaffold-GS renderer. The rendered semantic embedding is obtained by alpha-normalized compositing:

$$\mathbf{E}(\mathbf{p}) = \frac{\sum_{(i,k) \in \mathcal{N}(\mathbf{p})} w_{i,k}(\mathbf{p}) \mathbf{e}_{i,k}}{A(\mathbf{p}) + \epsilon}. \quad (10)$$

Therefore, semantic learning does not alter the underlying geometry or appearance field, but only learns semantic attributes attached to the existing neural Gaussian scaffold.

The rendered embedding map is classified by the same linear head:

$$\mathbf{Z}(\mathbf{p}) = \mathcal{C}(\mathbf{E}(\mathbf{p})) = \mathbf{W}\mathbf{E}(\mathbf{p}) + \mathbf{b}. \quad (11)$$

Given a 2D mask label map Y , we supervise the rendered logits with an opacity-masked cross-entropy loss:

$$\mathcal{L}_{2D} = \frac{1}{\sum_{\mathbf{p}} m(\mathbf{p}) + \epsilon} \sum_{\mathbf{p}} m(\mathbf{p}) \text{CE}(\mathbf{Z}(\mathbf{p}), Y(\mathbf{p})), \quad (12)$$

where $m(\mathbf{p}) \in [0, 1]$ is an opacity-valid weight derived from the accumulated opacity $A(\mathbf{p})$. This avoids enforcing semantic labels on pixels with unreliable or negligible Gaussian support.

We classify rendered embeddings rather than directly rendering

Table 1: Two-stage semantic optimization.

Stage	Trainable modules	Frozen modules	Supervision
Anchor warm-up	$\mathbf{f}^a, M_a, \mathcal{C}$	Scaffold-GS, M_c	opacity-masked 2D CE
Child refinement	$M_c, \mathcal{C}$	Scaffold-GS, $\mathbf{f}^a, M_a$	2D CE + child-level losses

class probabilities. This keeps anchor embeddings, child embeddings, and rendered embeddings in the same semantic space, allowing a single classifier to supervise both 3D child predictions and 2D rendered predictions.

Two-stage semantic optimization. Directly optimizing child-level semantic residuals from the beginning can be unstable, because early anchor embeddings have not yet formed a coherent semantic space. We therefore adopt a two-stage optimization strategy that first establishes a stable anchor-level semantic scaffold and then learns fine-grained child deviations.

In the first stage, child refinement is disabled and each child inherits the embedding of its parent anchor, i.e., $\mathbf{e}_{i,k} = \mathbf{e}_i$. We optimize the anchor semantic feature $\mathbf{f}_i^a$, the anchor semantic MLP M_a , and the shared classifier $\mathcal{C}$ using opacity-masked rendered 2D mask supervision. This stage learns a compact semantic scaffold aligned with the frozen Scaffold-GS geometry.

In the second stage, the pretrained Scaffold-GS representation and the anchor semantic scaffold are kept fixed. Specifically, we freeze the anchor semantic features $\mathbf{f}^a$ and the anchor semantic MLP M_a , and use the resulting anchor embeddings as fixed semantic context for child refinement. We optimize the child refinement module M_c together with the shared classifier $\mathcal{C}$. The rendered 2D loss is still applied to maintain view-consistent predictions. However, rendered supervision alone does not explicitly assign semantic ownership to individual Scaffold children. We therefore introduce child-level ownership losses in the second stage, derived from projected and multi-view child-mask associations, as detailed in Sec. 4.4. The main semantic objective is

$$\mathcal{L}_{\text{main}} = \lambda_{2D} \mathcal{L}_{2D} + \lambda_{\text{child}} \mathcal{L}_{\text{child}}. \quad (13)$$

This separation prevents noisy child-level optimization from corrupting the anchor semantic space and encourages the residual branch to focus on local semantic corrections near boundaries and parts. The trainable modules, frozen components, and supervision used in the two stages are summarized in Table 1.

4.4. Multi-view Child Ownership and Hierarchical Supervision

During child refinement, we jointly apply three complementary objectives: per-view projected supervision $\mathcal{L}_{\text{proj}}$, cross-view ownership aggregation $\mathcal{L}_{\text{mv}}$, and embedding-space regularization $\mathcal{L}_{\text{ctr}}$.

Projected child supervision. We first supervise visible children using labels derived from their projected valid pixels. For a training view v , let M_c^v be the mask of class c . Let $P_{i,k,c} = \text{softmax}(\mathcal{C}(\mathbf{e}_{i,k}))_c$ be the predicted probability of child (i,k) for class c . We accumulate the contribution of child (i,k) inside each mask as

$$s_{i,k,c}^v = \sum_{\mathbf{p} \in M_c^v} w_{i,k}^v(\mathbf{p}), \quad a_{i,k}^v = \sum_{\mathbf{p}} w_{i,k}^v(\mathbf{p}), \quad (14)$$

where pixels marked as background or invalid are excluded from the sums. To handle overlapping masks and children with different opacity mass, we use the normalized projected support

$$q_{i,k,c}^v = \frac{s_{i,k,c}^v}{a_{i,k}^v + \epsilon}, \quad \hat{y}_{i,k,v}^{\text{proj}} = \arg \max_c q_{i,k,c}^v. \quad (15)$$

A projected child-view tuple is considered reliable only if the child has sufficient visible contribution, $a_{i,k}^v > \tau_{\text{op}}$, and $\max_c q_{i,k,c}^v > \tau_{\text{proj}}$. If several masks overlap, the label with the largest normalized support is selected. Projected labels are applied per view rather than first aggregated across views, so the same child can receive multiple projected supervision terms when it is visible in multiple reliable views. The projected child classification loss is

$$\mathcal{L}_{\text{proj}} = -\frac{1}{|\Omega_{\text{proj}}|} \sum_{(i,k,v) \in \Omega_{\text{proj}}} \log P_{i,k,\hat{y}_{i,k,v}^{\text{proj}}}, \quad (16)$$

where Ω_{proj} denotes reliable projected child-view tuples.

Multi-view ownership aggregation. Because per-view projection does not uniquely determine child ownership when multiple children contribute to a pixel, we further aggregate child-mask associations across training views.

For each child (i,k) and semantic class c , we accumulate its splatting contribution inside and outside the corresponding 2D masks across training views. Using the splatting weights $w_{i,k}^v(\mathbf{p})$ defined in Sec. 4.3, these contributions are

$$A_{i,k,c}^{\text{in}} = \sum_{v \in \mathcal{V}} \sum_{\mathbf{p} \in M_c^v} w_{i,k}^v(\mathbf{p}), \quad A_{i,k,c}^{\text{out}} = \sum_{v \in \mathcal{V}} \sum_{\mathbf{p} \notin M_c^v} w_{i,k}^v(\mathbf{p}), \quad (17)$$

where $\mathcal{V}$ is the set of training views. We define the object contribution ratio as

$$\rho_{i,k,c} = \frac{A_{i,k,c}^{\text{in}}}{A_{i,k,c}^{\text{in}} + A_{i,k,c}^{\text{out}} + \epsilon}. \quad (18)$$

A child is assigned a pseudo label only when it receives reliable support across multiple views. Specifically, a child-class pair is considered reliable if $\rho_{i,k,c} \geq \tau_{\text{mv}}$ and the child is supported by the target mask in at least V_{min} views. If multiple classes satisfy this criterion, we select the class with the largest contribution ratio and use this ratio as the confidence weight. Children without reliable multi-view support are ignored. The confidence-weighted multi-view ownership loss is

$$\mathcal{L}_{\text{mv}} = -\frac{\sum_{(i,k) \in \Omega_{\text{mv}}} s_{i,k} \log P_{i,k,\hat{y}_{i,k}}}{\sum_{(i,k) \in \Omega_{\text{mv}}} s_{i,k} + \epsilon}, \quad (19)$$

where Ω_{mv} denotes the set of reliable children, $\hat{y}_{i,k}$ is the selected pseudo label, and $s_{i,k}$ is its confidence weight. Detailed pseudocode, intermediate quantities, and reliability conditions are provided in the supplementary material.

Child embedding regularization. We also apply a supervised contrastive loss [KTW*20] to regularize the child embedding space. The contrastive mini-batch $\mathcal{B}_{\text{ctr}}$ is sampled from reliable children in Ω_{mv} together with reliable projected children in Ω_{proj} ; unreliable and background-only children are excluded. For a query child $q \in \mathcal{B}_{\text{ctr}}$, positives $\mathcal{P}(q)$ are other sampled children with the

same pseudo label, and candidates $\mathcal{A}(q)$ are all other sampled reliable children. With cosine similarity $\text{sim}(\cdot, \cdot)$ and temperature T , we define

$$\mathcal{L}_{\text{ctr}} = -\frac{1}{|\Omega_{\text{ctr}}|} \sum_{q \in \Omega_{\text{ctr}}} \frac{1}{|\mathcal{P}(q)|} \sum_{p \in \mathcal{P}(q)} \log R(q, p), \quad (20)$$

$$R(q, p) = \frac{\exp(\text{sim}(\mathbf{e}_q, \mathbf{e}_p)/T)}{\sum_{a \in \mathcal{A}(q)} \exp(\text{sim}(\mathbf{e}_q, \mathbf{e}_a)/T)},$$

where Ω_{ctr} contains sampled query children that have at least one positive in the mini-batch. Negatives are children in $\mathcal{A}(q)$ with different pseudo labels. The child-level objective is

$$\mathcal{L}_{\text{child}} = \lambda_{\text{proj}} \mathcal{L}_{\text{proj}} + \lambda_{\text{mv}} \mathcal{L}_{\text{mv}} + \lambda_{\text{ctr}} \mathcal{L}_{\text{ctr}}. \quad (21)$$

Here $\mathcal{L}_{\text{proj}}$ provides per-view child supervision, $\mathcal{L}_{\text{mv}}$ enforces view-consistent child ownership, and $\mathcal{L}_{\text{ctr}}$ regularizes child embeddings in the semantic space.

Residualized hierarchical supervision. The semantic child field described above does not require hierarchical annotations. For datasets with only flat masks, we directly train the model in the flat class label space. When hierarchical annotations are available, we incorporate hierarchy-aware supervision without changing the underlying child semantic representation.

Hierarchical masks naturally contain overlapping labels. For example, a child Gaussian assigned to a part region also belongs to its parent object. Directly applying single-label supervision to overlapping parent and part masks would introduce conflicting targets. We therefore convert hierarchical annotations into a residualized leaf label space. Let a be a parent class and $\text{Ch}(a)$ be its annotated subclasses or part classes. Given a parent mask $\mathcal{M}_a$ and subclass masks $\{\mathcal{M}_j\}_{j \in \text{Ch}(a)}$, we define a residual parent leaf:

$$\mathcal{M}_{a^{\text{self}}} = \mathcal{M}_a \setminus \bigcup_{j \in \text{Ch}(a)} \mathcal{M}_j. \quad (22)$$

Each subclass mask is clipped by its parent region, and the resulting leaf masks are made mutually exclusive. The primary classifier $\mathcal{C}$ is trained in this non-overlapping leaf label space. Thus, the child pseudo labels used in child-level supervision are residual leaf labels. Parent-level predictions can be recovered by aggregating descendant leaf probabilities:

$$P_a^{\text{desc}} = \sum_{\ell \in \text{Desc}(a)} P_\ell, \quad (23)$$

where a is a parent class, $\text{Desc}(a)$ denotes its descendant leaf labels, and P_ℓ is the predicted probability of leaf label ℓ for a child or a rendered pixel.

To explicitly encode the hierarchy, we introduce hierarchy-aware constraints on supervised child predictions. For a hierarchy-supervised child $r = (i, k) \in \Omega_h$, let y_r and ω_r be its residual leaf label and pseudo-label confidence, and let A_r contain its annotated ancestors with at least two residual descendants. For $a \in A_r$, define $P_{r,a}^{\text{desc}} = \sum_{\ell \in \text{Desc}(a)} P_{r,\ell}$. The ancestor consistency loss $\mathcal{L}_{\text{anc}}$ encourages a child assigned to a leaf label to place probability mass within

the descendant leaf set of its ancestor.

$$\mathcal{L}_{\text{anc}} = -\frac{\sum_{r \in \Omega_h} \omega_r \sum_{a \in A_r} \log(P_{r,a}^{\text{desc}} + \epsilon)}{\sum_{r \in \Omega_h} \omega_r |A_r|}. \quad (24)$$

The sibling discrimination loss $\mathcal{L}_{\text{sib}}$ separates the target leaf within each ancestor's descendant set:

$$\mathcal{L}_{\text{sib}} = -\frac{\sum_{r \in \Omega_h} \omega_r \sum_{a \in A_r} \log \frac{P_{r,y_r}}{P_{r,a}^{\text{desc}}}}{\sum_{r \in \Omega_h} \omega_r |A_r|}. \quad (25)$$

In addition, we attach a lightweight ancestor multi-label head to the child semantic embedding. This head predicts parent-class memberships with sigmoid outputs and is supervised by the ancestors of the child leaf label. Let H denote this head and $\mathbf{t}_r^{\text{anc}}$ the multi-hot ancestor-membership target over the parent-class vocabulary. Its confidence-weighted loss is

$$\mathcal{L}_{\text{hml}} = \frac{\sum_{r \in \Omega_h} \omega_r \text{BCE}(\sigma(H(\mathbf{e}_r)), \mathbf{t}_r^{\text{anc}})}{\sum_{r \in \Omega_h} \omega_r}. \quad (26)$$

The final objective with hierarchical supervision is

$$\mathcal{L} = \mathcal{L}_{\text{main}} + \lambda_{\text{anc}} \mathcal{L}_{\text{anc}} + \lambda_{\text{sib}} \mathcal{L}_{\text{sib}} + \lambda_{\text{hml}} \mathcal{L}_{\text{hml}}. \quad (27)$$

For datasets without hierarchical annotations, the residualized leaf conversion, hierarchy-related losses, and ancestor multi-label head are omitted. Thus, hierarchy-aware learning is a lightweight extension of the proposed semantic Scaffold-GS pipeline rather than a separate model.

5. Experiments

5.1. Settings

Datasets and metrics. We evaluate our method on open-world 3D object segmentation and hierarchical object-part segmentation. For LERF, we use two object-mask settings, both evaluated in a query-specific manner with the Gaussian Grouping-style evaluation script and a fixed binarization threshold, and report mean IoU (mIoU) and Boundary IoU (BIOU); mBIOU denotes the mean BIOU across evaluated queries.

LERF-Mask is the public benchmark introduced by Gaussian Grouping [YDYK24], which provides held-out novel-view object masks for query-specific evaluation. It is evaluated strictly under the public Gaussian Grouping protocol: we use the public test-view object masks and the Gaussian Grouping evaluation script without modifying test annotations. The small number of evaluation views follows the public LERF-Mask protocol of Gaussian Grouping, where each scene contains 2–4 annotated novel test views.

LERF-Obj is our SAM3-assisted auxiliary setting built on three commonly used LERF scenes: *figurines*, *ramen*, and *teatime*. LERF-Obj follows the scene and query organization of LERF-Mask, using the same held-out evaluation views and object-query granularity where applicable. It is evaluated with the same

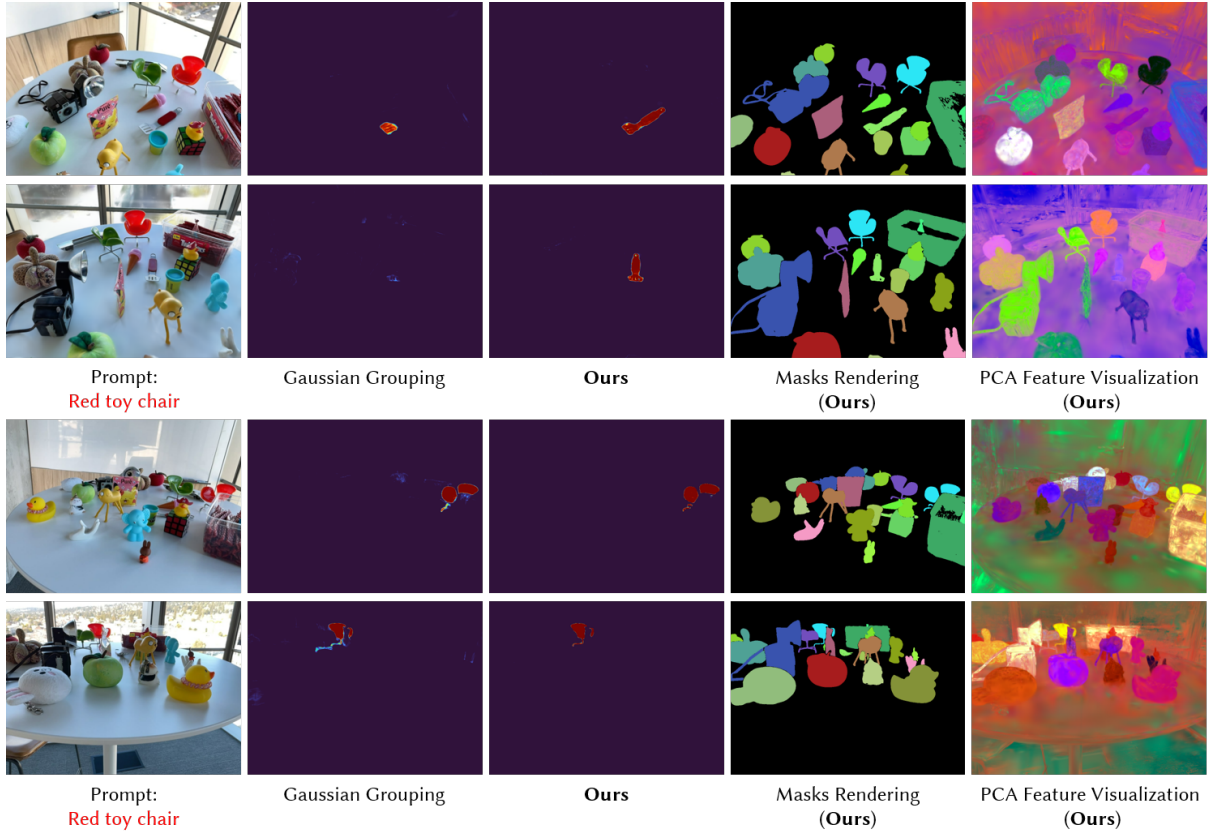


Figure 3: LERF qualitative comparison with Gaussian Grouping. ChildSplat produces localized object responses and coherent rendered semantic features.

Gaussian Grouping-style evaluation script, fixed thresholding rule, mIoU, and boundary IoU metrics as LERF-Mask. The difference is that LERF-Obj uses newly generated SAM3-assisted training-view object masks and object groups, and is therefore intended as an auxiliary analysis of mask-driven 3D grouping rather than a public benchmark setting. For LERF-Obj, we use SAM3 [CGH*25], which supports mask extraction with text prompts, to obtain object masks on selected training views, and associate them across views with a SAM-based tracking pipeline. Object names are used as semantic prompts, and only training-view masks are used for semantic learning. Held-out evaluation masks are never used for mask extraction, association, optimization, threshold tuning, or model selection; we will release the LERF-Obj prompts, selected training views, masks, and association metadata.

3D-OVS benchmark. We evaluate on the standard five scenes: *bed*, *bench*, *room*, *sofa*, and *lawn* [LZZ*23]. We follow the standard scene split and class list, and report per-scene mIoU and the overall score. Official test-view annotations are used only for evaluation and are never used for training, prompt selection, mask filtering, threshold tuning, pseudo-label generation, or model selection. For our method, 2D supervision is generated only on training views. We use the official class names as text prompts to SAM3 [CGH*25] to obtain class-specific seed masks, without point or box prompts.

Each seed mask is propagated to other training views with SAM mask prompts, producing multi-view masks for the same text-defined object class. Empty or invalid propagated masks are treated as unlabeled pixels rather than being manually corrected. The resulting training masks are fixed before semantic optimization and are used for both rendered 2D supervision and child-level pseudo-label generation.

3D-OVS comparison protocol. Following prior open-vocabulary 3D segmentation benchmarks, 3D-OVS comparisons are made under the same official test annotations and mIoU metric. The training masks, 2D priors, prompt strategies, and semantic supervision are method-specific components of each complete pipeline. Therefore, literature-reported baseline results should be interpreted as standard-test comparisons of complete systems rather than same-mask controlled comparisons. When reproduced results are unavailable, we use the numbers reported in the corresponding papers.

Implementation details. All experiments are conducted on NVIDIA RTX 3090 GPUs. For each scene, we first train or load a Scaffold-GS model for RGB reconstruction and freeze its geometry and appearance parameters. We then optimize only the semantic branch for 30k iterations. The semantic embedding dimension is 16, and each anchor spawns 10 child Gaussians following the

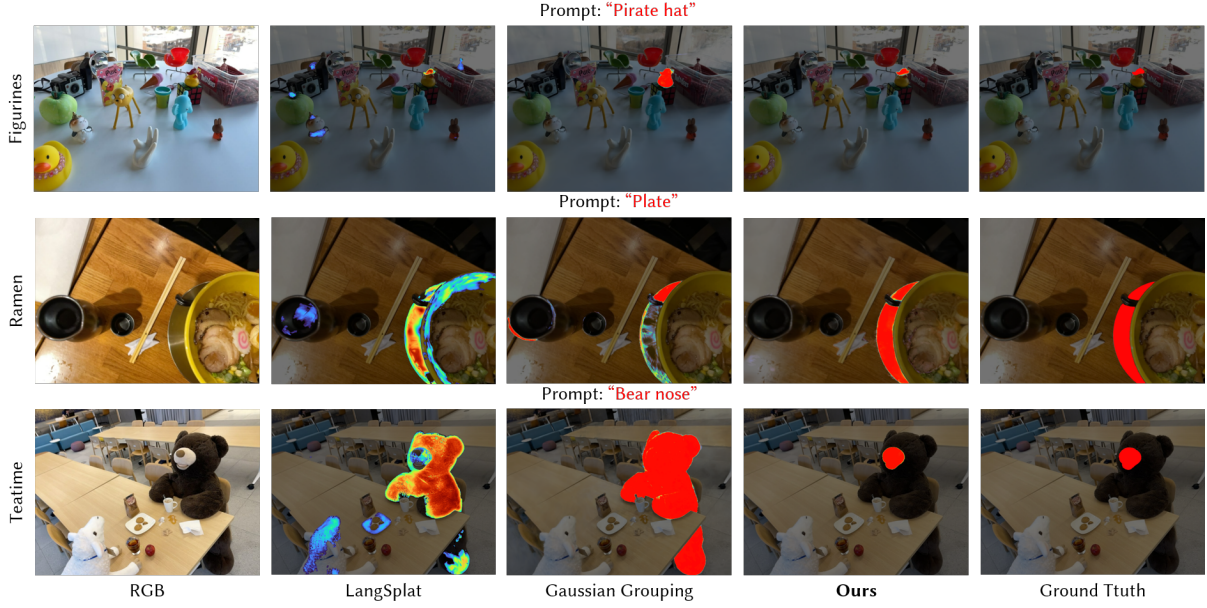


Figure 4: Query-specific object mask rendering on LERF.

Table 2: Statistics of LERF-Obj.

Scene	Masked Train Views	Object Groups	Eval Views
<i>figurines</i>	292	22	4
<i>ramen</i>	131	11	3
<i>teatime</i>	177	10	2

pretrained Scaffold-GS configuration. The child refinement MLP takes child offsets, scaled offsets, world positions, and slot indices as input, with a hidden dimension of 64. All variants use the same linear classifier on top of the rendered 16-D semantic embedding. For LERF mask evaluation, rendered object score maps are saved as grayscale prediction maps, and binary masks are obtained by the Gaussian Grouping evaluator using a fixed intensity threshold of 128. We report mIoU and mBIoU [CGD*21] computed by this script. Unless otherwise specified, all experiments use the same implementation constants. The anchor warm-up lasts 3k iterations, after which β_t is ramped from 0 to 1 during child refinement. For multi-view ownership, we set the ratio threshold to $\tau_{mv} = 0.6$ and the minimum supporting views to $V_{min} = 2$. For projected child labels, we use the visible contribution threshold $\tau_{op} = 10^{-4}$ and the projected support threshold $\tau_{proj} = 0.5$. The rendered 2D loss has weight $\lambda_{2D} = 1.0$, the global child loss weight is $\lambda_{child} = 1.0$, the child-level loss weights are $(\lambda_{proj}, \lambda_{mv}, \lambda_{ctr}) = (0.2, 0.2, 0.1)$, and the contrastive temperature is $T = 0.03$. All scenes use the same $(\lambda_{mv}, \tau_{mv}, V_{min}, T) = (0.2, 0.6, 2, 0.03)$ configuration without scene-specific tuning; the corresponding sensitivity results are reported in the supplementary material. These constants are fixed across scenes and compared variants unless explicitly ablated.

5.2. Results on LERF

We report results on both the public LERF-Mask benchmark and our SAM3-assisted LERF-Obj setting described above.

Quantitative Results. Table 3 reports the LERF object-mask evaluation. On the public LERF-Mask benchmark, ChildSplat achieves competitive performance among the compared methods, obtaining 80.2 mIoU and 76.4 mBIoU under the Gaussian Grouping evaluation script. We also report LERF-Obj as a SAM3-assisted auxiliary analysis of mask-driven 3D grouping under denser training-view object masks. Although LERF-Obj uses the same Gaussian Grouping-style held-out-view evaluator, its training masks and object groups are newly constructed, so it is not intended as a direct public-benchmark comparison with prior methods.

Visualization Results. Figures 3 and 4 visualize rendered semantic features and query-specific masks on LERF. ChildSplat produces spatially coherent feature structures, compact query activations, and masks that align well with object boundaries, indicating view-consistent 3D semantic learning from 2D mask supervision.

Editing Handles. For the local editing examples in Figure 1, we use the learned 3D semantic groups as object-selection handles and follow the Gaussian Grouping editing protocol [YDYK24]: selected regions are rendered to training views, inpainted with LaMa [SLM*21], and lifted back to the 3D Gaussian representation. This editing pipeline is used only to demonstrate the utility of the learned semantic groups and is not a separate contribution of our method.

Efficiency, storage, and training cost. We further compare runtime, memory usage, and semantic storage on the LERF-Obj *figurines* scene using NVIDIA RTX 3090 GPUs. Across the three

Table 3: LERF object-mask evaluation. LERF-Mask follows the public Gaussian Grouping benchmark protocol. LERF-Obj is a SAM3-assisted diagnostic setting that uses new training masks while retaining the LERF-Mask held-out views and evaluation protocol and is reported only as an auxiliary analysis. Best public-benchmark results on LERF-Mask are bolded. *: reproduced result.

Method	Figurines		Ramen		Teatime		Average	
	mIoU	mBIoU	mIoU	mBIoU	mIoU	mBIoU	mIoU	mBIoU
NeRF-based								
LERF [KKG*23]	33.5	30.6	28.3	14.7	49.7	42.6	37.2	29.3
3D-OVS [LZZ*23]	44.8	—	28.7	—	56.1	—	43.2	—
Laser [MDB*25]	63.5	—	44.6	—	62.4	—	56.8	—
OmniSeg3D-NeRF* [YYZ*23]	40.2	27.6	29.2	17.6	42.1	40.2	37.2	28.5
Open3DRF-NeRF [LYB*25]	—	—	—	—	—	—	46.4	45.4
3DGS-based								
OmniSeg3D-GS* [YYZ*23]	36.4	27.2	28.6	14.7	44.9	43.1	36.6	28.3
OpenGaussian [WML*24]	39.3	—	31.0	—	60.4	—	43.6	—
Open3DRF-GS [LYB*25]	—	—	—	—	—	—	44.4	44.6
SuperGSeg [LWL*26]	43.7	60.7	18.1	23.9	55.3	78.0	39.0	54.2
LangSplat [QLZ*24]	44.7	41.9	51.2	48.8	65.1	60.8	53.7	50.5
Gaussian Grouping [YDYK24]	69.7	67.9	77.0	68.7	71.7	66.1	72.8	67.6
Hi-LSplat [ZZWW26]	71.5	69.3	78.8	72.6	73.7	67.3	74.7	69.7
ILGS [JK25]	75.9	73.8	84.3	75.5	81.2	78.8	80.5	76.0
Unified-Lift [ZQL*25]	—	—	—	—	—	—	80.9	77.1
Gaga [LLK*24]	92.3	90.8	72.0	63.3	71.2	68.4	78.5	74.2
Ours, LERF-Mask	79.0	78.5	84.4	75.6	77.2	75.1	80.2	76.4
Ours, LERF-Obj	94.3	93.2	89.7	82.6	95.9	95.5	93.3	90.4

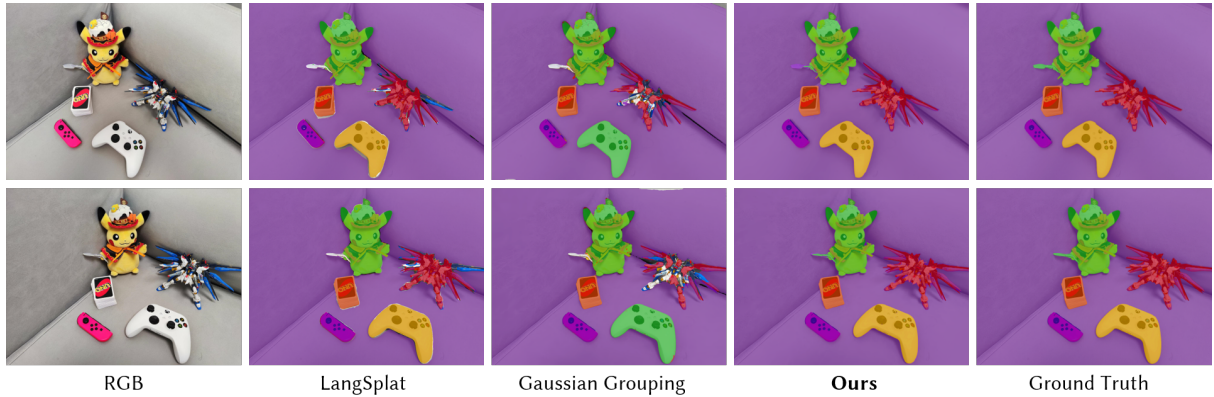


Figure 5: Multi-view semantic segmentation on 3D-OVS.

scenes, Scaffold-GS reconstruction takes 19.4–20.5 minutes, 30k-iteration semantic optimization takes 41.2–64.8 minutes, and peak training memory is 18.3–19.6 GiB on the same GPU. These times exclude external SAM3 mask generation and propagation; complete per-scene measurements are provided in the supplementary material. Table 5 separates full semantic inference from cached interactive querying. The initial full semantic inference remains slower than the compared baselines and is a limitation, although ChildSplat uses lower inference memory and more compact semantic storage. After caching predicted child labels once per scene without using ground-truth labels, query-specific masks render at 5.17 ms per view. The cached-query mode stores additional predicted child labels, increasing semantic storage from 26.26 MB to

48.87 MB, while still remaining more compact than LangSplat and Gaussian Grouping in Table 5.

5.3. Results on 3D-OVS

Quantitative Results. Table 4 compares ChildSplat with prior NeRF-based and 3DGS-based open-vocabulary 3D segmentation systems on 3D-OVS [LZZ*23]. All methods are compared on the standard official test annotations with the mIoU metric. Following prior work, baseline numbers are taken from the corresponding papers when reproduced results are unavailable. Because training masks, 2D priors, prompt strategies, and semantic supervision are method-specific, the table should be interpreted as a standard-test comparison of complete systems rather than a same-mask controlled comparison.

Table 4: Semantic segmentation results on the standard 3D-OVS benchmark [LZZ*23]. All methods are evaluated on the official test annotations using the mIoU metric. Baseline numbers are taken from their original papers when reproduced results are unavailable. The table compares complete open-vocabulary 3D segmentation systems under the standard 3D-OVS evaluation protocol. *: reproduced result.

Method	Semantic Segmentation (mIoU)					
	Bed	Bench	Room	Sofa	Lawn	Overall
NeRF-based						
LERF [KKG*23]	73.5	53.2	46.6	27.0	73.7	54.8
Open3DRF-NeRF [LYB*25]	—	—	—	—	—	77.5
3D-OVS [LZZ*23]	89.5	89.3	92.8	74.0	88.2	86.8
OmniSeg3D-NeRF* [YYZ*23]	91.6	92.1	91.4	81.2	89.6	89.2
Laser [MDB*25]	91.4	88.3	85.9	86.0	88.5	88.1
3DGS-based						
LEGaussians [SWDG24]	56.8	28.9	57.2	52.6	44.1	47.9
OmniSeg3D-GS* [YYZ*23]	91.9	90.1	91.4	78.6	90.3	88.5
Open3DRF [LYB*25]	—	—	—	—	—	77.5
FMGS [ZSZ*25]	80.6	84.5	87.9	90.8	92.6	87.3
Gaussian Grouping [YDYK24]	97.3	73.7	79.0	68.1	96.5	82.9
CGC [SDTDB24]	95.2	96.1	86.8	67.5	91.8	87.5
FastLGS [JZT*25]	94.7	95.1	95.3	90.6	93.9	93.9
LangSplat [QLZ*24]	92.5	94.2	94.1	90.0	96.1	93.4
Hi-L-Splat [ZZWW26]	95.9	97.3	96.7	92.8	97.6	96.1
ILGS [JK25]	96.4	95.5	93.6	92.2	94.2	94.4
Ours	97.6	96.8	97.2	95.3	97.7	96.9

Table 5: Efficiency comparison on Figurines. Full rows use full semantic inference; the cached-query row uses predicted child-label caching after training.

Method / Mode	Time	Inf. Mem.	Sem. Storage	Cache
LangSplat full	19.66 ms	3.71 GB	63.60 MB	No
Gaussian Grouping full	10.87 ms	4.21 GB	139.97 MB	No
ChildSplat full	35.28 ms	1.58 GB	26.26 MB	No
ChildSplat cached query	5.17 ms	2.87 GB	48.87 MB	Yes

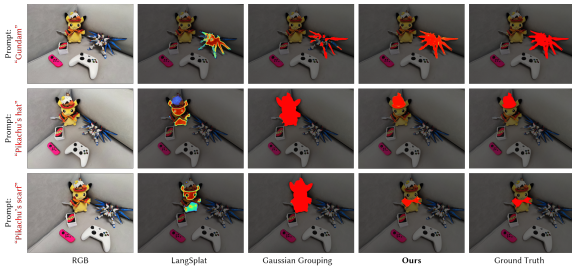


Figure 6: Query-specific segmentation results on 3D-OVS.

5.4. Ablation Study

We conduct controlled ablations on the LERF-Obj *figurines* scene to analyze where the gain of ChildSplat comes from. All variants use the same frozen Scaffold-GS reconstruction, training masks, query objects, and held-out evaluation views. These controlled variants are intended to isolate the effect of the proposed anchor-child semantic design under identical 2D supervision. We focus on

Table 6: Ablation of child-level semantic ownership on LERF-Obj *figurines*.

Variant	2D mIoU ↑	Pseudo-label Agreement ↑	Weighted Pseudo-label Agreement ↑	Child-render mIoU ↑
Anchor-only	94.4	57.7	57.8	62.4
Child residual, 2D only	95.4	65.0	65.2	68.0
Full (local cues only)	95.1	71.5	71.7	72.0
Full	93.7	89.1	89.1	72.6

whether the model learns correct semantic ownership at the child-Gaussian level, since rendered 2D mask quality alone can hide errors in the underlying 3D semantic assignment.

Child-level semantic ownership. Table 6 compares four variants. *Anchor-only* disables child refinement, so all children inherit the parent anchor semantics. *Child residual, 2D only* enables the child residual branch but trains it only through rendered 2D supervision. *Full (local cues only)* adds the multi-view child-level losses but removes the normalized child world position and slot index from the child refinement input. *Full* uses the complete geometry-aware child refinement and child-level supervision.

In addition to rendered 2D mIoU, we report direct child-level metrics. *Pseudo-label Agreement* and its confidence-weighted variant measure consistency with training-derived multi-view child pseudo labels rather than accuracy against independent 3D ground truth. We additionally report *Child-render mIoU*: for each query, children predicted with confidence above 0.95 are rendered alone on held-out views, their accumulated alpha is thresholded at 0.5, and the resulting masks are compared with held-out object masks. This metric evaluates whether the predicted child assignments recover object support without relying on the shared rendered-embedding classifier.

The results show that rendered 2D mIoU alone does not fully reflect 3D semantic quality. The 2D-only child residual variant achieves the highest rendered 2D mIoU (95.4), but its two agreement scores (65.0/65.2) and Child-render mIoU (68.0) remain substantially lower. The full model reduces rendered 2D mIoU to 93.7 while increasing these metrics to 89.1/89.1 and 72.6, respectively. This trade-off arises because alpha compositing can reproduce accurate 2D masks even when overlapping children have incorrect semantic assignments. The projected and multi-view losses instead favor reliable child ownership, which is important for child selection, editing, and view-consistent object queries.

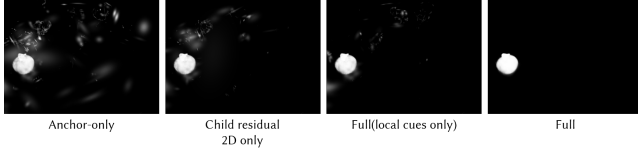
Effect of two-stage optimization. We further ablate when child refinement and child-level losses are enabled. As shown in Table 7, activating child refinement from the beginning is less stable because the anchor semantic scaffold has not yet formed a coherent semantic space. A short warm-up improves the result, while an overly long warm-up delays local correction and reduces final mask quality. Starting child refinement after a 3k-iteration anchor warm-up gives the best result among these settings, supporting the proposed two-stage optimization.

5.5. Hierarchical Query Extension

We further evaluate whether ChildSplat can support object-part queries within the same 3D representation. On the 3D-OVS *sofa*

Table 7: Ablation of the child-refinement start iteration.

Training strategy	Child start iter.	mIoU $\uparrow$
No anchor warm-up	0	87.94
Short anchor warm-up	200	92.30
Ours two-stage	3000	93.50
Long anchor warm-up	10000	90.63

**Figure 7:** Qualitative ablation of selected semantic children on LERF-Obj.

scene, we construct SAM3-assisted annotations for parent objects and parts, such as *Pikachu*, *Pikachu's eyes*, *Pikachu's fork*, and *Xbox controller buttons*. Since parent and part masks overlap, we residualize them into non-overlapping leaf regions for semantic optimization and add hierarchy-aware supervision for parent-part membership. For evaluation, we split the 28 sofa views into 23 training views and five annotated held-out views; the held-out views are excluded from mask propagation, pseudo-label generation, semantic optimization, threshold tuning, and model selection. For the hierarchy-aware experiments, we set $\lambda_{\text{anc}} = \lambda_{\text{sib}} = \lambda_{\text{hml}} = 0.1$. We report fixed 30k-iteration checkpoints and evaluate parent mIoU, part mIoU, part boundary IoU, and hierarchical consistency. This analysis is intended as a controlled validation of the hierarchy-aware extension rather than a full hierarchical segmentation benchmark.

Table 8 shows that full hierarchy improves parent and part mIoU over residualized leaves, while residual-only training yields higher part boundary IoU. This suggests that the current hierarchy losses mainly improve parent-part organization rather than boundary sharpness.

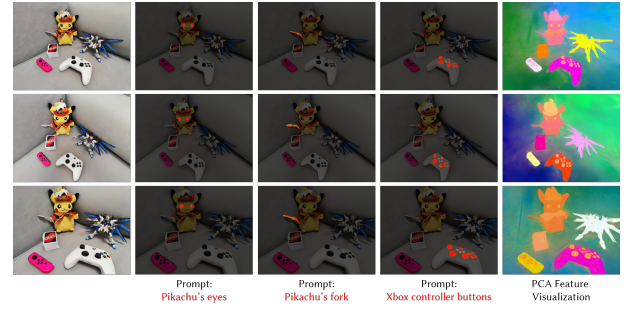
Figure 8 shows that ChildSplat can render fine-grained part queries, such as eyes, accessories, and buttons, from the same semantic field used for object-level segmentation. The PCA visualization further indicates coherent object-part structures in the learned embedding space.

6. Conclusion

We presented ChildSplat, a mask-supervised semantic extension of Scaffold-GS that combines compact anchor-level context with child-specific semantic refinement and ownership supervision. Experiments on LERF and 3D-OVS demonstrate effective and compact 3D semantic grouping, while controlled ablations validate the benefit of child-level modeling. The same representation also supports preliminary hierarchical object-part queries when hierarchical annotations are available. Current limitations include dependence on reconstruction and 2D mask quality, semantic inference overhead, and limited evaluation of hierarchy-aware segmentation.

Table 8: Quantitative analysis of hierarchical queries on the 3D-OVS sofa scene. Parent mIoU uses descendant-aggregated parent predictions; Part mIoU and Part BIoU evaluate fine-grained part queries; HCons measures the fraction of predicted part pixels contained in the corresponding parent prediction. Higher is better.

Variant	Parent mIoU $\uparrow$	Part mIoU $\uparrow$	Part BIoU $\uparrow$	HCons $\uparrow$
Residual leaves only	93.68	84.38	63.81	100.00
Full hierarchy	95.18	86.05	58.13	99.98

**Figure 8:** Part-level query rendering on the 3D-OVS sofa scene.

7. Acknowledgments

The authors thank all anonymous reviewers for their constructive feedback. This work was supported in part by the National Key R&D Program of China (No. 2022YFB3303400).

References

- [CCZ*24] CHEN Y., CHEN Z., ZHANG C., WANG F., YANG X., WANG Y., CAI Z., YANG L., LIU H., LIN G.: Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2024), pp. 21476–21485. 3
- [CFY*25] CEN J., FANG J., YANG C., XIE L., ZHANG X., SHEN W., TIAN Q.: Segment any 3d gaussians. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2025), vol. 39, pp. 1971–1979. 2
- [CGD*21] CHENG B., GIRSHICK R., DOLLÁR P., BERG A. C., KIRILLOV A.: Boundary iou: Improving object-centric image segmentation evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021). 8
- [CGH*25] CARION N., GUSTAFSON L., HU Y.-T., DEBNATH S., HU R., SURIS D., RYALI C., ALWALA K. V., KHEDR H., HUANG A., LEI J., MA T., GUO B., KALLA A., MARKS M., GREER J., WANG M., SUN P., RÄDLE R., AFOURAS T., MAVROUDI E., XU K., WU T.-H., ZHOU Y., MOMENI L., HAZRA R., DING S., VAZE S., PORCHER F., LI F., LI S., KAMATH A., CHENG H. K., DOLLÁR P., RAVI N., SAENKO K., ZHANG P., FEICHTENHOFER C.: Sam 3: Segment anything with concepts, 2025. 7
- [CSK*25] CHOI S., SONG H., KIM J., KIM T., DO H.: Click-gaussian: Interactive segmentation to any 3d gaussians. In *European Conference on Computer Vision* (2025), Springer, pp. 289–305. 2
- [CTM*21] 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 International Conference on Computer Vision (ICCV)* (2021). 2
- [JK25] JANG S., KIM W.: Identity-aware language gaussian splatting for open-vocabulary 3d semantic segmentation. In *Proceedings of the International Conference on Computer Vision* (2025). 3, 9, 10

- [JKG*23] JATAVALLABHULA K. M., KUWAJERWALA A., GU Q., OMAMA M., IYER G., SARYAZDI S., CHEN T., MAALOUF A., LI S., KEETHA N. V., TEWARI A., TENENBAUM J. B., DE MELO C. M., KRISHNA M., PAULL L., SHKURTI F., TORRALBA A.: Conceptfusion: Open-set multimodal 3d mapping. In *Robotics: Science and Systems XIX* (Daegu, Republic of Korea, July 2023). 2
- [JZT*25] JI Y., ZHU H., TANG J., LIU W., ZHANG Z., TAN X., XIE Y.: Fastlgs: Speeding up language embedded gaussians with feature grid mapping. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2025), vol. 39, pp. 3922–3930. 10
- [KKG*23] KERR J., KIM C. M., GOLDBERG K., KANAZAWA A., TANCIK M.: Lrf: Language embedded radiance fields. In *International Conference on Computer Vision (ICCV)* (2023). 2, 9, 10
- [KKLD23] KERBL B., KOPANAS G., LEIMKÜHLER T., DRETTAKIS G.: 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics* 42, 4 (July 2023). 2, 4
- [KMR*23] KIRILLOV A., MINTUN E., RAVI N., MAO H., ROLLAND C., GUSTAFSON L., XIAO T., WHITEHEAD S., BERG A. C., LO W.-Y., DOLLAR P., GIRSHICK R.: Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (October 2023), pp. 4015–4026. 2
- [KMS22] KOBAYASHI S., MATSUMOTO E., SITZMANN V.: Decomposing nerf for editing via feature field distillation. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)* (2022), pp. 23311–23330. 2
- [KTW*20] KHOSLA P., TETERWAK P., WANG C., SARNA A., TIAN Y., ISOLA P., MASCHINOT A., LIU C., KRISHNAN D.: Supervised contrastive learning. In *Advances in Neural Information Processing Systems (NeurIPS)* (2020), vol. 33, pp. 18661–18673. 5
- [KWK*24] KIM C. M., WU M., KERR J., GOLDBERG K., TANCIK M., KANAZAWA A.: Garfield: Group anything with radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2024), pp. 21530–21539. 3
- [LLK*24] LYU W., LI X., KUNDU A., TSAI Y.-H., YANG M.-H.: Gaga: Group any gaussians via 3d-aware memory bank. *arXiv preprint arXiv:2404.07977* (2024). 3, 9
- [LWL*26] LIANG S., WANG S., LI K., NIEMEYER M., GASPERINI S., LENSCH H. P., NAVAB N., TOMBARI F.: Supergseg: Open-vocabulary 3d segmentation with structured super-gaussians. In *2026 International Conference on 3D Vision (3DV)* (2026), pp. 1650–1659. 9
- [LWM*25] LI H., WU Y., MENG J., GAO Q., ZHANG Z., WANG R., ZHANG J.: Instancegaussian: Appearance-semantic joint gaussian representation for 3d instance-level perception. In *Proceedings of the Computer Vision and Pattern Recognition Conference* (2025). 2
- [LYB*25] LEE H., YUN Y., BAE J., KIM S., UH Y.: Rethinking Open-Vocabulary Segmentation of Radiance Fields in 3D Space. In *AAAI Conference on Artificial Intelligence* (2025), pp. 4491–4498. 9, 10
- [LYX*24] 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* (2024), pp. 20654–20664. 2, 3, 4
- [LZL*26] LI H., ZOU Z., LIU F., ZHANG X., HONG F., CAO Y., LAN Y., ZHANG M., YU G., ZHANG D., LIU Z.: IGGT: Instance-Grounded Geometry Transformer for Semantic 3D Reconstruction. In *International Conference on Learning Representations* (2026). 3
- [LZZ*23] LIU K., ZHAN F., ZHANG J., XU M., YU Y., SADDIK A. E., THEOBALT C., XING E., LU S.: Weakly supervised 3d open-vocabulary segmentation. *arXiv preprint arXiv:2305.14093* (2023). 2, 7, 9, 10
- [MAAD*23] MIRZAEI A., AUMENTADO-ARMSTRONG T., DERPANIS K. G., KELLY J., BRUBAKER M. A., GILITSCHENSKI I., LEVIN-SHTEIN A.: SPIn-NeRF: Multiview segmentation and perceptual inpainting with neural radiance fields. In *CVPR* (2023). 3
- [MDB*25] MIAO X., DUAN H., BAI Y., SHAH T., SONG J., LONG Y., RANJAN R., SHAO L.: Laser: Efficient language-guided segmentation in neural radiance fields. *IEEE Trans. Pattern Anal. Mach. Intell.* 47, 5 (2025), 3922–3934. 9, 10
- [PGJ*23] PENG S., GENOVA K., JIANG C. “., TAGLIASACCHI A., POLLEFEYS M., FUNKHOUSER T.: Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2023), pp. 815–824. 2
- [QLZ*24] 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* (2024), pp. 20051–20060. 2, 3, 9, 10
- [RKH*21] RADFORD A., KIM J. W., HALLACY C., RAMESH A., GOH G., AGARWAL S., SAstry G., ASKELL A., MISHKIN P., CLARK J., KRUEGER G., SUTSKEVER I.: Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning* (18–24 Jul 2021), Meila M., Zhang T. (Eds.), vol. 139 of *Proceedings of Machine Learning Research*, PMLR, pp. 8748–8763. 2
- [SDTDB24] SILVA M. C., DAHAGHIN M., TOSO M., DEL BUE A.: Contrastive gaussian clustering for weakly supervised 3d scene segmentation. In *International Conference on Image Analysis and Processing (ICIAP)* (2024). 10
- [SLM*21] SUVOROV R., LOGACHEVA E., MASHIKHIN A., REMIZOVA A., ASHUKHA A., SILVESTROV A., KONG N., GOKA H., PARK K., LEMPITSKY V.: Resolution-robust large mask inpainting with fourier convolutions. *arXiv preprint arXiv:2109.07161* (2021). 8
- [SWDG24] SHI J., WANG M., DUAN Y., GUAN H.: Language embedded 3d gaussians for open-vocabulary scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2024), IEEE, pp. 5333–5343. 10
- [WML*24] WU Y., MENG J., LI H., WU C., SHI Y., CHENG X., ZHAO C., FENG H., DING E., WANG J., ZHANG J.: Opengaussian: Towards point-level 3d gaussian-based open vocabulary understanding. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)* (2024), pp. 19114–19138. 2, 3, 9
- [YDYK24] YE M., DANELLJAN M., YU F., KE L.: Gaussian grouping: Segment and edit anything in 3d scenes. In *ECCV* (2024). 2, 3, 6, 8, 9, 10
- [YYZ*23] YING H., YIN Y., ZHANG J., WANG F., YU T., HUANG R., FANG L.: Omnise3d: Omniversal 3d segmentation via hierarchical contrastive learning. *arXiv preprint arXiv:2311.11666* (2023). 2, 3, 9, 10
- [ZQL*25] ZHU R., QIU S., LIU Z., HUI K.-H., WU Q., HENG P.-A., FU C.-W.: Rethinking end-to-end 2d to 3d scene segmentation in gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2025), pp. 3656–3665. 3, 9
- [ZSZ*25] ZUO X., SAMANGOEI P., ZHOU Y., DI Y., LI M.: Fmgs: Foundation model embedded 3d gaussian splatting for holistic 3d scene understanding. *International Journal of Computer Vision* 133 (2025), 611–627. 10
- [ZZWW26] ZHAN C., ZHANG Y., WANG G., WANG H.: Hi-Isplat: Hierarchical 3d language gaussian splatting. *IEEE Transactions on Visualization and Computer Graphics* (2026), 1–18. 2, 3, 9, 10