DIGS: Distribution-Informed Gaussian Splatting for Training-Free Open-Vocabulary 3D Segmentation

Zhifan Zeng[1] **Wenbo Xiao**[1] **Arcot Sowmya**[1] **Changming Sun**[2]
[1]UNSW Sydney [2]CSIRO Technology
zhifan.zeng@unsw.edu.au wenbo.xiao@unsw.edu.au
a.sowmya@unsw.edu.au changming.sun@csiro.au

Abstract

Open-vocabulary 3D Gaussian segmentation typically binds semantic features to the per-Gaussian state via per-scene gradient training (taking 1–4 h) or the more recent closed-form distillation. Both commit each primitive to a single feature; therefore, boundary Gaussians spanning multiple 2D instances are forced into a hard decision that degrades retrieval accuracy and yields unstable mask boundaries at render time. We present **DIGS**, a training-free framework with two stages, *class-agnostic mapping* and *multi-modal encoding*, that decouples instance discovery from linguistic naming. At the Gaussian level, our distribution-informed representation maintains an explicit top-M label distribution per primitive, updated by a tempered Dirichlet rule; preservation of multi-peaked label distributions at boundary primitives allows a late-argmax renderer to recover clean instance boundaries. A bidirectional state machine aggregates these distributions into a persistent class-agnostic instance ontology. The instance-level encoding stage then fuses the CLIP visual embedding of each instance's isolated 3D footprint with the CLIP-text embedding of an LLM-generated attribute description, disambiguating visually similar instances. On a single A100, DIGS completes mapping and instance encoding on LERF-OVS in 1.0–3.2 min per scene, depending on whether LLM fusion is enabled, and serves native 3D queries in 5.4 ms. It reaches 71.95% mean mIoU on LERF-OVS, outperforming both the training-free SFS and training-based LaGa baselines. DIGS also achieves the highest mean mIoU on LERF-Mask and 3D-OVS and surpasses Gaga and ObjectGS on Replica under their respective evaluation protocols. Code is available at `https://github.com/ZephyrZos/DIGS`.

1 Introduction

Semantic understanding of a 3D environment from sequential observations is essential for embodied agents [1], virtual reality [2], and language-driven scene editing [3]. Open-vocabulary segmentation on 3D Gaussian Splatting (3DGS) [4] requires a per-Gaussian semantic representation that supports text queries. Prior methods bind a single semantic feature (a continuous CLIP embedding or a clustered instance ID) to each Gaussian via per-scene gradient training [5, 6, 7, 8, 9, 10]. This design couples two failures. First, boundary Gaussians spanning multiple 2D instances are forced to one feature, blurring rendered edges. Second, the per-scene optimisation (taking 1–4 h) bakes the language signal into the per-Gaussian state, locking each reconstructed asset to one language model. Recent work eases one symptom at a time. The closed-form Splat Feature Solver [11] reduces training to minutes but inherits the boundary blur. ObjectGS [12] sharpens boundaries by splitting cross-instance Gaussians under joint photometric and cross-entropy optimisation, which can hurt RGB fidelity when the 2D mask supervision is noisy. The Bayesian alternative [13] treats each query as a binary Beta–Bernoulli problem. No prior design, however, resolves the coupling between training cost and boundary fidelity within a single per-Gaussian representation.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

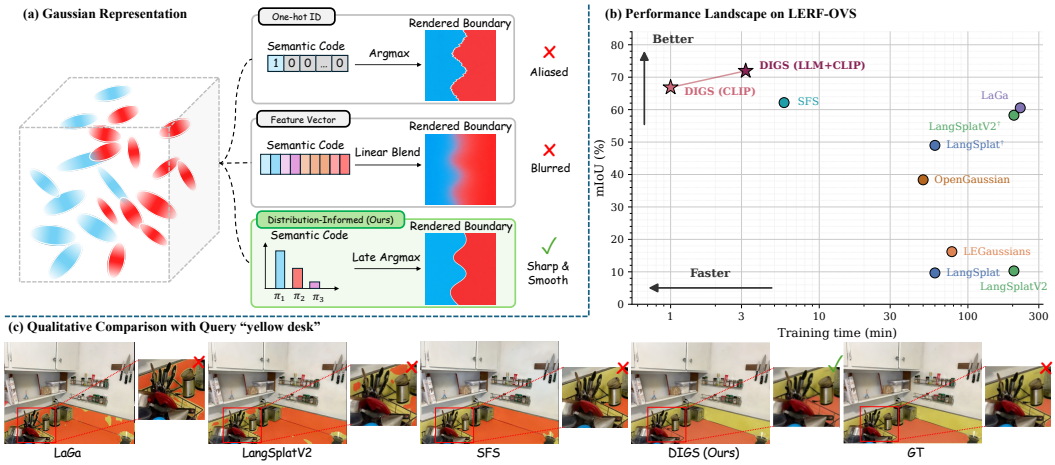


Figure 1: DIGS at a glance. **(a)** Three per-Gaussian semantic representations (One-hot ID, Feature Vector, and distribution-informed per-Gaussian voting (ours)) and their rendered boundaries. **(b)** LERF-OVS performance landscape (with our approach performing the best). **(c)** Qualitative comparison with adversarial query (Waldo_kitchen, *yellow desk*).

The two failures are not independent symptoms but consequences of what each Gaussian stores as the semantic state, as Figure 1a illustrates. A single one-hot label (e.g., ObjectGS [12]) yields sharp argmax-composited boundaries, but the discrete cross-entropy loss forces boundary Gaussians to split during training, degrading the reconstructed RGB whenever the 2D mask supervision is noisy or view-inconsistent. A continuous CLIP-style feature (e.g., LangSplat [5]) tolerates approximate geometry, but linear blending of adjacent-instance embeddings at boundary pixels produces semantically ambiguous mixtures that may match neither class at query time. Both choices commit each primitive to a single semantic mode: the discrete one pays in geometry, the continuous one in semantics. Neither breaks the boundary–cost coupling.

We present DIGS, a training-free framework whose distribution-informed per-Gaussian representation stores an explicit top-M label distribution, updated by a tempered Dirichlet rule. Targeting the boundary failure, the distribution is composited at the pixel level via late-argmax rendering; boundary Gaussians contribute proportionally to adjacent instances and the final label is decided by the majority of unambiguous neighbours, lifting LERF-Mask mean mIoU to 91.71 ($+3.3$ over the previous best ObjectGS at 88.4) and adding $+8.84$ mean mIoU in the LERF-OVS distribution-preserving ablation (§4.3). Targeting the cost-coupling failure, the tempered Dirichlet update is closed-form with a provably bounded per-frame injection (Proposition 1) and executes as a single parallel pass per frame, completing class-agnostic mapping in 0.91 min on a single A100—versus 1–4 h of per-scene gradient training for distillation baselines, a 60–250$\times$ reduction, as Figure 1b visualises.

On LERF-OVS, these choices place DIGS at 71.95 mean mIoU, surpassing the strongest training-free baseline SFS and the strongest training-based baseline LaGa, defining a new accuracy/wall-time frontier. The advantage sharpens on adversarial prompts: as Figure 1c shows on *yellow desk* in Waldo_kitchen, where LaGa, LangSplatV2, and SFS all leak the mask into surrounding fixtures, DIGS returns a clean boundary that covers more of the object than even the human-annotated ground truth. These outcomes follow from three contributions: a distribution-informed per-Gaussian representation (top-M label distribution) that resolves the boundary failure via late-argmax rendering; a closed-form tempered Dirichlet voting rule with a provably bounded per-frame injection that yields a class-agnostic instance ontology in a single parallel pass without per-scene gradient training; and a post-hoc multi-modal encoding that fuses each instance's CLIP visual embedding with the CLIP-text embedding of an LLM-generated attribute description, keeping the mapping output reusable across language backbones.

2 Related Work

3D scene semantic segmentation. Since 3D Gaussian Splatting [4] became a mainstream scene representation, a growing body of work has addressed how to assign semantic or instance information

to its primitives. The dominant recipe uses a 2D class-agnostic mask generator, typically SAM [14], and lifts its outputs onto the Gaussians through per-scene optimisation [9, 15, 16, 17, 18]. This inherits the 2D-to-3D lifting tradition from neural-field predecessors: Semantic-NeRF [19] and Panoptic Lifting [20] embed per-frame labels into implicit fields, while Contrastive Lift [21] and SA3D [22] lift instance or SAM masks; all are bottlenecked by the slow volumetric rendering of their underlying representation. Earlier point-cloud methods [23, 24] train from scratch on annotated 3D data, and open-vocabulary point-cloud systems [25, 26, 27] extend this to CLIP-driven text queries. A complementary line of work casts the lifting problem as a one-shot assignment: FlashSplat [28] solves it using linear programming with an optimal closed-form solution but operates on a single target mask per query; B^3-Seg [13] formulates segmentation as sequential per-query Beta–Bernoulli Bayesian updates, without a persistent multi-instance ontology; Splat Feature Solver [11] casts feature lifting as a closed-form sparse linear inverse problem, achieving CLIP feature fields at minute-scale per-scene but committing each Gaussian to a single continuous feature without per-primitive label uncertainty; and Occam's LGS [29] formulates language-feature assignment as a MAP-based weighted multi-view aggregation, training-free but again binding each Gaussian to a single CLIP feature. Most closely related to our two-stage design, DCSEG [30] and ExtrinSplat [31] also decouple class-agnostic 3D grouping from downstream language assignment via a 2D foundation model, but require per-scene optimisation. DIGS departs from these paradigms by performing multi-class instance lifting with an explicit per-Gaussian top-M distribution in a single streaming pass, without gradient computation.

Per-Gaussian semantic representation. What each Gaussian stores as semantic state is a decisive design axis. A first family distils a continuous vision–language feature into every Gaussian, extending the feature-field distillation idea of DFF [32], N3F [33], and LERF [34] to Gaussian primitives: LangSplat [5], LangSplatV2 [6], LEGaussians [7], and 3D-VLGS [35] all bind each Gaussian to a CLIP embedding through tens of thousands of training iterations, producing a feature field tied to the training-time language model. A second family supervises a per-Gaussian identity feature through dense 2D masks in three forms: InfoNCE-style contrastive losses (OpenGaussian [8], Unified-Lift [17], and OpenSplat3D [16]), cross-entropy classification over 2D mask IDs (Gaussian Grouping [9] and ObjectGS [12]) and 3D-aware memory-bank grouping (Gaga [15]). All of them commit each Gaussian to a single identity mode, enforced across frames by the training loss. A third option uses a binary Bayesian posterior [13], but only for one target at a time; this is itself the binary special case of a longer lineage of per-primitive Bayesian semantic aggregation on surfels [36] and voxels [37, 38], whose multi-class Dirichlet–Categorical form we adopt as the starting point of our voting rule. DIGS introduces a fourth option that none of the above provides: an explicit multi-peaked discrete distribution over instance identities, maintained per Gaussian, updated by a closed-form tempered Bayesian rule, and persistent across an arbitrary number of subsequent text queries.

Open-vocabulary querying and foundation models. Recent 3DGS open-vocabulary systems rely on vision–language foundation models such as CLIP [39] and open-set detectors [40] to connect text prompts with image content; methods that bind these models into the 3D representation at training time inherit a tight coupling, since the choice of language model becomes part of the scene asset itself. DIGS instead decouples lifting from language entirely: the mapping stage is class-agnostic, and foundation models enter only at the downstream per-instance encoding step, so the same map serves any text-driven query interface without re-running the lifting.

3 Method

3.1 Overview

DIGS operates in two decoupled stages on a pre-reconstructed 3DGS scene (Figure 2). Stage 1 (class-agnostic mapping, §3.2–§3.5) builds a language-free instance ontology by distribution-informed per-Gaussian voting (top-M state under a tempered Dirichlet rule). Stage 2 (multi-modal instance encoding, §3.6) attaches a fused CLIP-visual / LLM-text embedding to each stable instance.

3.2 Preliminaries: Scene Representation and Rendering

Scene representation. The scene is represented by N 3D Gaussian primitives $\mathcal{G} = \{g_i\}_{i=1}^{N}$, each parameterised by a position $\boldsymbol{\mu}_i \in \mathbb{R}^3$, an anisotropic covariance $\boldsymbol{\Sigma}_i \in \mathbb{R}^{3\times3}$, a scalar opacity $o_i \in [0, 1]$, and view-dependent colour $\mathbf{c}_i$ encoded as spherical-harmonic coefficients. All four

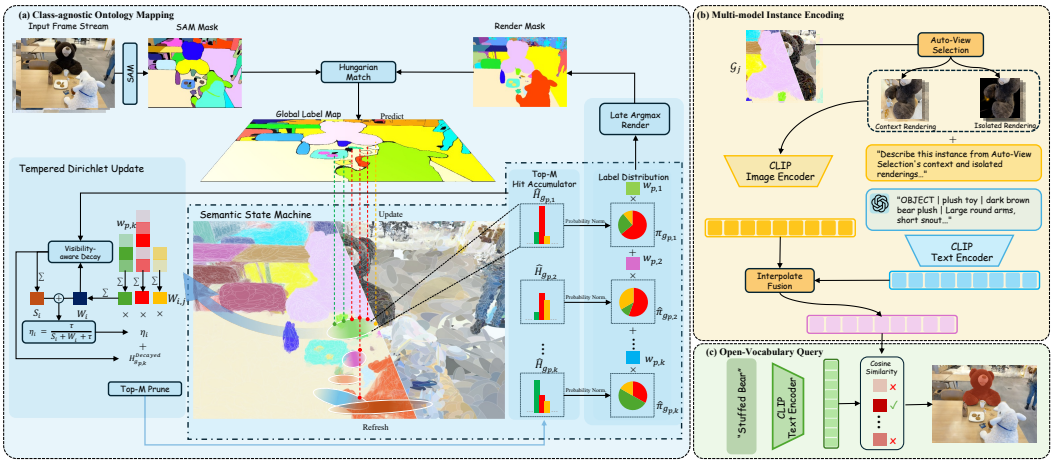


Figure 2: DIGS architecture. **(a) Class-agnostic ontology mapping**: per-frame SAM masks are Hungarian-matched to the Late-Argmax-rendered label map; the global label map drives the Tempered Dirichlet Update (injection bound $\eta_i^{(t)} \cdot W_i^{(t)} < \tau$, Proposition 1) on the semantic state machine, maintaining each Gaussian's top-M Hit Accumulator $\hat{H}_{g_{p,k}}$ and label distribution $\hat{\pi}_{g_{p,k}}$ alongside per-instance footprint $\mathcal{G}_j$ and lifecycle counters. **(b) Multi-modal instance encoding**: Auto-View Selection yields paired context / isolated renderings of $\mathcal{G}_j$, whose CLIP visual embedding is linearly fused with the CLIP-text embedding of an LLM-generated attribute description. **(c) Open-vocabulary query**: a text prompt is CLIP-text-encoded and matched to the per-instance embeddings by cosine similarity, with no per-query retraining.

are held fixed throughout the semantic pipeline; we augment g_i with a distribution-informed (DI) semantic state: an explicit top-M categorical label distribution per primitive. Its update rule and consumers are detailed in §3.3–§3.5.

Top-K geometry-aware rendering. We extend the CUDA rasteriser of 3DGS [4], following recent 3DGS segmentation pipelines that expose per-pixel Gaussian contributions during rendering [8, 9, 28], to output alongside the rendered RGB image the top-K contributing Gaussian indices and their alpha-compositing weights for every pixel:

$$\mathcal{G}_p = \{(g_{p,k}, w_{p,k})\}_{k=1}^{K}, \quad w_{p,k} = o_{p,k} \prod_{k'<k} (1 - o_{p,k'}), \tag{1}$$

where $o_{p,k}$ is the effective opacity of $g_{p,k}$ at pixel p and $w_{p,k}$ is the resulting alpha-compositing weight. This top-K correspondence acts as both the back-projection operator for voting and the forward operator for rendering, with no learned matching.

Class-agnostic 2D observation. For each frame, SAM is applied in automatic mask generation mode to produce class-agnostic 2D instance masks $\mathcal{S}^{(t)} = \{\mathbf{\Omega}_i\}_{i=1}^{n_s}$. A mask-NMS post-processing step removes redundant highly-overlapping masks. These masks carry no category information; class attribution is deferred to the query stage (§3.6).

3.3 Tempered Dirichlet Foundation

Both the state machine (§3.4) and the voting pipeline (§3.5) read from a per-Gaussian evidence accumulator $\mathbf{H}_i \in \mathbb{R}^J_{\geq 0}$ updated from rendered SAM masks across frames ($\mathcal{J} = \{0, \ldots, J-1\}$ is the label registry, with 0 reserved for background; J is distinct from the per-pixel top-K count of Eq. (1)). Naive additive accumulation $\mathbf{H}_i \mathrel{+}= \mathbf{W}_i^{(t)}$ gives high-weight observations disproportionate influence and permits arbitrarily large evidence increments in one frame. In a closed loop, these increments can reinforce early association errors. We cap each frame's contribution with a closed-form tempered update.

Tempered update. Motivated by a Dirichlet–Categorical evidence model and generalised Bayesian tempering [41], we use a per-Gaussian per-frame learning rate

$$\eta_i^{(t)} = \frac{\tau}{S_i + W_i^{(t)} + \tau}, \quad S_i = \textstyle\sum_j H_{i,j}, \quad W_i^{(t)} = \textstyle\sum_j W_{i,j}^{(t)} \tag{2}$$

yielding the hit-accumulator evolution

$$H_{i,j} \leftarrow H_{i,j} + \eta_i^{(t)} W_{i,j}^{(t)} \tag{3}$$

which executes in a single parallel pass per frame. The rate depends on τ and the current evidence, without a hand-designed η schedule.

Injection bound. The structural property that underwrites the rest of the method is below:

Proposition 1 (Injection bound). *For $\tau > 0$, $S_i \geq 0$, and $W_i^{(t)} \geq 0$, we have $\eta_i^{(t)} \cdot W_i^{(t)} < \tau$.*

One-step properties. Proposition 1 bounds the evidence injected by each voting step; it does not establish correct association or closed-loop convergence. At zero stored evidence, the injected mass approaches τ as the vote grows (Proposition 2). When stored evidence dominates the current vote and τ, the learning rate decreases approximately as τ/S_i (Proposition 3). Proofs and implementation-dependent qualifications are in Appendix A.2.

3.4 Bidirectional Two-Level State Machine

DIGS maintains a class-agnostic instance ontology at two bidirectionally-coupled levels: each Gaussian g_i carries a top-M label distribution (DI) derived from $\mathbf{H}_i$, and each non-background registry slot $j \in \mathcal{J} \setminus \{0\}$ carries a lifecycle record. Boundary straddling and occlusion fragmentation can split a Gaussian's evidence across multiple instance labels, motivating top-M evidence retention over argmax.

Gaussian-level semantic state. Let $j_{i,1}, \ldots, j_{i,M}$ denote the indices of the M largest entries of $\mathbf{H}_i$ in descending order and let $\mathcal{T}_i = \{j_{i,m}\}_{m=1}^{M}$. Each g_i stores a hard top-1 label ℓ_i (set to background 0 before any evidence is received) and a normalised top-M evidence distribution supported on $\mathcal{T}_i$,

$$\hat{\pi}_i(j) = \begin{cases} \dfrac{H_{i,j}\mathbb{1}[j \in \mathcal{T}_i]}{\sum_{j' \in \mathcal{T}_i} H_{i,j'}}, & \sum_{j' \in \mathcal{T}_i} H_{i,j'} > 0, \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

which sums to one when the Gaussian has positive evidence; an uninitialised Gaussian contributes no instance vote. The distribution retains relative support for multiple labels when evidence is split. We write $\hat{\pi}_{g_{p,k}}(j) := \hat{\pi}_i(j)$ with $i = g_{p,k}$ as the pixel-contributor shorthand used by the Late-Argmax renderer (Eq. (6)). After each voting step we sparsify $\mathbf{H}_i$ to its M-support; computing $\eta_i^{(t)}$ from this sparse state preserves the voting-increment bound (Appendix B.2).

Instance lifecycle and footprint. Each discovered instance j is characterised by a *Gaussian-space footprint*

$$\mathcal{G}_j = \{\, g_i \in \mathcal{G} : \ell_i = j \,\} \tag{5}$$

(the subset of primitives committing to j), and a lifecycle record with two counters: match_j (incremented on a successful Hungarian assignment) and miss_j (incremented when j is visible but unmatched, reset on a match). When $\text{miss}_j > \theta_{\text{miss}}$, instance j becomes a *zombie*; as the registry approaches capacity J, zombies are recycled first (longest-missing preferred), keeping the ID registry within a fixed maximum capacity J independent of sequence length.

3.5 Distribution-Preserving Voting Pipeline

The voting pipeline drives the state machine (§3.4) forward under streaming observations, preserving the top-M distribution end-to-end.

Predict step. Before observing the current frame, the state machine renders a predicted label map $\hat{L}^{(t)}$ (Eq. (6)) with predicted regions $\{R_j\}$. SAM masks $\{\mathbf{\Omega}_i\}$ are associated with global instance IDs by Hungarian bipartite matching on an IoU cost; an IoU threshold θ_{iou} decides new-ID allocation versus match, and a greedy fallback permits many-to-one matching under occlusion-induced fragmentation. The result is the global label map $\mathcal{L}^{(t)} \in \mathbb{Z}^{H \times W}$ consumed by the update step.

Update step. Given $\mathcal{L}^{(t)}$, the tempered Dirichlet update (Eq. (3)) refreshes $\mathbf{H}$ in a single parallel pass. Before voting, a visibility-aware leaky integrator with decay rate $\gamma \in (0, 1)$ decays only substantially-visible Gaussians ($W_i^{(t)} > w_{\min}$), so occluded or out-of-view primitives retain their evidence across long traversals.

Late-Argmax rendering. Rather than committing each Gaussian to a hard label before rendering (Appendix B.2), we fuse the full top-M distribution at the pixel level:

$$\hat{\ell}(p) = \arg\max_{j>0} \sum_{k=1}^{K} w_{p,k}\, \hat{\pi}_{g_{p,k}}(j). \tag{6}$$

When every instance score is zero, the output label is background (0). Boundary Gaussians, which typically hold comparable evidence for adjacent labels, contribute proportionally to both sides and the final label is decided by the majority vote of neighbouring non-ambiguous Gaussians; this suppresses the staircase boundary artefacts that characterise hard-label rendering. The output $\hat{L}^{(t)}$ feeds back into the next frame's predict step, closing the loop.

3.6 Multi-Modal Spatial Instance Representation

Stage 2 attaches a fixed multi-modal embedding to each stable instance: the CLIP visual embedding of its isolated 3D footprint $\mathcal{G}_j$ (Eq. (5)), fused with the CLIP-text embedding of an LLM-generated attribute description. Computed once post-mapping, the embeddings support arbitrary downstream retrieval without re-running the voting pipeline.

CLIP visual retrieval. We retain the *stable subset* $\mathcal{I}_{\text{stable}} = \{j : \text{match}_j \geq \theta_{\text{match}} \wedge \text{miss}_j \leq \theta_{\text{miss}}\}$ to discard short-lived SAM fragments and zombie slots. For each $j \in \mathcal{I}_{\text{stable}}$ we render $\mathcal{G}_j$ alone (all other Gaussians at zero opacity) from its top-B training views $\mathcal{V}_j$ (ranked by projected area), crop to the instance bounding box, and compute a CLIP visual embedding

$$\mathbf{e}_j = \frac{\bar{\mathbf{c}}_j}{\|\bar{\mathbf{c}}_j\|_2}, \quad \bar{\mathbf{c}}_j = \frac{1}{|\mathcal{V}_j|} \sum_{t \in \mathcal{V}_j} \text{CLIP}_{\text{img}}\big(\text{Crop}(\text{IsoRender}(\mathcal{G}_j, \text{cam}_t))\big) \tag{7}$$

Isolated rendering prevents neighbouring-instance leakage into the embedding. At query time, the prompt is matched by cosine similarity, with $j^* = \arg\max_{j \in \mathcal{I}_{\text{stable}}} \cos(\mathbf{e}_j, \text{CLIP}_{\text{text}}(\mathbf{q}))$, in $O(|\mathcal{I}_{\text{stable}}|)$ cosines per prompt; the predicted mask is then obtained by indexing the Late-Argmax label map $\hat{L}_{\text{test}}$ rendered at the test viewpoint $\hat{M}_{\mathbf{q}} = \mathbb{1}[\hat{L}_{\text{test}} = j^*]$.

LLM-enhanced fusion. A multi-modal LLM provides an attribute-rich text description per instance, CLIP-text-encoded into a complementary language channel that attaches *post hoc* to the language-free ontology. The same isolated crops as in Eq. (7) are passed to the LLM with a fixed prompt, producing description d_j encoded into $\mathbf{e}_j^{\text{txt}}$; the fused embedding is

$$\mathbf{e}_j^{\text{fused}} = \lambda\, \mathbf{e}_j + (1 - \lambda)\, \mathbf{e}_j^{\text{txt}}, \quad \lambda \in [0, 1] \tag{8}$$

Retrieval uses $\mathbf{e}_j^{\text{fused}}$ in place of $\mathbf{e}_j$; no second voting pass or re-optimisation is required (§4.3 quantifies the gain).

Detection-bridging (optional). When the evaluation protocol supplies the test-frame RGB, an open-vocabulary 2D detector can resolve the what, while the 3D instance field supplies the where via IoU matching against $\hat{L}_{\text{test}}$. The full specification is in Appendix B.4.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluated on four open-vocabulary 3D segmentation benchmarks chosen to probe complementary aspects of the mapping pipeline. **LERF-OVS** [8] (4 scenes) follows the OpenGaussian 3D-native protocol with fine-grained, attribute-rich queries on cluttered scenes; it emphasizes 3D open-vocabulary retrieval, requiring that the correct instance be both localised and fully segmented, with mIoU and mAcc@0.25 as metrics. **LERF-Mask** [9] (3 scenes) follows the Gaussian-Grouping protocol with test-frame binary masks for short-noun queries; it stresses 2D mask boundary precision,

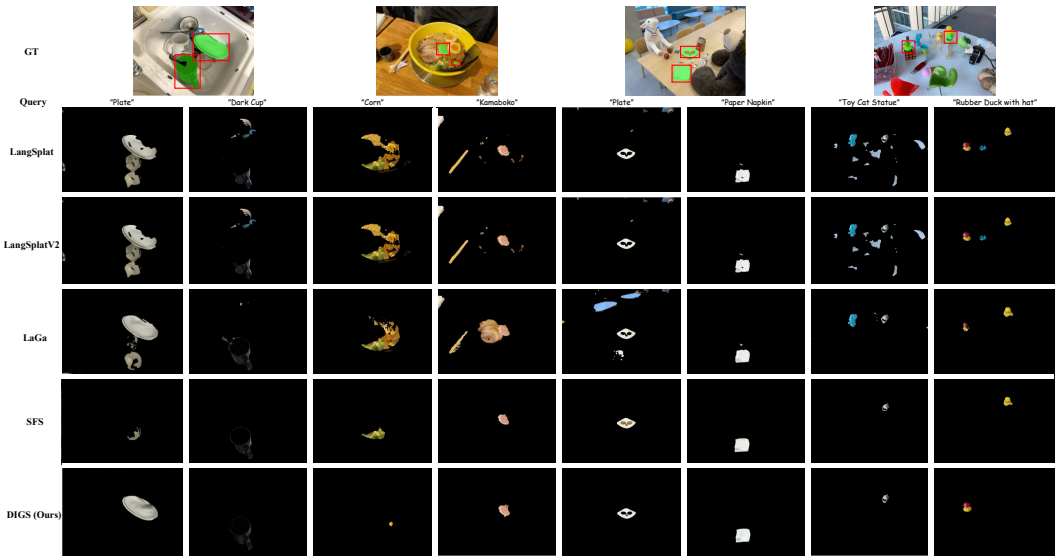


Figure 3: Qualitative comparison on LERF-OVS against LangSplat, LangSplatV2, LaGa, and the training-free comparator SFS. Two prompts per scene across the four LERF-OVS scenes.

with mIoU and the boundary-sensitive mBIoU as metrics. **3D-OVS** [42] (5 scenes) provides 6–7 free-form language queries per scene on simple, spatially separated objects; we follow the original protocol and report mIoU only. **Replica** [43] (8 scenes) provides long indoor trajectories with revisits and many repeated, near-identical objects; we follow the protocols of Gaga [15] and ObjectGS [12]. Across all benchmarks, we follow the corresponding evaluation protocols and match evaluation settings across methods to ensure fair comparisons.

Implementation details. The 3DGS reconstruction was a fixed Mip-Splatting [44] model trained for 30,000 iterations to fit the RGB scene; the resulting Gaussians were frozen for the rest of the pipeline. We used SAM ViT-H [14] for class-agnostic masks, Open CLIP ViT-B/16 [39] for visual/textual embeddings, GroundingDINO Swin-B [40] for detection-bridging box prompting, and GPT-4.1 for per-instance LLM descriptions. Stage 1 (mapping) ran once per scene with a single hyperparameter set shared across all benchmarks (τ=5, M=3, γ=0.95, θ_{iou}=0.4 (0.15 on Replica's longer trajectories), and λ=0.7), producing a reusable instance ontology. All experiments ran on one NVIDIA A100.

4.2 Quantitative Results

Computational cost. The numbers in Table 1 are per-scene means on a single A100; LangSplat/LEGaussians figures are taken from [6]. Training-based baselines bind language features via 1–4 h of per-scene gradient training, whereas DIGS is gradient-free: DIGS + (1) completes class-agnostic mapping and CLIP encoding in **1.00** min by collapsing the gradient loop into a single closed-form Dirichlet pass per frame (225× faster than LaGa and $\sim 6\times$ faster than the closest training-free comparator SFS), and the LLM-fusion variant DIGS + (1) + (2) adds ~ 2.2 min dominated by GPT-4.1 API I/O rather than local compute, still 70× faster than the strongest distillation baseline. VRAM drops from a 12.4 GB transient peak during mapping to a 5.4 GB steady state once the per-instance embeddings are computed. At inference, the pre-computed per-instance embeddings reduce each query to $O(|\mathcal{I}_{\text{stable}}|)$ cosine similarities (mean $\approx$ 33) plus a Late-Argmax label-

Table 1: Per-scene computational cost on LERF-OVS. Units: Map (min), VRAM (GB), Per-prompt, and 3D-only (ms).

Method	Train	Map	VRAM	Per-prompt	3D-only
LangSplat [5]	✓	60	6.2	–	–
LEGaussians [7]	✓	78	11.0	–	–
LangSplatV2 [6]	✓	204	16.8	71.4	–
LaGa [10]	✓	225	12.5	101.7	72.8
SFS [11]	×	5.8	13.3	161.0	8.0
DIGS (map)	×	0.91	12.4	–	–
(1) +CLIP enc.	×	+0.09	6.6	–	–
(2) +LLM fus.	×	+2.2	5.8	–	–
DIGS + (1)	×	**1.00**	**5.4**	**28.1**	**5.4**
DIGS + (1) + (2)	×	**3.2**	**5.4**	**28.1**	**5.4**

Table 2: LERF-OVS (mIoU / mAcc, %). OpenGaussian (OG) 3D-native protocol unless ‡(LangSplat-family 2D). *re-eval by OG [8]; †our reproduction; unmarked: paper-reported.

Method	Figurines		Ramen		Teatime		Waldo_kitchen		Mean	
	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc
LangSplat* [5]	10.16	8.93	7.92	11.27	11.38	20.34	9.18	9.09	9.66	12.41
LangSplat‡ [5]	46.44	73.21	48.96	67.61	60.85	84.75	39.65	63.64	48.98	72.30
LEGaussians* [7]	17.99	23.21	15.79	26.76	19.27	27.12	11.78	18.18	16.21	23.82
OpenGaussian* [8]	39.29	55.36	31.01	42.25	60.44	76.27	22.70	31.82	38.36	51.43
LaGa [10]	64.10	–	55.60	–	70.90	–	65.60	–	64.00	–
LaGa† [10]	54.43	69.30	54.86	79.50	68.53	74.90	64.37	70.80	60.55	73.63
LangSplatV2† [6]	3.92	0.00	11.07	11.27	13.50	16.95	12.64	9.09	10.28	9.33
LangSplatV2‡ [6]	57.20	71.43	48.21	59.15	69.91	84.75	57.74	68.18	58.27	70.88
SuperGSeg [45]	43.68	60.71	18.07	23.94	55.31	77.97	26.71	45.45	35.94	52.02
Splat Feature Solver [11]	67.60	–	62.30	–	68.50	–	62.10	–	65.10	–
Splat Feature Solver† [11]	63.43	76.79	59.60	67.61	71.47	**88.14**	54.14	72.73	62.16	76.32
DIGS (CLIP)	74.81	91.07	56.15	73.24	74.70	84.75	61.72	81.82	66.85	82.72
DIGS (LLM+CLIP)	**78.71**	**92.86**	**64.68**	**80.28**	**77.65**	86.44	**66.76**	**81.82**	**71.95**	**85.35**

Table 3: LERF-Mask (mIoU / mBIoU, %) under the Gaussian Grouping protocol.

Method	Figurines		Ramen		Teatime		Mean	
	mIoU	mBIoU	mIoU	mBIoU	mIoU	mBIoU	mIoU	mBIoU
LERF [34]	33.5	30.6	28.3	14.7	49.7	42.6	37.2	29.3
SA3D [22]	24.9	23.8	7.4	7.0	42.5	39.2	24.9	23.3
LangSplat [5]	52.8	50.5	50.4	44.7	69.5	65.6	57.6	53.6
GG [9]	69.7	67.9	77.0	68.8	71.7	66.1	72.8	67.6
FlashSplat (U) [28]	60.2	57.5	68.4	61.5	80.4	76.3	69.6	65.1
FlashSplat (F) [28]	71.6	69.1	71.4	66.3	86.6	83.9	76.5	73.1
Gaga [15]	90.7	89.0	64.1	61.6	69.3	66.0	74.7	72.2
ObjectGS [12]	88.2	85.2	88.0	79.9	88.9	88.6	88.4	84.6
Unified-Lift [17]	–	–	–	–	–	–	80.9	77.1
B^3-Seg [13]	88.3	85.4	75.3	69.7	89.8	88.0	84.5	81.0
DIGS (Ours)	**92.73**	**91.02**	**90.79**	**80.04**	**91.61**	**89.67**	**91.71**	**86.91**

map lookup, giving 28.1 ms (prompt $\rightarrow$ 2D mask) / 5.4 ms (3D-only) latency, 2.5–$5.7\times$ faster than LangSplatV2/LaGa/SFS.

LERF-OVS. Table 2 reports DIGS at 71.95 mean mIoU and 85.35 mean mAcc on LERF-OVS under the OpenGaussian 3D-native protocol, exceeding the strongest training-based baseline LaGa by +11.40 mIoU and the closest training-free comparator SFS by +9.79 mIoU; against LangSplatV2‡'s strongest reported 58.27, the gap widens to $+13.68$ (its OG row drops to 10.28). The gain decomposes cleanly: DIGS (CLIP) alone reaches 66.85 mean mIoU, isolating the boundary-preserving top-M representation as the primary lever; LLM fusion contributes a further $+5.10$, with DIGS leading on every scene and the largest gap on attribute-dense Figurines. Figure 3 visualises two prompts per scene across all four LERF-OVS scenes, exposing two complementary failure modes. On near-synonymous prompts (e.g., *rubber duck with hat* versus *with buoy*), all CLIP-based baselines including SFS retrieve the wrong instance because their CLIP embeddings collapse to near-identical points, whereas DIGS's LLM-text fusion disambiguates the attribute. On prompts where SFS does identify the correct instance (e.g., *toy cat statue*), its closed-form lift still yields fuzzy mask edges, whereas DIGS's distribution-informed late-argmax produces tight boundaries.

LERF-Mask. With test-frame RGB available, we evaluated DIGS in detection-bridging mode (§3.6). Pairing the Stage 1 ontology with GroundingDINO+SAM isolates mapping quality from the CLIP/LLM query pathway. Table 3 shows DIGS at 91.71 mean mIoU and 86.91 mean mBIoU, +3.3 mIoU over the previous best ObjectGS and leading every prior method on both mIoU and mBIoU across all three scenes; the dominance on the boundary-sensitive mBIoU metric directly evidences the boundary-fidelity gain from top-M + late-argmax, addressing the boundary failure flagged in §1.

3D-OVS. In the detection-bridging query mode (§3.6), Table 4 reports DIGS at 96.45 mean mIoU on 3D-OVS, the highest mean across all prior methods, surpassing the strongest training-based baselines ObjectGS [12] (96.4) and LaGa [10] (95.3) without any per-scene gradient training. The strongest methods already exceed 95% mean mIoU on these spatially separated objects, leaving limited headroom. Appendix E.1 (Figure 6) visualises multi-query coverage across all five scenes.

Table 4: Quantitative results on 3D-OVS (mIoU, %).

Method	Bed	Bench	Lawn	Room	Sofa	Mean
LERF [34]	73.5	53.2	73.7	46.6	27.0	54.8
3D-OVS [42]	89.5	89.3	88.2	92.8	74.0	86.8
GG [9]	83.0	91.5	90.6	85.9	87.3	87.7
LEGaussians [7]	84.9	91.1	92.5	86.0	87.8	88.5
LangSplat [5]	92.5	94.2	96.1	94.1	90.0	93.4
LangSplatV2 [6]	93.0	94.9	96.6	96.1	92.3	94.6
LaGa [10]	96.8	92.8	**96.9**	97.0	93.0	95.3
ObjectGS [12]	**98.0**	**96.4**	95.1	**97.2**	**95.4**	96.4
DIGS (Ours)	97.63	96.18	96.86	96.40	95.15	**96.45**

Replica. We evaluate all eight scenes under two established protocols. Gaga's protocol evaluates class-agnostic instance segmentation on held-out views via Hungarian matching, while ObjectGS's panoptic-lifting protocol evaluates rendered labels on held-out views. The 2D inputs are matched across methods within each protocol: SAM ViT-H masks for Gaga's and ground-truth masks for ObjectGS's.

Table 5: Replica results under two protocols (%).

		Gaga protocol			ObjectGS protocol		
Method	Train	IoU	Prec.	Rec.	IoU	Dice	Acc.
GG [9]	✓	21.76	25.00	19.72	83.36	91.84	94.70
Gaga [15]	✓	46.21	39.53	50.62	–	–	–
ObjectGS [12]	✓	–	–	–	88.39	92.39	95.65
DIGS (Ours)	×	**51.39**	**42.48**	**57.53**	**91.46**	**95.15**	**98.68**

Table 5 shows that DIGS improves mean IoU over Gaga by 5.18 points (51.39 vs. 46.21) and over ObjectGS by 3.07 points (91.46 vs. 88.39), while also leading on the remaining metrics in both protocols. Appendix C reports the per-scene results and mapping statistics. The gains in complex indoor scenes with repeated, visually similar objects and exploratory camera trajectories support generalization beyond the LERF benchmarks.

4.3 Component Ablation



Figure 4: Hard-argmax versus Late-Argmax rendering on LERF-OVS.

On LERF-OVS, we incrementally enabled stability filter (SF), distribution-preserving (DP) design, and LLM fusion atop a minimal baseline to isolate each component's contribution to retrieval mIoU and verify that the three are complementary. Table 6 summarises the contributions. SF (a → b) adds +6.03 mIoU by discarding 92–97% of unstable SAM proposals (Appendix D.4). DP (b → c) is the largest single jump (+8.84), preserving multi-peaked label distributions at boundary primitives; Figure 4 shows the boundary-cleanliness gain on Figurines and Teatime. LLM fusion (c → d) adds a further +5.10, peaking at +8.53 on Ramen where dense visually similar food instances need attribute-level disambiguation; Figure 5 illustrates one prompt per scene where the LLM-text channel disambiguates instances that share near-identical CLIP visual embeddings. The three are complementary, taking mean mIoU from 51.98 to 71.95.

Table 6: Component ablation on LERF-OVS (mIoU, %). SF: stability filter; DP: distribution-preserving design. Tempered Dirichlet is enabled in all rows.

Row	SF	DP	LLM	Figurines	Ramen	Teatime	Waldo_kitchen	Mean
(a)	×	×	×	64.77	32.45	60.17	50.53	51.98
(b)	✓	×	×	65.09	42.30	72.71	51.93	58.01
(c)	✓	✓	×	74.81	56.15	74.70	61.72	66.85
(d)	✓	✓	✓	**78.71**	**64.68**	**77.65**	**66.76**	**71.95**

Table 7: Tempered Dirichlet ablation on Row (c) of Table 6 (mIoU, %).

Scene	Naive η	Temp. η	Δ
Figurines	68.71	74.81	+6.10
Ramen	40.94	56.15	+15.22
Teatime	73.30	74.70	+1.40
Waldo_kitchen	52.23	61.72	+9.49
Overall	58.80	**66.85**	+8.05

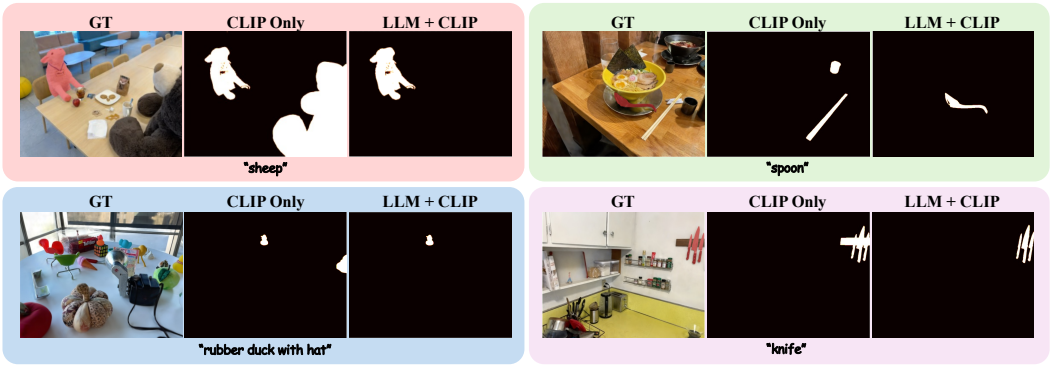


Figure 5: CLIP-only versus LLM-enhanced retrieval on LERF-OVS.

Tempered Dirichlet. Table 7 reports the effect of replacing $\eta_i^{(t)}$ by η=1 on Row (c) of Table 6: mean mIoU drops by 8.05 points. The drop is larger on scenes with cross-view SAM instability (Ramen and Waldo_kitchen, 12.35 points) than on scenes with more stable masks (Figurines and Teatime, 3.75 points). Per-class results are given in Appendix D.5. These gains indicate that tempered evidence updates improve robustness to cross-view mask instability.

5 Conclusion

In this paper, we present DIGS, a training-free framework for open-vocabulary 3D Gaussian segmentation. Existing methods bind a single semantic feature to each Gaussian via per-scene gradient training, blurring boundary Gaussians and locking each scene to one language model. We addressed this with two designs. First, a distribution-informed per-Gaussian representation: each primitive stores a top-M label distribution updated under a tempered Dirichlet rule with a provably bounded per-frame injection, allowing a late-argmax renderer to recover clean instance boundaries. Second, a streaming voting pipeline that updates a bidirectional state machine into a class-agnostic instance ontology in one closed-form Dirichlet pass per frame, removing per-scene gradient training. Open-vocabulary querying is a lightweight post-hoc step fusing each instance's CLIP visual embedding with an LLM-generated attribute description. On LERF-OVS, DIGS maps a scene in 1–3 minutes on a single A100 and serves native 3D queries at 5.4 ms latency. It achieves state-of-the-art results on LERF-OVS, LERF-Mask, 3D-OVS, and Replica, while its CLIP-only configuration maps LERF-OVS scenes 225× faster than the strongest training-based baseline.

Limitations and perspectives. DIGS depends on its foundation-model stack: SAM's mask granularity caps the finest discoverable instance, while GroundingDINO, CLIP, and LLM jointly bound linguistic discrimination. Fusing multi-granularity SAM masks into a hierarchical ontology under the same voting machinery is a natural next step. A second limitation is that the pre-reconstructed Gaussian geometry is fixed throughout the semantic pipeline; jointly refining geometry and instance evidence may further improve boundary fidelity on transparent or thin structures.

Acknowledgments

This work was supported by the UNSW HDR Development and Research Training Grant. We acknowledge computational support from UNSW Sydney's Katana cluster and NCI Australia.

References

[1] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied AI research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9338–9346, 2019. doi: 10.1109/ICCV.2019.00943.

[2] Ying Jiang, Chang Yu, Tianyi Xie, Xuan Li, Yutao Feng, Huamin Wang, Minchen Li, Henry Lau, Feng Gao, Yin Yang, and Chenfanfu Jiang. VR-GS: A physical dynamics-aware interactive Gaussian splatting system in virtual reality. In *ACM SIGGRAPH 2024 Conference Papers*, 2024. doi: 10.1145/3641519.3657448. Article 78.

[3] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. GaussianEditor: Swift and controllable 3D editing with Gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21476–21485, 2024.

[4] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139:1–139:14, 2023. doi: 10.1145/3592433.

[5] 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*, pages 20051–20060, 2024.

[6] Wanhua Li, Yujie Zhao, Minghan Qin, Yang Liu, Yuanhao Cai, Chuang Gan, and Hanspeter Pfister. LangSplatV2: High-dimensional 3D language Gaussian splatting with 450+ FPS. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

[7] 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*, pages 5333–5343, 2024.

[8] Yanmin Wu, Jiarui Meng, Haijie Li, Chenming Wu, Yahao Shi, Xinhua Cheng, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, and Jian Zhang. OpenGaussian: Towards point-level 3D Gaussian-based open vocabulary understanding. *Advances in Neural Information Processing Systems*, 37:19114–19138, 2024.

[9] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3D scenes. In *European Conference on Computer Vision*, pages 162–179. Springer, 2024.

[10] Jiazhong Cen, Xudong Zhou, Jiemin Fang, Changsong Wen, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Tackling view-dependent semantics in 3D language Gaussian splatting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pages 7013–7034. PMLR, 2025.

[11] Butian Xiong, Rong Liu, Kenneth Xu, Meida Chen, and Andrew Feng. Splat feature solver. In *International Conference on Learning Representations*, 2026.

[12] Ruijie Zhu, Mulin Yu, Linning Xu, Lihan Jiang, Yixuan Li, Tianzhu Zhang, Jiangmiao Pang, and Bo Dai. ObjectGS: Object-aware scene reconstruction and scene understanding via Gaussian splatting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8350–8360, 2025.

[13] Hiromichi Kamata, Samuel Arthur Munro, and Fuminori Homma. B^3-Seg: Camera-free, training-free 3DGS segmentation via analytic EIG and Beta-Bernoulli Bayesian updates. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26867–26876, 2026.

[14] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloé Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3992–4003, 2023. doi: 10.1109/ICCV51070.2023.00371.

[15] Weijie Lyu, Xueting Li, Abhijit Kundu, Yi-Hsuan Tsai, and Ming-Hsuan Yang. Gaga: Group any Gaussians via 3D-aware memory bank. *Transactions on Machine Learning Research*, 2026.

[16] Jens Piekenbrinck, Christian Schmidt, Alexander Hermans, Narunas Vaskevicius, Timm Linder, and Bastian Leibe. OpenSplat3D: Open-vocabulary 3D instance segmentation using Gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 5246–5255, 2025. doi: 10.1109/CVPRW67362.2025.00520.

[17] Runsong Zhu, Shi Qiu, Zhengzhe Liu, Ka-Hei Hui, Qianyi Wu, Pheng-Ann Heng, and Chi-Wing Fu. 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*, pages 3656–3665, 2025.

[18] Yiren Lu, Yunlai Zhou, Yiran Qiao, Chaoda Song, Tuo Liang, Jing Ma, Huan Wang, and Yu Yin. Segment then splat: Unified 3D open-vocabulary segmentation via Gaussian splatting. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

[19] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15818–15827, 2021. doi: 10.1109/ICCV48922.2021.01554.

[20] Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Buló, Norman Müller, Matthias Nießner, Angela Dai, and Peter Kontschieder. Panoptic lifting for 3D scene understanding with neural fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9043–9052, 2023.

[21] Yash Bhalgat, Iro Laina, João F. Henriques, Andrea Vedaldi, and Andrew Zisserman. Contrastive lift: 3D object instance segmentation by slow-fast contrastive fusion. In *Advances in Neural Information Processing Systems*, volume 36, pages 9092–9116, 2023.

[22] Jiazhong Cen, Zanwei Zhou, Jiemin Fang, Chen Yang, Wei Shen, Lingxi Xie, Dongsheng Jiang, Xiaopeng Zhang, and Qi Tian. Segment anything in 3D with NeRFs. *Advances in Neural Information Processing Systems*, 36:25971–25990, 2023.

[23] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.

[24] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3D: Mask transformer for 3D semantic instance segmentation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8216–8223. IEEE, 2023.

[25] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. OpenScene: 3D scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 815–824, 2023.

[26] Runyu Ding, Jihan Yang, Chuhui Xue, Wenqing Zhang, Song Bai, and Xiaojuan Qi. PLA: Language-driven open-vocabulary 3D scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7010–7019, 2023.

[27] Ayça Takmaz, Elisabetta Fedele, Robert W Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. OpenMask3D: Open-vocabulary 3D instance segmentation. In *Advances in Neural Information Processing Systems*, volume 36, pages 68367–68390, 2023.

[28] Qiuhong Shen, Xingyi Yang, and Xinchao Wang. FlashSplat: 2D to 3D Gaussian splatting segmentation solved optimally. In *European Conference on Computer Vision*, pages 456–472. Springer, 2024.

[29] Jiahuan Cheng, Jan-Nico Zaech, Luc Van Gool, and Danda Pani Paudel. Occam's LGS: An efficient approach for language Gaussian splatting. In *Proceedings of the British Machine Vision Conference*, 2025.

[30] Luis Wiedmann, Luca Wiehe, and David Rozenberszki. DCSEG: Decoupled 3D open-set segmentation using Gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 5217–5226, 2025. doi: 10.1109/CVPRW67362.2025.00517.

[31] Jiayu Ding, Xinpeng Liu, Zhiyi Pan, Shiqiang Long, and Ge Li. ExtrinSplat: Decoupling geometry and semantics for open-vocabulary understanding in 3D Gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 31019–31028, 2026.

[32] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing NeRF for editing via feature field distillation. *Advances in Neural Information Processing Systems*, 35:23311–23330, 2022.

[33] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural feature fusion fields: 3D distillation of self-supervised 2D image representations. In *2022 International Conference on 3D Vision (3DV)*, pages 443–453. IEEE, 2022.

[34] 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*, pages 19672–19682, 2023. doi: 10.1109/ICCV51070.2023.01807.

[35] Qucheng Peng, Benjamin Planche, Zhongpai Gao, Meng Zheng, Anwesa Choudhuri, Terrence Chen, Chen Chen, and Ziyan Wu. 3D vision-language Gaussian splatting. In *International Conference on Learning Representations*, 2025.

[36] John McCormac, Ankur Handa, Andrew Davison, and Stefan Leutenegger. SemanticFusion: Dense 3D semantic mapping with convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4628–4635. IEEE, 2017.

[37] Lu Gan, Ray Zhang, Jessy W Grizzle, Ryan M Eustice, and Maani Ghaffari. Bayesian spatial kernel smoothing for scalable dense semantic mapping. *IEEE Robotics and Automation Letters*, 5(2):790–797, 2020.

[38] Joey Wilson, Yuewei Fu, Joshua Friesen, Parker Ewen, Andrew Capodieci, Paramsothy Jayakumar, Kira Barton, and Maani Ghaffari. ConvBKI: Real-time probabilistic semantic mapping network with quantifiable uncertainty. *IEEE Transactions on Robotics*, 40:4648–4667, 2024.

[39] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 8748–8763. PMLR, 2021.

[40] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pages 38–55. Springer, 2024. doi: 10.1007/978-3-031-72970-6_3.

[41] Pier Giovanni Bissiri, Chris C Holmes, and Stephen G Walker. A general framework for updating belief distributions. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 78(5):1103–1130, 2016.

[42] Kunhao Liu, Fangneng Zhan, Jiahui Zhang, Muyu Xu, Yingchen Yu, Abdulmotaleb El Saddik, Christian Theobalt, Eric P. Xing, and Shijian Lu. Weakly supervised 3D open-vocabulary segmentation. In *Advances in Neural Information Processing Systems*, volume 36, pages 53433–53456, 2023.

[43] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

[44] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-Splatting: Alias-free 3D Gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19447–19456, 2024.

[45] Siyun Liang, Sen Wang, Kunyi Li, Michael Niemeyer, Stefano Gasperini, Hendrik P. A. Lensch, Nassir Navab, and Federico Tombari. SuperGSeg: Open-vocabulary 3D segmentation with structured super-Gaussians. In *2026 International Conference on 3D Vision*, pages 1650–1659, 2026. doi: 10.1109/3DV69130.2026.00157.

[46] Peter Grünwald. The safe Bayesian: learning the learning rate via the mixability gap. In *International Conference on Algorithmic Learning Theory*, pages 169–183. Springer, 2012.

A Theoretical Derivations

This appendix supplies the proofs and derivations omitted from §3.3.

A.1 Tempered Dirichlet Update

We use a Dirichlet–Categorical model to motivate the evidence update. Let $\boldsymbol{\theta}_i \sim \mathrm{Dir}(\boldsymbol{\alpha}_i)$, $\boldsymbol{\alpha}_i^{(0)} = \frac{\tau}{J}\,\mathbf{1}_J$, $\tau > 0$. In an untempered, unpruned count model, $\alpha_{i,j} = H_{i,j} + \tau/J$ relates additive counts to the hit accumulator $\mathbf{H}_i$. In DIGS, however, the "observations" are not discrete counts but continuous rendering weights $W_{i,j}^{(t)} = \sum_{p:\, g_i \in \mathcal{G}_p,\, \mathcal{L}^{(t)}(p)=j} w_{p,i}$, and direct addition is unstable in two ways: *scale dominance*—a large-scale Gaussian can acquire a disproportionately large increment in one frame—and *closed-loop feedback*, where the rendered label map drives the next-frame association and can reinforce early errors.

Following the framework of generalised Bayesian inference [41], in the spirit of safe-Bayesian tempering of misspecified likelihoods [46], we introduce a per-Gaussian per-frame learning rate $\eta_i^{(t)}$. Within a voting step, after any visibility-aware decay and before top-M pruning, the corresponding tempered pseudo-count update is

$$\alpha_{i,j}^{(t)} = \alpha_{i,j}^{(t-1)} + \eta_i^{(t)} \cdot W_{i,j}^{(t)}. \tag{9}$$

Two design goals motivate our choice of $\eta_i^{(t)}$: (D1) cap the total evidence injected by one frame below τ; (D2) decrease the learning rate as accumulated evidence grows, without a manual schedule. Both are satisfied by Eq. (2), equivalently $\tau/(S_i + \tau + W_i^{(t)})$, yielding the hit-accumulator update in Eq. (3). The deployed state also undergoes visibility-aware decay and top-M pruning, so its normalised evidence distribution is not an exact Dirichlet posterior.

A.2 Proofs of Propositions 1–3

Proof of Proposition 1 (Injection bound). Since $S_i + \tau > 0$, we have $0 \le W_i^{(t)}/(S_i + W_i^{(t)} + \tau) < 1$ for every $W_i^{(t)} \ge 0$, including $W_i^{(t)} = 0$. Multiplying by τ gives the result.

Scope and conditional total-state bound. The one-step bound applies to non-negative votes even from erroneous associations and to retained top-M evidence ($S_i \ge 0$); it says nothing about association quality or the net state change after decay and pruning. Let $S_i^{(t)}$ denote retained evidence. If $S_i^{(0)} = 0$ and every positive vote triggers decay, then $S_i^{(t)} < \gamma S_i^{(t-1)} + \tau$ on positive-vote frames, while zero-vote frames cannot increase the state; hence $S_i^{(t)} \le \tau/(1-\gamma)$. The implementation omits decay when $0 < W_i^{(t)} \le w_{\min}$, so no uniform total-state bound holds for arbitrary streams: a fixed one-label vote $W_i^{(t)} = c \in (0, w_{\min}]$ gives $S_i^{(t)} = S_i^{(t-1)} + \tau c/(S_i^{(t-1)} + c + \tau)$, which grows without bound.

Proposition 2 (Scale suppression). *For fixed $\tau > 0$ and $S_i = 0$, let $m_i(W) = \eta_i^{(t)} W$ with $W_i^{(t)} = W$. Then $m_i(W)/W \to 1$ as $W/\tau \to 0$ for $W > 0$, and $m_i(W)/\tau \to 1$ as $W/\tau \to \infty$.*

Proof. With $S_i = 0$, the injected mass is $\tau W_i^{(t)}/(W_i^{(t)} + \tau)$. Its ratio to $W_i^{(t)}$ tends to one as $W_i^{(t)}/\tau \to 0$, while its ratio to τ tends to one as $W_i^{(t)}/\tau \to \infty$.

Proposition 3 (Evidence inertia). *For fixed $\tau > 0$, as $S_i \to \infty$ with $W_i^{(t)}/S_i \to 0$, $\eta_i^{(t)} = (\tau/S_i)(1 + o(1))$ and $\eta_i^{(t)} W_i^{(t)} = (\tau W_i^{(t)}/S_i)(1 + o(1))$. The rate remains positive at every finite state.*

Proof. Dividing the denominator of Eq. (2) by S_i gives $\eta_i^{(t)} = (\tau/S_i)/(1 + (W_i^{(t)} + \tau)/S_i)$. The denominator tends to one under the stated conditions. At finite S_i and $W_i^{(t)}$, it is positive, so $\eta_i^{(t)} > 0$.

Together, Propositions 1–3 characterise bounded injection, suppression of large votes at zero evidence, and attenuation under accumulated evidence. A positive learning rate at finite evidence permits later revision, but does not guarantee correction, top-1 stability, correct association, or closed-loop convergence.

B Implementation Details

B.1 Pipeline Pseudocode

Algorithm 1 summarises the mapping and encoding stages of DIGS. Stage 1 builds the class-agnostic instance ontology under the tempered Dirichlet voting rule; Stage 2 computes per-instance multi-modal embeddings once, post-mapping. The returned assets $\{\mathbf{e}_j^{\text{fused}}, \mathcal{G}_j, \ell_i\}$ are reusable across arbitrary downstream prompts.

B.2 Top-M State, Sparsification, and Rendering

Top-M state. Let $j_{i,1}, \ldots, j_{i,M}$ index the M largest entries of $\mathbf{H}_i$ in descending order. Define

$$\mathcal{T}_i = \{j_{i,1}, \ldots, j_{i,M}\} \subseteq \mathcal{J}, \tag{10}$$

$$\ell_i = \begin{cases} j_{i,1}, & S_i > 0, \\ 0, & S_i = 0, \end{cases} \tag{11}$$

$$\pi_{i,m} = H_{i,\,j_{i,m}} \big/ \textstyle\sum_{j \in \mathcal{T}_i} H_{i,j}, \quad m = 1, \ldots, M,\ S_i > 0, \tag{12}$$

$$\hat{H}_i(j) = H_{i,j}\, \mathbb{1}[j \in \mathcal{T}_i], \tag{13}$$

For $S_i > 0$, $\hat{\pi}_i(j) = \hat{H}_i(j)/\sum_{j'} \hat{H}_i(j')$; otherwise $\hat{\pi}_i(j) = 0$, as in Eq. (4). When $M \geq 2$ and $S_i > 0$, the gap $\pi_{i,1} - \pi_{i,2}$ is a heuristic measure of ambiguity among retained labels, not calibrated posterior uncertainty. After each voting step we zero entries outside $\mathcal{T}_i$ and refresh ℓ_i according to Eq. (11). Top-M pruning is a sparsity–fidelity heuristic, not a Bayesian operation: it breaks conjugacy on truncated coordinates, and a label leaving $\mathcal{T}_i$ loses its stored evidence. The per-frame injection bound remains valid when $\eta_i^{(t)}$ is computed from the truncated S_i, because that sum is non-negative. The bound concerns the injected vote, not state changes due to decay or pruning. M=3 covers the dominant multi-peaked cases (boundary Gaussians with two adjacent labels; over-segmentation transitions). The implementation uses a dense $N \times J$ tensor for scatter updates but retains at most M nonzero evidence entries per Gaussian; M controls evidence support rather than the allocated tensor size.

Visibility-aware decay. A uniform decay $\mathbf{H} \leftarrow \gamma \mathbf{H}$ would erode occluded primitives whose evidence cannot be refreshed. We therefore decay only sufficiently visible Gaussians:

$$H_{i,j} \leftarrow \begin{cases} \gamma \cdot H_{i,j} & W_i^{(t)} > w_{\min}, \\ H_{i,j} & \text{otherwise}, \end{cases} \quad \gamma \in (0,1),\ \gamma{=}0.95,\ w_{\min} > 0. \tag{14}$$

Hard-label rendering. Given discrete $\{\ell_i\}$ (no top-M distribution), the rendered label at pixel p is

$$\hat{\ell}(p) = \arg\max_{j > 0} \sum_{k=1}^{K} w_{p,k}\, \mathbb{1}[\ell_{g_{p,k}} = j], \tag{15}$$

Algorithm 1 DIGS Pipeline: Mapping and Encoding

Input: pre-reconstructed Gaussians $\mathcal{G}$, frame stream $\{(\text{cam}_t, \text{rgb}_t)\}_{t=1}^{T}$, SAM, CLIP, multi-modal LLM
Parameters: τ, γ, θ_{iou}, $w_{\min}$, M, θ_{miss}, J, θ_{match}, λ
Init: $\mathbf{H} \leftarrow \mathbf{0}_{N\times J}$; $\{\ell_i\} \leftarrow \mathbf{0}_N$; match, miss $\leftarrow \mathbf{0}$
// – Stage 1: Class-agnostic mapping –
for each frame $t = 1 \dots T$ **do**
  rgb_t, topk_ids, topk_w $\leftarrow$ TopKRender($\mathcal{G}$, cam_t) ▷ Eq. (1)
  $\hat{L}^{(t)} \leftarrow$ LateArgmaxRender(topk_ids, topk_w, $\mathbf{H}$) ▷ Eq. (6)
  $\mathcal{S}^{(t)} \leftarrow$ SAM(rgb_t)
  $C \leftarrow$ BatchIoU($\mathcal{S}^{(t)}$, $\hat{L}^{(t)}$); $\sigma^* \leftarrow$ Hungarian($1 - C$)
  $\phi \leftarrow$ AssignGlobalIDs(σ^*, C, θ_{iou}); $\mathcal{L}^{(t)} \leftarrow$ Remap($\mathcal{S}^{(t)}$, ϕ)
  Update match, miss counters from σ^*
  $W^{(t)} \leftarrow$ `scatter_add`(topk_w, topk_ids)
  $\mathbf{H}[\,W^{(t)} > w_{\min}\,] \mathrel{*}= \gamma$ ▷ Eq. (14)
  $\eta \leftarrow \tau \,/\, (\mathbf{H}.\text{sum}(\dim{=}1) + W^{(t)} + \tau)$ ▷ Eq. (2)
  $\mathbf{H}$.`scatter_add_`(topk_ids, topk_w $\cdot\, \eta$[topk_ids], $\mathcal{L}^{(t)}$) ▷ Eq. (3)
  $\{\ell_i\} \leftarrow \arg\max(\mathbf{H}, \dim{=}1)$; $\mathbf{H} \leftarrow$ TopM_Prune($\mathbf{H}$, M)
  Recycle($\mathbf{H}$, miss, θ_{miss})
end for
// – Stage 2: Multi-modal instance encoding –
$\mathcal{I}_{\text{stable}} \leftarrow \{j : \text{match}_j \geq \theta_{\text{match}} \ \wedge \ \text{miss}_j \leq \theta_{\text{miss}}\}$
for each $j \in \mathcal{I}_{\text{stable}}$ **do**
  $\mathcal{V}_j \leftarrow$ SelectTopViews(j, train_cams, $\{\ell_i\}$)
  $\text{crops}_j \leftarrow$ BBoxCrop(IsoRender($\mathcal{G}_j$, $\mathcal{V}_j$))
  $\mathbf{e}_j \leftarrow \ell_2$-norm(Mean(CLIP_img($\text{crops}_j$))) ▷ Eq. (7)
  $d_j \leftarrow$ LLM(crops_j, attribute_prompt)
  $\mathbf{e}_j^{\text{txt}} \leftarrow$ CLIP_text(d_j)
  $\mathbf{e}_j^{\text{fused}} \leftarrow \lambda\, \mathbf{e}_j + (1-\lambda)\, \mathbf{e}_j^{\text{txt}}$ ▷ Eq. (8)
end for
return $\{\mathbf{e}_j^{\text{fused}}, \mathcal{G}_j\}_{j\in\mathcal{I}_{\text{stable}}}$, $\{\ell_i\}$

which Eq. (6) strictly generalises (Singleton support reduces to Eq. (15)).

B.3 Association and Lifecycle

Hungarian association. The IoU cost matrix $C_{i,j} = |\mathbf{\Omega}_i \cap R_j| / |\mathbf{\Omega}_i \cup R_j|$ between SAM masks $\{\mathbf{\Omega}_i\}$ and predicted regions $\{R_j\}$ is computed as a single batched tensor operation; the Hungarian algorithm solves $\sigma^* = \arg\min_\sigma \sum_i (1 - C_{i,\sigma(i)})$ with

$$\phi(i) = \begin{cases} \text{ID}(R_{\sigma^*(i)}) & \text{if } C_{i,\sigma^*(i)} \geq \theta_{\text{iou}}, \\ \text{NewID}() & \text{otherwise.} \end{cases} \tag{16}$$

A greedy fallback inherits the best overlapping global ID if $\max_j C_{i,j} \geq \theta_{\text{iou}}$, permitting many-to-one matching under occlusion-induced fragmentation. We use θ_{iou}=0.4.

Lifecycle and zombie recycling. Each instance traverses four states: *nascent* (just created), *active* (matched each frame), *missing* (unmatched within θ_{miss} window), and *zombie* ($\text{miss}_j > \theta_{\text{miss}}$). When the registry approaches capacity J, zombies are recycled first (longest-missing preferred), followed by all-zero columns; recycling zeroes the column and returns the ID to the pool, so $|\mathcal{I}| \leq J$ regardless of sequence length, unlike B^3-Seg [13] (single query object per run) or Gaussian Grouping [9] (single batched optimisation, no streaming).

B.4 Detection-Bridging Query

When the test-frame RGB is available, we localise the prompt $\mathbf{q}$ with GroundingDINO [40], obtain a 2D mask M_{det} via SAM with box prompting, and match to the active registry by IoU:

$$j^* = \arg\max_{j \in \mathcal{I}} \ \text{IoU}\big(M_{\text{det}}, \{\hat{L}_{\text{test}} = j\}\big), \quad \hat{M}_{\mathbf{q}} = \mathbb{1}[\hat{L}_{\text{test}} = j^*], \tag{17}$$

with rejection threshold θ_{bridge} guarding against detector misfires or off-scene prompts. No retraining or embedding precomputation is required.

B.5 LLM Description Cache

On the first mapping run for a scene, the optional LLM-fusion stage issues exactly one GPT-4.1 vision-language API call per stable instance $j \in \mathcal{I}_{\text{stable}}$ with paired context / isolated view crops as input. The model returns a single line of text with three pipe separators (LABEL | CATEGORY | IDENTITY | DETAILS), which is written to a per-scene description cache keyed by integer instance ID. For all subsequent evaluation runs on the same scene we load this cache directly—no further API calls are issued—so that ablations, hyperparameter sweeps, and re-evaluations share identical per-instance descriptions and remain reproducible. At fusion time the cached line is CLIP-text-encoded into $\mathbf{e}_j^{\text{txt}}$ for use in Eq. (8). We release the description cache alongside the code so that all numerical results in the paper can be reproduced without new API calls.

Prompt template. The text passed to GPT-4.1 alongside the rendered crops is reproduced verbatim below.

```
You are given rendered images of ONE 3D instance extracted from a
reconstructed scene. The images come in pairs from multiple viewpoints:
- Odd-numbered images (1st, 3rd, 5th, ...): scene crops centered on the
  target object, rendered from the full 3D reconstruction. Other objects
  may appear at the edges.
- Even-numbered images (2nd, 4th, 6th, ...): the same viewpoints but
  with ONLY the target object rendered on a black background. This
  shows the exact boundary and shape of the target.

Use the scene crops to understand context, recognize characters or
brands. Use the black-background images to determine the object's true
shape, color, and boundary. Describe ONLY the target object visible in
the black-background images.

Output exactly ONE line with three pipe separators:

LABEL | CATEGORY | IDENTITY | DETAILS

LABEL -- one of:
- OBJECT:     a complete, well-formed item visible across all views.
- FRAGMENT:   partial, blurry, cut-off, or too distorted to name
              confidently.
- BACKGROUND: wall, floor, table, pillar, or other large flat surface.
- AMBIGUOUS:  multiple unrelated objects merged, or unrecognizable.

CATEGORY -- 1-2 bare nouns, no color, no adjectives
            (e.g. duck, chair, bag, camera).

IDENTITY -- exactly 3-4 words: [color] [object name] or
            [color] [proper name] [object type]. Use the simplest, most
            common noun. If you recognize a specific character or brand,
            include its name. Do NOT add extra modifiers, materials, or
            adjectives beyond color and name.

DETAILS  -- one sentence, 10-20 words. Only factual visual features:
            shape, markings, text, accessories. No guessing, no
            function, no subjective words.

Calibration examples (NOT from the actual scene):
OBJECT    | plush toy   | orange tabby plush toy
```

```
                        | Green glass eyes, stitched nose, red bow on neck.
OBJECT     | teapot     | blue ceramic teapot
                        | White floral pattern, curved bamboo handle, round lid.
FRAGMENT   | edge       | brown wooden edge
                        | Only one corner visible, rest cut off by frame.
BACKGROUND| surface     | white countertop surface
                        | Smooth matte finish with light scratches near edge.
AMBIGUOUS | shapes      | mixed colorful shapes
                        | Multiple objects merged with heavy blur, no clear boundary.

Rules:
1. IDENTITY must be exactly 3-4 words. No more. Start with color, end
   with object name.
2. Use the simplest, most common everyday noun. Prefer short familiar
   words over technical or formal synonyms.
3. Only describe the object in the black-background images. Ignore
   everything else in scene crops.
4. When uncertain, choose FRAGMENT over OBJECT.
5. BACKGROUND is only for flat structural surfaces with no distinct 3D
   shape. If it has depth, edges, or a recognizable form, use OBJECT.
6. One line only. No line breaks, no markdown, no JSON, no quotes.
   Exactly three pipes.
```

C Replica: Per-Scene Results and Mapping Scale

Tables 8 and 9 expand the two protocols summarised in Table 5. Under Gaga's protocol, *raw* counts active instances at the end of mapping and *stable* counts those retained by the stability filter. Under ObjectGS's protocol, every scene uses 1,750 mapping frames and 250 held-out test frames; mapping time is measured on one NVIDIA A100.

Table 8: Replica under Gaga's instance protocol (segmentation metrics in %).

Scene	Mapped frames	Raw	Stable	IoU	Precision	Recall
office0	180	224	92	42.85	31.25	50.00
office1	133	178	48	55.05	52.94	61.36
office2	164	268	98	52.84	42.55	57.97
office3	180	313	112	48.94	38.94	53.66
office4	161	427	79	53.94	35.79	60.71
room0	180	386	98	44.92	43.43	46.24
room1	180	210	78	60.12	48.15	72.22
room2	180	243	77	52.44	46.75	58.06
Mean	170	281	85	**51.39**	**42.48**	**57.53**

Table 9: Replica under ObjectGS's panoptic-lifting protocol (segmentation metrics in %).

Scene	Instances	Mapping time (min)	IoU	Dice	Accuracy
office0	34	5.7	92.84	96.05	98.87
office1	26	5.6	95.40	97.50	99.29
office2	29	5.9	90.01	94.17	98.64
office3	58	6.4	88.97	93.80	98.23
office4	20	6.1	92.76	95.86	99.15
room0	42	7.3	88.79	93.54	97.53
room1	38	7.4	91.95	95.33	99.09
room2	34	6.2	90.92	94.97	98.63
Mean	35	**6.3**	**91.46**	**95.15**	**98.68**

The longer ObjectGS streams take 5.6–7.4 minutes per scene, averaging 6.3 minutes (about 215 ms per mapped frame).

D Hyperparameter Sensitivity and Ablations

We probe the three deployed hyperparameters—LLM/CLIP fusion weight λ (§3.6), top-M support size (§3.4), and Dirichlet prior strength τ (§3.3)—on LERF-OVS at the CLIP-best configuration (Stability Filter on, Distribution-Preserving on; row (c) of Table 6), holding all other hyperparameters fixed. λ=1 recovers CLIP-only retrieval; M=1 collapses Late-Argmax to hard-label rendering (Eq. (15)). Smaller τ tightens the injection bound (Proposition 1); larger τ approaches additive accumulation.

D.1 LLM/CLIP Fusion Weight λ

λ sensitivity. Table 10 sweeps $\lambda \in \{0.0, 0.2, 0.3, 0.5, 0.7, 0.8, 1.0\}$ on LERF-OVS with retrieval thresholds disabled. Both extremes degrade—λ=0 (text-only) drops to 49.16 because the LLM-text channel alone cannot localise instance footprints, λ=1 (CLIP-only) reaches only 61.12—and the global default λ=0.7 peaks at mean mIoU 64.34. Per-scene optima diverge: Figurines prefers the CLIP-heavy end ($\lambda \geq 0.8$: 80.01), Ramen prefers balanced fusion (λ=0.5: 60.40), and Teatime saturates at the default (77.65).

Table 10: λ sweep on LERF-OVS, embedding-fusion path with retrieval thresholds disabled (mIoU, %).

λ	Figurines	Ramen	Teatime	Waldo_kitchen	Mean
0.0	67.61	43.23	48.66	37.12	49.16
0.2	75.97	43.25	52.58	37.30	52.28
0.3	75.97	51.69	52.58	37.30	54.39
0.5	75.97	60.40	60.26	41.02	59.41
0.7	77.60	58.38	77.65	43.75	64.34
0.8	80.01	50.30	77.65	43.75	62.93
1.0	80.01	47.94	71.35	45.19	61.12

Retrieval thresholds. Table 11 sweeps the top-1 / top-2 margin filter $\Delta_{\min}$ at fixed absolute fused-cosine floor σ_{abs}=0.25 and λ=0.7. $\Delta_{\min}$=0 recovers the unfiltered baseline at 64.34 mean mIoU; raising $\Delta_{\min}$ monotonically improves the mean up to the deployed $\Delta_{\min}$=0.12 (71.95, +7.61 over no filter), beyond which over-aggressive filtering rejects valid matches and the mean drops to 60.42 at $\Delta_{\min}$=0.20. The deployed setting concentrates its gain on the noisy scenes Ramen (+6.30) and Waldo_kitchen (+**23.01**); Figurines (+1.11) and Teatime (+0.00) are unaffected. The per-scene split tracks scene clutter rather than scene size: Waldo_kitchen contains dozens of visually near-identical fixtures (white cabinets, similarly-shaped jars and utensils) that produce dense top-k similarity clusters where the top-1 wins by a near-tie, so $\Delta_{\min}$ rejects most such fragile picks; Ramen exhibits a milder version of the same effect among food items in one bowl. Figurines and Teatime, in contrast, contain spatially separated objects with distinctive geometry and colour, producing wide top-1 margins that the filter rarely activates on.

Table 11: $\Delta_{\min}$ sweep at σ_{abs}=0.25 on LERF-OVS at λ=0.7 (mIoU, %). $\Delta_{\min}$=0 is equivalent to disabling the filter pair. Deployed $\Delta_{\min}$=0.12.

$\Delta_{\min}$	Figurines	Ramen	Teatime	Waldo_kitchen	Mean
0.00	77.60	58.38	77.65	43.75	64.34
0.05	80.58	60.81	77.65	48.33	66.84
0.10	78.71	64.68	77.65	57.85	69.72
0.12	78.71	64.68	77.65	66.76	71.95
0.15	78.71	57.94	74.41	57.43	67.12
0.17	75.96	55.17	68.72	56.02	63.97
0.20	73.13	50.21	62.40	55.94	60.42

Decomposition. The same thresholds account for the gaps between Table 10 (no thresholds) and the deployed Table 6: 64.34 → 71.95 at λ=0.7 recovers row (d) exactly, and the −5.73 gap between the λ=1.0 row (61.12) and the deployed CLIP-only row (c) (66.85) reflects the same filtering applied at λ=1. The threshold values are calibrated once and held fixed across λ.

D.2 Top-M Evidence Support

Table 12 sweeps $M \in \{1, 2, 3, 4, 5\}$ at the CLIP-best configuration (row (c) of Table 6). M=1 collapses Late-Argmax to hard-label rendering and incurs the largest drop; mean mIoU peaks at the deployed default M=3 (66.85), +5.46 over M=1 (61.39), and degrades beyond (M=4: 64.07, M=5: 64.07) as the truncated support sparsifies retained mass over labels with little genuine evidence.

Table 12: Top-M sweep on LERF-OVS (mIoU / mAcc, %).

M	Figurines		Ramen		Teatime		Waldo_kitchen		Mean	
	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc
1	65.32	85.71	55.25	71.83	74.81	86.44	50.18	68.18	61.39	78.04
2	70.66	87.50	54.78	73.24	71.10	84.75	53.37	68.18	62.48	78.42
3	74.81	91.07	56.15	73.24	74.70	84.75	61.72	81.82	66.85	82.72
4	74.78	91.07	51.33	60.56	75.55	84.75	54.62	72.73	64.07	77.28
5	74.76	91.07	51.58	71.83	75.48	84.75	54.46	72.73	64.07	80.09

D.3 Dirichlet Prior Strength τ

Table 13 sweeps $\tau \in \{1, 2, 5, 10, 20, 50\}$ at the CLIP-best configuration. Both extremes degrade: small τ over-tempers the injection (τ=1 drops mean mIoU to 60.41), while large τ weakens the per-frame bound and approaches additive accumulation (τ=20: 61.15). The deployed default τ=5 achieves the best mean mIoU 66.85, with the sharpest gain on the adversarial scenes Ramen (+10.74 over τ=2) and Waldo_kitchen (+11.62 over τ=2), matching the qualitative analysis of §4.3.

Table 13: τ sweep on LERF-OVS (mIoU / mAcc, %).

τ	Figurines		Ramen		Teatime		Waldo_kitchen		Mean	
	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc
1	73.59	92.86	38.89	53.52	74.96	84.75	54.20	72.73	60.41	75.96
2	74.81	91.07	45.41	61.97	74.97	84.75	50.10	68.18	61.32	76.49
5	74.81	91.07	56.15	73.24	74.70	84.75	61.72	81.82	66.85	82.72
10	74.80	91.07	51.60	66.20	74.55	84.75	53.47	72.73	63.61	78.69
20	71.48	87.50	52.03	70.42	71.97	84.75	49.13	68.18	61.15	77.71
50	74.70	91.07	49.95	66.20	76.56	86.44	56.20	77.27	64.35	80.25

D.4 Stability Filter: Per-scene Proposal Counts

Table 14 isolates the Stability Filter (SF) effect on LERF-OVS at the *deployed* CLIP-best configuration (DP on, LLM off; row (c) of Table 6 with the filter toggled). The filter discards 92–97% of unstable raw instance proposals per scene, converting noisy SAM mask fragmentation into a compact, high-reuse set of stable instances.

Table 14: Stability Filter counts on LERF-OVS. *raw* = total instance proposals discovered after mapping; *pass* = instances meeting the strict criterion $M_i > 50 \wedge m_i < 30$; *filt* = instances zeroed out.

Scene	raw	pass	filt	mIoU off	mIoU on	Δ
Figurines	511	31	443	74.00	74.81	+0.81
Ramen	255	20	161	44.21	56.15	+11.95
Teatime	511	15	461	58.20	74.70	+16.50
Waldo_kitchen	1023	67	709	54.90	61.72	+6.82
Mean	575	33	444	57.83	66.85	+9.02

D.5 Tempered Dirichlet: Per-class ΔIoU

Table 15 lists per-class ΔIoU (tempered η − naive η) on the two adversarial scenes; listed classes are those with $|\Delta| \geq 10$. Dramatic gains concentrate on queries that require cross-view identity

stabilisation (*wavy noodles*, *red cup*, *kamaboko*), while classes with already-distinct appearance sit near ceiling under both configurations and show negligible change.

Table 15: Per-class ΔIoU (tempered η − naive η) on the two adversarial scenes.

Class	Naive η	Tempered η	ΔIoU
Ramen			
wavy noodles	0.00	85.62	+85.62
kamaboko	35.27	82.30	+47.02
bowl	38.27	56.22	+17.95
napkin	30.59	41.21	+10.62
corn	39.95	26.45	−13.49
Waldo_kitchen			
red cup	0.00	85.05	+85.05
yellow desk	51.54	87.90	+36.37
spoon	0.00	28.46	+28.46
plate	26.01	48.26	+22.25
ottolenghi	78.21	96.88	+18.67
ketchup	15.81	27.11	+11.30
cabinet	97.41	81.31	−16.10

E Qualitative Results and Failure Cases

E.1 3D-OVS: Multi-query Coverage across Five Scenes

Figure 6 visualises DIGS's predictions on all five 3D-OVS scenes under the detection-bridging query mode (§3.6). Each frame carries multiple free-form prompts (e.g. *Lawn*: bottle, ball, headphones, apron) and the predicted masks align tightly with GT boundaries even on thin or textured objects, consistent with the 96.45 mean mIoU in Table 4; no per-scene gradient training is performed.

E.2 Tempered Dirichlet η: Qualitative Comparison

Figure 7 visualises one representative prompt per scene across all four LERF-OVS scenes, comparing rendered Late-Argmax label maps under naive (η=1) and tempered $\eta_i^{(t)}$ (Eq. (2)). Tempered η produces visibly cleaner instance footprints in every panel, and the resulting spatial-retrieval mask aligns markedly more tightly with the ground truth.

E.3 Failure Cases

Despite leading mean mIoU on all three benchmarks, DIGS exhibits two residual failure modes that trace to upstream rendering and segmentation limits; representative examples are in Figures 8 and 9.

F1. Transparency couples voting weight to background. Our per-Gaussian voting weight $W_i^{(t)}$ is the alpha-compositing weight from rendering, so transparent and translucent objects (glass / plastic surfaces) propagate non-trivial weight through the foreground surface to background Gaussians at the same pixel. The closed-loop association then assigns the foreground SAM mask partly to background Gaussians, "smearing" the instance footprint past the transparent surface and inflating its 3D extent. This is visible on the *transparent box* in Figurines and the *transparent cup* in Ramen.

F2. SAM-granularity caps part-level prompts. SAM's automatic-mask generator commits to a single granularity per scene; when a prompt names a sub-region of a larger SAM mask, our reconstructed semantic map has no slot for the part and retrieval returns the whole parent instance. Affected queries on LERF-OVS include *waldo* (a printed character on the cabinet), *pirate hat* (part of a figurine), and *hooves* together with *dall-e brand* (a body region and a printed logo on the same figurine); in each case the relevant SAM mask covers the whole parent, capping the achievable IoU on the part-level query. This reflects the SAM-granularity ceiling discussed in §5 (*Limitations and outlook*).

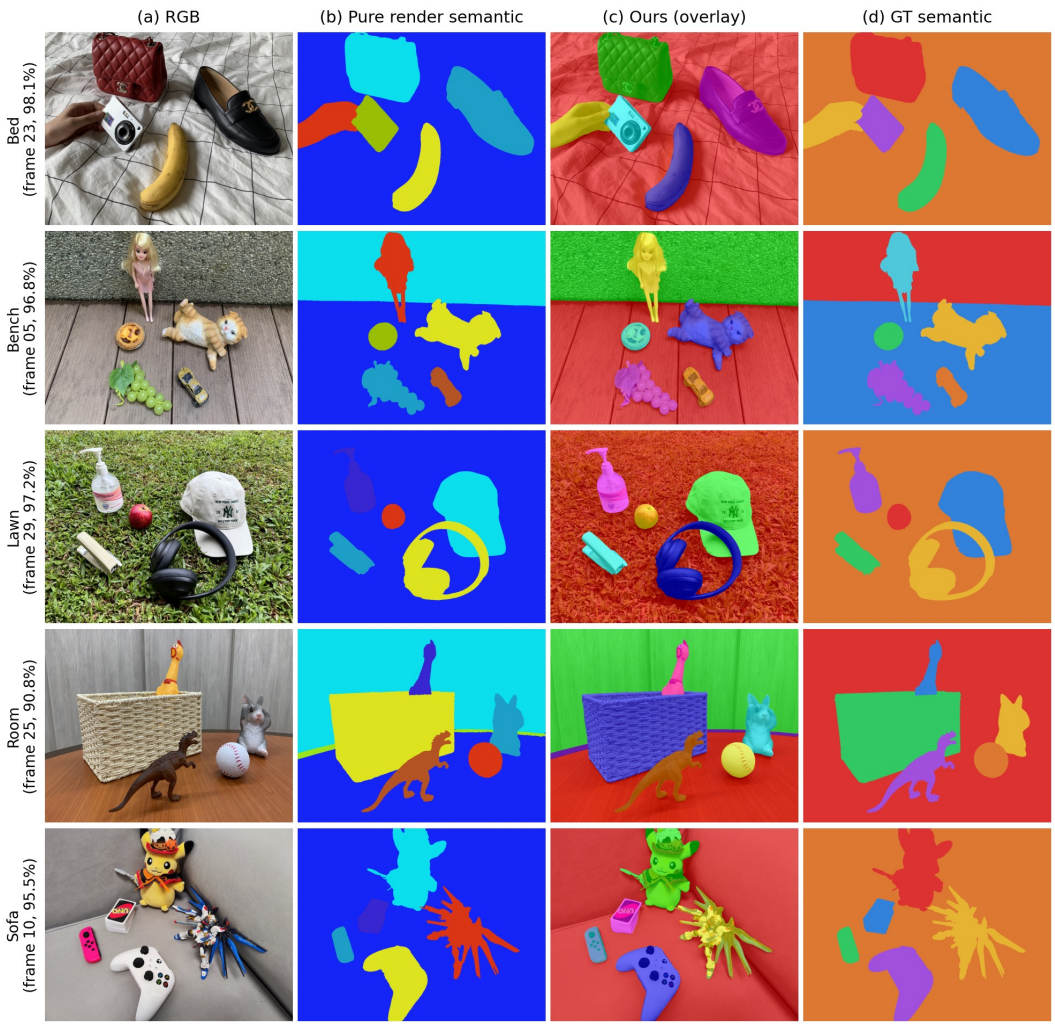


Figure 6: 3D-OVS qualitative results across all five scenes.

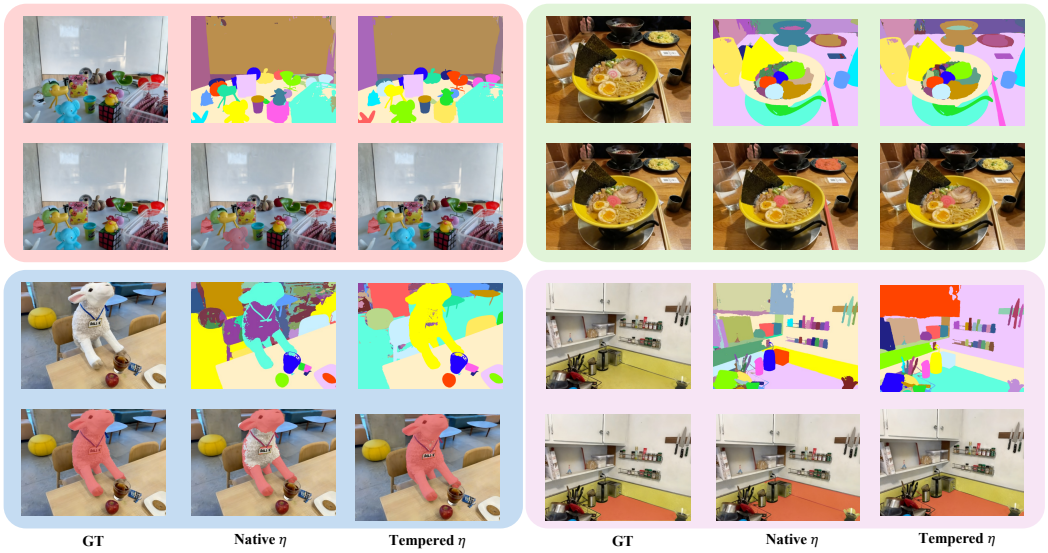


Figure 7: Naive η versus tempered $\eta_i^{(t)}$ across all four LERF-OVS scenes. One representative prompt per scene. Tempered η produces visibly cleaner instance footprints in every panel.



Figure 8: F1: Transparency couples voting weight to background—*transparent box* (Figurines) and *transparent cup* (Ramen).

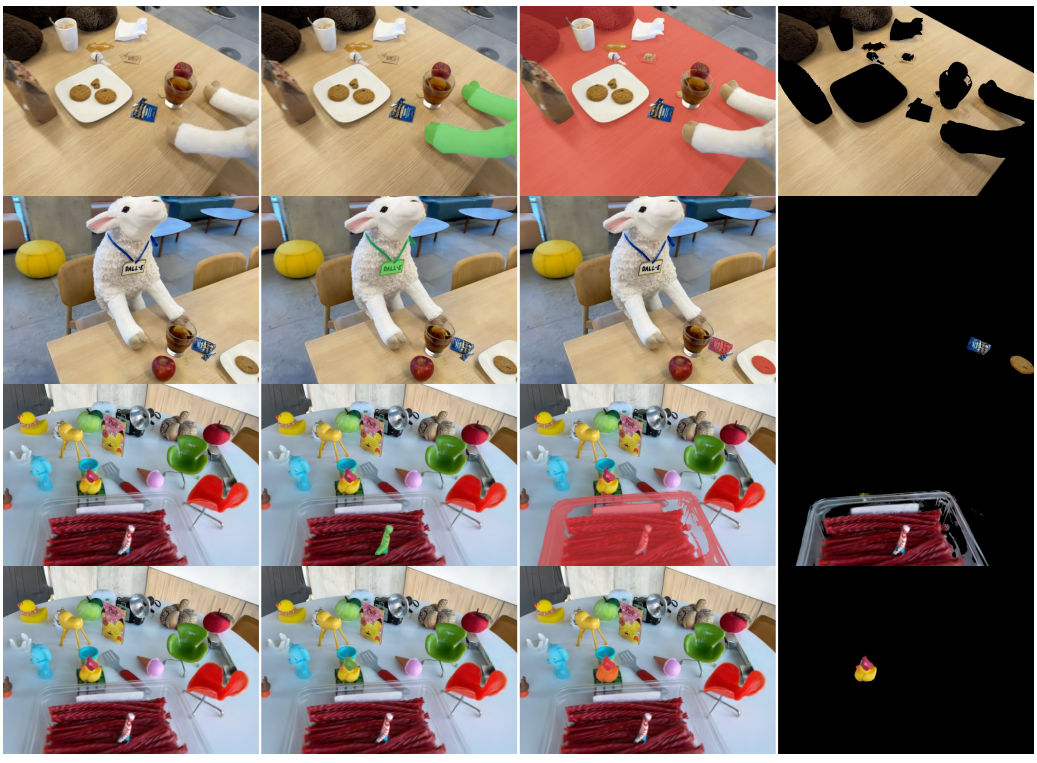


Figure 9: F2: SAM-granularity caps part-level prompts—*waldo*, *pirate hat*, and *hooves / dall-e brand* (paired on the same figurine) all retrieve the whole parent instance.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and Section 1 (Introduction) accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discussed the limitations in the Conclusion and the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The theoretical results are stated in Section 3.3 (Method) with full assumptions and proofs in the Appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have included the pipeline algorithm in the Appendix and released the code.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We have released the source codes.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We have included it in the Experimental Setup.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: It is not standard in this field.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have included it in the Experimental Setup.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research conducted in the paper conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discussed the broader impacts in the Conclusion.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have properly cited the original papers at first use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `https://paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects, so no IRB approval is required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We used an LLM in our experiments; the Appendix documents the prompt template and the LLM model used, and we release the LLM-generated data.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.