ISSN (O) 2278-1021, ISSN (P) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511150

Splatsearch: Real Time Semantic Digital Twin

Madhushree M[1], Arfain Sultana[2], Chandana N[3], H Tilak[4], Kruthika S[5]

Assistant Professor, Department of Computer Science, East West College of Engineering, Bengaluru, India[1]

Student, Department of Computer Science, East West College of Engineering, Bengaluru, India[2]

Student, Department of Computer Science, East West College of Engineering, Bengaluru, India[3]

Student, Department of Computer Science, East West College of Engineering, Bengaluru, India[4]

Student, Department of Computer Science, East West College of Engineering, Bengaluru, India[5]

Abstract: To meet the demand for high-fidelity spatial awareness in robotics and augmented reality (AR), this project introduces a unified framework for generating real-time, linguistically-aware digital twins. Traditional Simultaneous Localization and Mapping (SLAM) systems typically suffer from "semantic blindness"—they capture 3D geometry but lack object context. Our system bridges the gap between spatial data and human language by integrating SplaTAM for dense Gaussian Splatting, LangSplat for open-vocabulary embedding, and Online LangSplat for high-speed encoding.
The architecture leverages a robust client-server model where an iPhone streams synchronized RGB-D data directly to a GPU-accelerated backend. This setup enables real-time 3D reconstruction at over 45 frames per second. During this generation phase, the system dynamically assigns CLIP-based language features to millions of 3D Gaussians.
The resulting digital twin is not only photorealistic but fully searchable via natural language queries. Users can seamlessly localize and identify specific objects within previously unknown environments with remarkable semantic accuracy. By eliminating the latency gap between mapping and understanding, this research delivers a scalable, highly efficient solution perfectly suited for advanced smart-home automation and assistive navigation technologies.

Keywords: 3D Gaussian Splatting, Open-vocabulary labelling, Object search, COLMAP, Neural rendering, Network-free browser viewer.

I. INTRODUCTION

Photos and videos record a place from one side at a time. A radiance field records it as a scene that can be looked at from new viewpoints. 3D Gaussian Splatting made that practical at interactive rates: a scene is a set of coloured ellipsoids that a GPU can sort and draw in real time [1]. What those viewers still do not answer is a simple question. Where is the bottle? Which splats are the oven gloves?

Bounding boxes and a separate detector at query time are a weak fit. A thin object such as keys, a remote, or a bottle occupies few pixels, and a box does not say which Gaussian belongs to it. A language field such as a CLIP embedding on each point can rank regions [8], [9], but the user then sees a heatmap rather than the object's own colour.
SplatSearch takes a different route. Each Gaussian carries a discrete object id. The id is painted on from 2D masks, so search, hover, and fly-to all read the same label. The viewer runs in the browser on an ordinary laptop. Training of a new scene runs on a free Colab T4, not on the user's machine.

The aim is a usable scene-search tool: reconstruct or load a Gaussian scene, attach object names, and let a person find those objects by typing or hovering. Real-time packet-style capture of a live room, and a formal detection benchmark, are left as later work.
The contributions are:

1. A pipeline from a phone walkaround, or from a pretrained 3D Gaussian scene, to a set of Gaussians that each have an object label.
2. A visibility-aware lift from Grounding DINO and SAM masks onto those Gaussians, followed by k-nearest-neighbour smoothing.
3. A browser viewer that searches by name, keeps hits in their true colour, dims the rest, and flies the camera to the match.
4. Hover picking that returns the nearest labelled splat under the cursor, including thin objects, and that hides background classes and classes with too few splats.
5. Two phone scenes trained on a Colab T4, plus four public scenes labelled with the same method, with the measured train time and PSNR reported only where they were recorded.

 This work is licensed under a Creative Commons Attribution 4.0 International License

ISSN (O) 2278-1021, ISSN (P) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511150

II. RELATED WORK

Asking where an object is in a reconstructed scene sits between three lines of work: novel-view synthesis, open-vocabulary vision models, and methods that attach those models to a 3D representation.

1) From radiance fields to 3D Gaussians

Mildenhall et al. represent a scene as a continuous function of position and direction and render a new view by sampling rays [12]. Quality is high. A search interface that must redraw the scene on every camera move is a poor fit, because each ray is expensive. Müller et al. make the same representation fast enough to train in minutes with a hash grid, but a view is still integrated along rays rather than drawn as a set of primitives [13].3D Gaussian Splatting replaces the field with an explicit set of anisotropic Gaussians, optimised by gradient descent and rasterised by sorting [1]. That is the representation used here. The four public scenes are the published reconstructions of the Mip-NeRF 360 captures, trained for 30,000 iterations [1], [2]. New phone videos are not trained that way. Their cameras come from COLMAP [3], and the Gaussians are optimised with the Markov-chain Monte Carlo trainer in gsplat for 12,000 steps [6], [7]. These papers solve view synthesis. None of them stores an object name.

2) Open-vocabulary models in 2D

The names have to come from somewhere. CLIP trains an image encoder and a text encoder so that a phrase and a picture of the same thing land near each other [14]. LSeg uses that idea for dense 2D segmentation, assigning a language feature to every pixel [15]. CLIP and LSeg know words. They do not know the 3D point those words belong to, and a label predicted in one photo need not agree with the next photo of the same object. Two later models are used as the 2D front end of this work. Grounding DINO takes a text prompt and returns boxes [5]. The Segment Anything Model turns a box, or a point, into a mask, including for objects it was not specifically trained to name [4]. Both are image models. A mask is a claim about pixels in one view. It is not yet a claim about a Gaussian.

3) Language fields

The first 3D search methods distil CLIP into a radiance field. LERF optimises a language feature along rays so that a query such as “a mug” highlights a region in a new view [8]. Distilled feature fields use the same idea for robotics, where the field is queried to find an object a robot should grasp [16]. The answer is a score along a ray, not a primitive that can be kept in its own colour. GARField is closer to grouping than to naming. It trains a scale-conditioned affinity field from SAM masks, so the same 3D point can belong to a small part or to the whole object depending on the scale of the query [17]. That handles a real ambiguity, for example whether a wheel is its own object. It still lives in a radiance field, and the group does not carry a fixed name. Once Gaussian splatting was fast enough to be the scene itself, the same distillation moved onto Gaussians. LangSplat compresses CLIP features into a scene-specific space and renders a language map at interactive rates [9]. LEGaussians quantises language features onto Gaussians so an open-vocabulary query can be answered without storing a full CLIP vector on every primitive [18]. Feature 3DGS distils several foundation features, not only CLIP, into the same splats [10]. All three can rank regions for a phrase that was never in a fixed list. The rendered result is still a relevancy or feature image. The Gaussian’s own colour is not the answer, and two nearby objects with similar CLIP vectors are easy to confuse.

4) An identity on each Gaussian

A second family skips the continuous language vector and stores a discrete identity. Gaussian Grouping adds a short identity code to each Gaussian and supervises it by rendering into SAM masks, with a spatial regulariser so the codes stay consistent in 3D [11]. The goal is to segment and then edit: remove an object, in paint the hole, or recolour a group. SAGA learns an affinity feature so that a click or a text prompt can segment any group of Gaussians after training, again using SAM [19]. Open Gaussian pushes this to point level for open-vocabulary understanding: each Gaussian is associated with a language feature, and the scene can be queried without rendering a 2D map first [20].These are the nearest methods. They show that a 2D mask can be lifted onto Gaussians, and that a discrete or point-level code is more editable than a heatmap. They are not built to answer a typed name inside a viewer that leaves the matching splats in their original colour. Gaussian Grouping and SAGA optimise the identity during reconstruction. This work labels Gaussians that have already been trained, including published checkpoints that are not retrained, and it collapses the identity to one class id rather than a code that must be clustered at query time.

5) What this paper takes from that work

Table I places the method against the closest of these systems. Novel-view papers supply the scene [1], [2], [6]. CLIP-style fields supply the idea that a person should be able to ask for an object in words [8], [9], [18]. Mask-lifting papers supply the idea that the answer should be a set of Gaussians, not a heat blob [11], [17], [19], [20]. The step taken here is

 This work is licensed under a Creative Commons Attribution 4.0 International License

ISSN (O) 2278-1021, ISSN (P) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511150

narrower. A fixed list of names is masked with Grounding DINO and SAM, a visibility vote writes one id onto each existing Gaussian, and search reads that id. There is no second feature field, and there is no editing stage.

TABLE I: COMPARISON OF SCENE SEARCH APPROACHES

Reference	Representation	What a query returns	How identity is obtained	Limitation
Kerbl et al. [1]	3D Gaussians	A new viewpoint only	Not represented	No object identity
LERF [8]	Radiance field + CLIP	A language score along rays	Distilled CLIP features	Not a real-time splat; the result is a heatmap
GARField [17]	Radiance field + affinity	A group at a chosen scale	SAM masks, contrastive grouping	Groups have no fixed name; not a Gaussian scene
Gaussian Grouping [11] LangSplat[9], LEGaussians [18]	Gaussians + identity code Gaussians + language features	A segment that can be edited A rendered relevancy map	SAM masks supervised through rendering Distilled CLIP, often with SAM masks	Trained for editing; identity is a code, not a named class Open vocabulary, but the answer is not the object's own colour
OpenGaussian [20]	Gaussians + point language features	An open-vocabulary point set	SAM masks and CLIP on Gaussians	Built for benchmark segmentation, not a fixed-name search of a finished scene
This work	One discrete class id per Gaussian	The named splats, in their own colour	Grounding DINO + SAM, visibility vote on a frozen scene	Cannot name an object the mask stage never saw

This comparison provides the motivation for storing a discrete class id on each Gaussian, and for treating masks and a finished reconstruction as part of the method rather than as a separate detector. The review shows that current work covers novel-view synthesis, language fields, and open-vocabulary grouping. Some papers address radiance fields, some address 3D Gaussians, and some address editing or benchmark segmentation. What a query returns depends on that choice. A relevancy map or a segmentation score does not, by itself, leave the named object in its own colour [9], [18], [20]. The suggested system takes that requirement as the target. It labels two phone captures and four public Mip-NeRF 360 scenes with the same visibility vote, reports reconstruction quality and label coverage, and returns the named splats in the viewer. A detection-accuracy benchmark, names for objects the mask stage never saw, and capture in real time remain future work.

III.PROPOSED SYSTEM

The system has two paths that meet in one viewer. A new scene is a walkaround video or a photo set. Frames are extracted, COLMAP estimates a camera for each registered frame, and gsplat MCMC trains a 3D Gaussian model on a free Colab T4. Grounding DINO and SAM then mask the named objects in a set of posed views. A visibility-aware vote, with k-nearest-neighbour smoothing, writes one class id onto each Gaussian. The output is an .splat file and a label file.
A public scene skips training. The Gaussians are the published INRIA reconstruction of a Mip-NeRF 360 capture [1], [2]. Only the labelling stage is run.

The viewer does not retrain. It draws a short prefix of the Gaussians first, then the full set. Search, hover, and fly-to read the class id already stored on each Gaussian.

Phone video or pretrained scene → posed views → 3D Gaussians → 2D object masks → per-splat label → browser search.

 This work is licensed under a Creative Commons Attribution 4.0 International License

ISSN (O) 2278-1021, ISSN (P) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511150

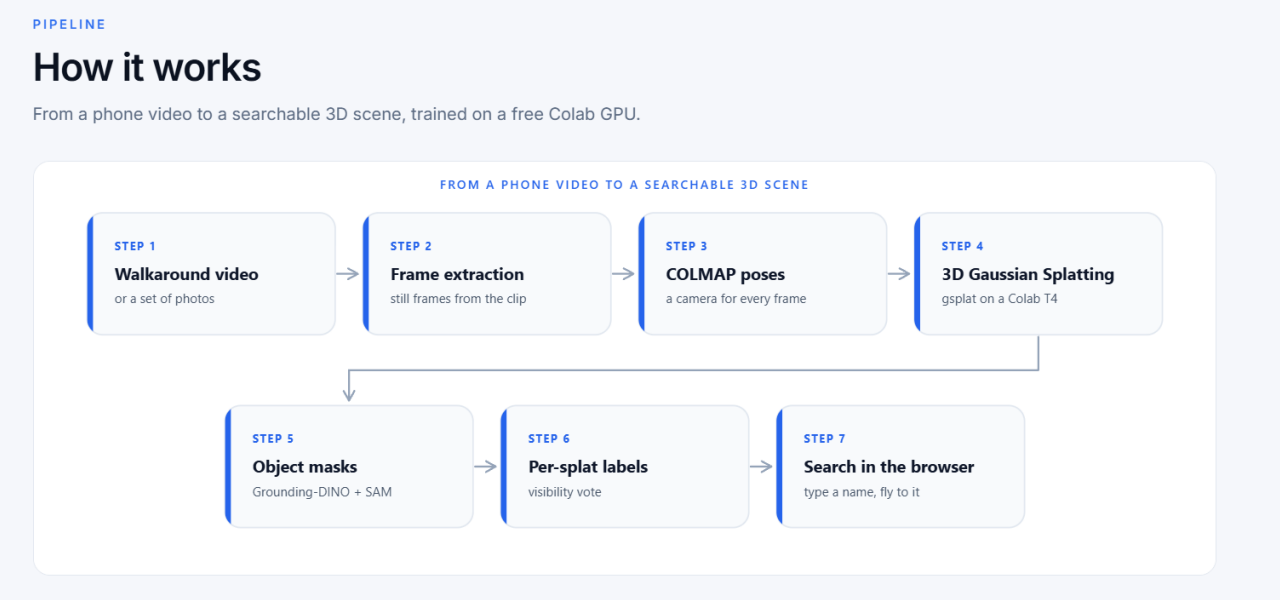


Fig. 1. From a phone video to a searchable scene: frame extraction, COLMAP, gsplat on a Colab T4, Grounding DINO and SAM, a visibility vote, and search in the browser.

A. System architecture

The browser holds the renderer, the camera, the label index, and the search state. WebGL2 draws the Gaussians. A worker sorts them by depth. Search does not call a server.

A new capture is the only exception. The user pastes the URL of a running Colab notebook and uploads a video or photos, with the scene name, the object list, and a quality flag. The notebook returns the splat and the labels. The page stores the job id locally, so a reload does not upload the video again.

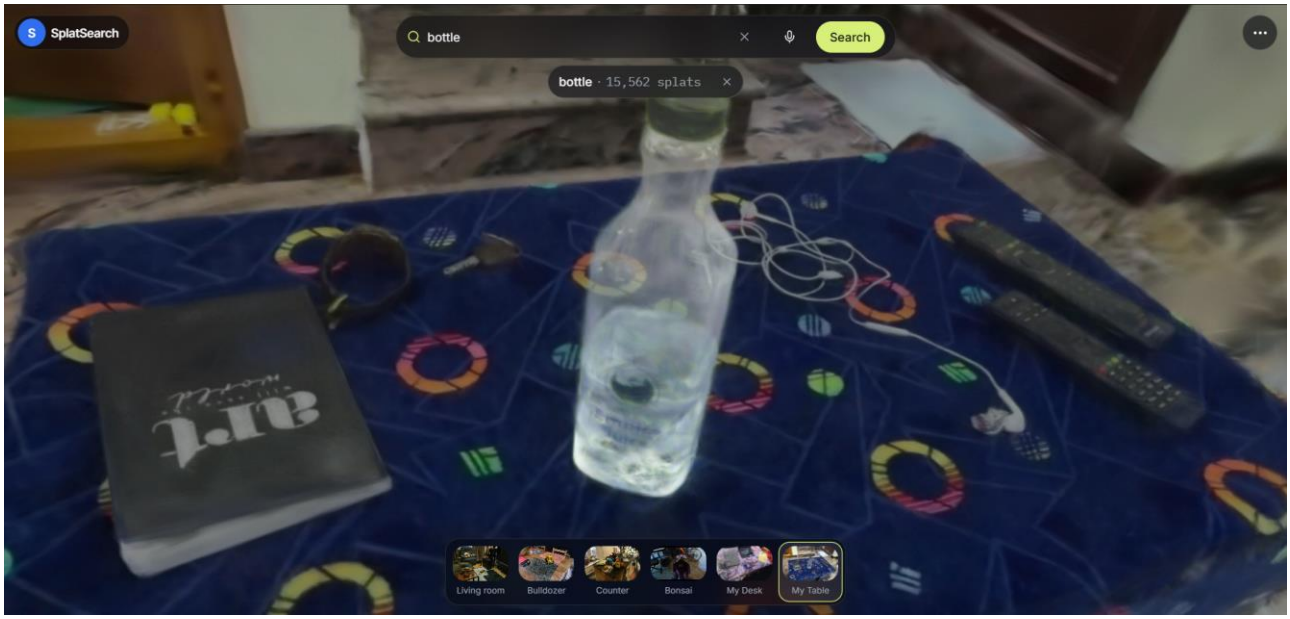


Fig. 2. Search on My Table. The query “bottle” keeps that object in colour, dims the rest, and moves the camera.

IV. METHODOLOGY

A. Scenes

Six scenes are used. Two were filmed on a phone for this project. Four are the Mip-NeRF 360 “room”, “kitchen”, “counter”, and “bonsai” captures [2], reconstructed by the published 3D Gaussian Splatting models at iteration 30,000 [1].

 This work is licensed under a Creative Commons Attribution 4.0 International License

ISSN (O) 2278-1021, ISSN (P) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511150

TABLE II: SCENES USED FOR RECONSTRUCTION AND LABELLING

Scene	Origin	Frames or images	Gaussians (full)	First paint	Label views	Object classes with splats
My Table	Phone video, this project	150 of 150 placed by COLMAP	366,117	250,000	60	6
My Desk	Phone video, this project	118 of 150 placed by COLMAP	353,436	250,000	60	12
Living Room	Mip-NeRF 360 “room”	311	780,000	150,000	52	20
Lego Bulldozer Table	Mip-NeRF 360 “lego”	279	780,000	250,000	70	11
Kitchen Counter	Mip-NeRF 360 “kitchen”	240	780,000	250,000	60	19
Bonsai Studio	Mip-NeRF 360 “bonsai"	292	780,000	250,000	72	16

The first-paint file is a prefix of the full file. Gaussians are stored coarse-to-fine, so the prefix is a complete but softer scene, not a crop. The living-room prefix is 150,000 Gaussians. The others use 250,000. The full file replaces it when it has loaded. Scale is not boosted.

My Table’s named objects are table, bottle, remote control, earphones, notebook, and keys. My Desk’s viewer chips are laptop, keyboard, notebook, desk, and electronics. The four public scenes expose the classes that actually received splats, for example sofa, piano, and teddy bear in the living room, the Lego bulldozer and oven gloves on the kitchen table, egg carton and bread on the counter, and the bonsai, piano keyboard, and bicycle in the studio.

B. Reconstruction of a phone scene

A clip of about one to two minutes is sampled into frames. COLMAP registers as many of those frames as it can [3]. My Table registered every extracted frame (150 of 150). My Desk registered 118 of 150. Failed frames are dropped, not invented.

The registered views train a 3D Gaussian model with gsplat’s MCMC trainer for 12,000 steps at Standard quality on a Colab T4 [6], [7]. The coordinate frame is already gravity-aligned: +y is up and the floor sits near y = 0. No extra transform is applied in the viewer.

The public scenes are not retrained. Kitchen, counter, and bonsai were cropped, floaters were pruned, and view-dependent colour was baked from spherical harmonics of degree 3 over the training views. The living room was cleaned and pruned. Those steps change the Gaussian set that labelling sees. They are not a new training run, so this paper does not report a PSNR for them.

C. Preprocessing before labelling

Labelling needs the same camera model the renderer uses, a finite colour and opacity on every Gaussian, and a list of names. Names for My Table were typed for that job. My Desk used an automatic vocabulary. The public scenes used a fixed list per scene, including a few phrase groups such as “something to sit on”.

Invalid Gaussians are not given a class of their own. Anything that matches no object mask can fall into other. Structural classes such as wall, ceiling, floor, and ground exist so they can be excluded from search chips later.

D. Feature preparation

Each Gaussian keeps the attributes 3D Gaussian Splatting already optimised: position, scale, rotation, colour, and opacity [1]. The only added feature is an 8-bit class id, one byte per Gaussian, in the same order as the splat file. The label file is therefore N bytes for N Gaussians. A side JSON stores the class names, the home camera, saved views, and the per-class counts.

The byte order must match the splat order. A swapped label buffer would paint the right names on the wrong objects. The viewer checks the counts against the file length and does not invent a class for a short buffer.

 This work is licensed under a Creative Commons Attribution 4.0 International License

ISSN (O) 2278-1021, ISSN (P) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511150

V. OBJECT LABLES AND SEARCH

Five mechanisms make the scene searchable. They share one id per Gaussian.

A. Masks

For each labelling view, Grounding DINO proposes boxes for the text prompts [5], and SAM ViT-B turns each box into a mask [4]. The detector used in this project is Grounding-DINO-tiny. The views are posed cameras already known from COLMAP or from the original capture, not random snapshots. Table II lists how many views were used: 52 for the living room, 60 for My Table, My Desk, and the counter, 70 for the Lego table, and 73 for the bonsai studio.

B. Visibility vote

A mask is a 2D claim. A Gaussian is a 3D object that may sit behind the surface the mask shows. Projecting every Gaussian into the image and accepting the mask would label hidden points.

The lift therefore counts a vote only when the Gaussian projects inside the mask and is treated as visible in that view. Across the labelling views, each Gaussian keeps a tally per class. Its label is the class with the strongest tally. Gaussians with no real vote stay in a background class. A k-nearest-neighbour smooth then replaces isolated labels with the label of nearby Gaussians, which removes speckle without drawing a box. The living-room labels were produced by the vote alone. The other five scenes use the vote and the smoothing pass.

C. Search by name

The query is a class name or a phrase group, not a free-form sentence embedding. “bottle”, “oven gloves”, and “something to sit on” all resolve to one or more class ids. Every Gaussian with one of those ids is a hit.

Hits are drawn in the colour stored on the Gaussian. Non-hits are dimmed. A light rim marks the hit so it remains visible when the true colour is dark. The camera moves from its current pose to a view that frames the hit. This is a fly-to of the existing camera rig, not a cut to a stored thumbnail.

D. Hover

Hover has to work on thin objects. A pick on every mouse event is wasteful, so the viewer keeps the latest pointer position and picks once per animation frame.

Candidate Gaussians come from a coarse spatial hash. The pick is the labelled splat whose centre projects closest to the cursor, preferring the one in front when two centres land near the same pixel. The accept radius is 16 CSS pixels. A label already showing is kept until another class is clearly closer, so the name does not flicker across a gap. Classes named other, wall, ceiling, floor, or ground are not shown. A class with fewer than 50 Gaussians in the loaded file is not shown either. That rule is why the unlabelled watch on My Table never appears as a chip. The name hides when the pointer leaves, and while the user is orbiting. On a touch screen, a tap shows the name and a tap elsewhere clears it.

E. Adding a scene

A person who does not have a GPU can still add a scene. The Colab worker accepts a video or photos, a name, a comma-separated object list, and a quality of standard or high. Standard is the default and is the setting used for My Table and My Desk. High is offered for a longer run. The browser does not start a second upload if a job id for that file is already stored.

VI.IMPLEMENTATION

A. Viewer

The viewer is a single web page. The scene is drawn with WebGL2. The splat file is the antimatter format: 32 bytes per Gaussian, position and scale as float32, colour and opacity as bytes, rotation as four bytes. Because the file is ordered coarse to fine, drawing the prefix first is a progressive load, not a second model.

The home camera, field of view, and saved views are read from the label JSON. The landing page starts My Table at that home camera and orbits slowly until the user drags. Dragging turns the automatic orbit off. The full studio adds the same search field, hover, and an upload sheet. Search does not change the label of a Gaussian. It only changes colour, dimming, and the camera.

 This work is licensed under a Creative Commons Attribution 4.0 International License

ISSN (O) 2278-1021, ISSN (P) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511150

B. Training service

Training is a notebook on Colab, not a process in the browser. The notebook registers frames, runs COLMAP, trains, labels, and returns files. The browser's only long-lived state is the worker URL and the job id. If the Colab session has died, the page says so and does not pretend the old job is still running.

C. What the user sees

The studio shows a scene library of the six captures, a search box, and the canvas. A dashboard in the sense of the intrusion-detection paper is not needed. The result of a query is the picture itself: the object in colour, the room dimmed, the camera moved. Counts that matter for debugging, such as how many Gaussians carry a class, stay in the label file and in the chips that pass the 50-splat rule.

VII. RESULTS AND DISCUSSION

A. Reconstruction

PSNR and wall-clock train time were recorded for the two scenes this project actually trained. They were not recorded for the four scenes that reuse a published checkpoint. Those cells are left blank on purpose.

TABLE III: TRAINING RESULTS WHERE THEY WERE MEASURED

Scene	Steps	GPU	Quality	Train time	PSNR (dB)	Gaussians
My Table	12,000	Colab T4	Standard	26 min 12 s (1,572 s)	29.69	366,117
My Desk	12,000	Colab T4	Standard	16 min 39 s (999 s)	31.82	353,436
Living Room	30,000 (published)	—	—	—	—	780,000
Lego Bulldozer Table	30,000 (published)	—	—	—	—	780,000
Kitchen Counter	30,000 (published)	—	—	—	—	780,000
Bonsai Studio	30,000 (published)	—	—	—	—	780,000

My Desk is the shorter train and the higher PSNR, and it is also the capture that lost 32 frames at registration. My Table used every frame and still sits 2.13 dB lower. The gap is the scene, not a claimed ranking of the trainer. Both phone scenes are softer than the published Mip-NeRF 360 reconstructions, which is expected from a short video and 12,000 steps against a 30,000-iteration research capture.

B. Label coverage

Coverage is the number of classes that received at least one Gaussian, not a precision or recall. Those detection metrics were not computed against a hand-labelled Gaussian set, so they are not reported.

On My Table, 10 classes were defined and 9 received splats. The watch was typed as a name and received none, so it cannot be searched. The six classes exposed as search targets are table (212,040 splats), bottle (15,562), notebook (13,284), remote control (12,607), earphones (12,525), and keys (1,485). Ceiling received 44 splats and is hidden. other and wall hold a large share of the scene and are hidden from chips and from hover.

On My Desk, 17 classes were defined and 15 received splats. Floor and ground are empty. Twelve object classes received splats under the automatic vocabulary. The viewer does not offer all twelve as chips. Notes on that scene mark tool, shoe, game controller, computer mouse, and remote control as shaky even when the count is at least 50. The reliable search targets used in the app are laptop, keyboard, notebook, desk, and electronics.

The public scenes are denser. Living Room has 20 object classes with splats over 52 views. The Lego table has 11 over 70 views. The counter has 19 over 60 views. The bonsai studio has 16 over 73 views. Example searches that resolve on

 This work is licensed under a Creative Commons Attribution 4.0 International License

ISSN (O) 2278-1021, ISSN (P) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511150

those labels are sofa, piano, and teddy bear; Lego bulldozer, oven gloves, and placemat; egg carton, bread, and steel bowl; bonsai tree, piano keyboard, and bicycle.

C. Search behaviour

Because the label is an id, a query either matches a class or it does not. There is no score threshold to tune and no heatmap to read. On My Table, "bottle" frames the bottle and leaves the blue cloth dimmed. "earphones" does the same for the earphones. On the counter, "oven gloves" frames the gloves. Hover agrees with search: the name drawn beside the pointer is the class id of the splat under it, after the filters in Section V-D.

This is a functional result, not a measured ranking. A future evaluation would need a list of queries, a clicked ground-truth object, and a rate for wrong class, missed class, and hover flicker.

D. What the numbers do not say

A PSNR of 29.69 dB does not mean the bottle is labelled correctly. A class count of 20 does not mean all 20 are equally clean. My Desk is the example: more object classes than My Table, and several of them marked unreliable. High accuracy on a reconstruction metric and a clean label are different claims. This paper makes only the first, and only for two scenes.

VIII. DISCUSSION

The useful property of the system is that one byte per Gaussian serves every interaction. Search, hover, chips, and the dimming pass cannot disagree about what an object is, because they read the same id. A heatmap system can rank a region highly and still not know which splats to keep in colour.

The cost of that simplicity is a closed vocabulary. If the name was not in the prompt list, the splat cannot acquire it later by rephrasing the question. Phrase groups such as "something to sit on" only help when someone has already mapped the phrase onto classes that exist.

The phone path shows that a free T4 is enough for a small table or desk at Standard quality. It also shows the failure modes of a short video. COLMAP dropped 32 frames on My Desk. The watch on My Table was never masked, so the vote had nothing to count. Thin, reflective, or rarely seen objects are exactly where a visibility vote is weakest.

Compared with a signature-style detector run every frame, the offline vote is cheap at view time and blind to anything new. Compared with a language field, it is less flexible and much easier to explain. The user sees a name, not a score.

IX. LIMITATIONS

However, there are a number of problems in the current system.

- The two phone scenes are single rooms, filmed once. They do not represent all cameras, lighting, or motion blur.
- PSNR and train time exist only for those two Standard runs. The four public scenes were not retrained here.
- No precision, recall, or F1 was measured for the labels. Coverage counts are not detection accuracy.
- My Desk's automatic vocabulary produced classes the viewer then refused to trust. Typed names, as on My Table, still miss an object the detector never segments.
- Input for a new scene is a video or a CSV-like file of images, then a wait. The viewer does not capture packets of live video from a webcam and label them on the fly.
- Hover and search both fail closed: a class under 50 splats, or a background class, is invisible. That hides noise and also hides a real object that received only a few votes.
- A dead Colab session stops training. The browser cannot finish the job locally.

X. FUTURE SCOPE

The above mentioned system can be extended in many ways.

A. A label benchmark

Hand-label a sample of Gaussians, or click the object that a query should have found, and report precision and recall per class. Until that exists, Table III should stay a reconstruction table.

B. Live capture

Stream a phone or a webcam into the same COLMAP-free pose tracker, update the Gaussians, and run masks only on keyframes. The current upload-and-wait path is the blocking limitation for a demo that walks through a room.

C. A safer vocabulary

Let the user confirm chips before they are searchable, and drop classes the vote only barely won. My Desk's shaky labels are the test case.

 This work is licensed under a Creative Commons Attribution 4.0 International License

ISSN (O) 2278-1021, ISSN (P) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511150

D. Why this splat

A short explanation, for example the views that voted for the bottle, would tell the user why a name appeared. The id is already stored. The votes are not.

E. Higher quality, measured

High quality is already a worker option. It should be run on the same two videos and added as rows in Table III, including train time, PSNR, and which previously empty objects, such as the watch, receive splats.

F. Open phrases without giving up colour

A CLIP rank could propose a class the closed list missed, while the accepted hit is still drawn in true colour. That would connect this viewer to LERF and LangSplat [8], [9] without returning to a heatmap as the answer.

XI. CONCLUSION

SplatSearch labels 3D Gaussians with object names and lets a person search those names in a browser. A phone video is posed with COLMAP, trained with gsplat MCMC for 12,000 steps on a Colab T4, and labelled from Grounding DINO and SAM masks by a visibility vote. Four published Mip-NeRF 360 reconstructions are labelled the same way and shown beside the phone scenes.

My Table, with all 150 frames registered, reached 29.69 dB in 26 min 12 s and 366,117 Gaussians. My Desk, with 118 of 150 frames registered, reached 31.82 dB in 16 min 39 s and 353,436 Gaussians. Search keeps the named splats in their own colour, dims the rest, and moves the camera. Hover reads the same labels, including thin objects, and suppresses background and tiny classes.

The system is a complete path from a clip to a search box. It is not a measured detector. Labels are only as good as the masks, the watch on My Table was missed entirely, and live capture is future work.

Conflict of interest: The authors declare that there is no conflict of interest regarding the publication of this paper.

Data: The four public scenes are the Mip-NeRF 360 captures and the INRIA 3D Gaussian Splatting reconstructions cited below. My Table and My Desk are phone captures collected for this project. Train time, PSNR, frame counts, and per-class splat counts quoted above were read from those scenes' label files.

AI-assisted editing: Language and section order were drafted with an assistant to match the required paper template. The numbers, datasets, and system behaviour were checked against the project files.

REFERENCES

[1] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3D Gaussian Splatting for Real-Time Radiance Field Rendering," *ACM Trans. Graph.*, vol. 42, no. 4, art. 139, 2023, doi: 10.1145/3592433.

[2] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022, pp. 5470–5479, doi: 10.1109/CVPR52688.2022.00539.

[3] J. L. Schönberger and J.-M. Frahm, "Structure-from-Motion Revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 4104–4113, doi: 10.1109/CVPR.2016.445.

[4] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment Anything," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023, pp. 4015–4026, doi: 10.1109/ICCV51070.2023.00371.

[5] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, J. Zhu, and L. Zhang, "Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2024, pp. 38–55, doi: 10.1007/978-3-031-72970-6_3.

[6] S. Kheradmand, D. Rebain, G. Sharma, W. Sun, Y.-C. Tseng, H. Isack, A. Kar, A. Tagliasacchi, and K. M. Yi, "3D Gaussian Splatting as Markov Chain Monte Carlo," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2024.

[7] V. Ye, R. Li, J. Kerr, M. Turkulainen, B. Yi, Z. Pan, O. Seiskari, J. Ye, J. Hu, M. Tancik, and A. Kanazawa, "gsplat: An Open-Source Library for Gaussian Splatting," *J. Mach. Learn. Res.*, vol. 26, no. 225, pp. 1–17, 2025.

[8] J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik, "LERF: Language Embedded Radiance Fields," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023, pp. 19729–19739, doi: 10.1109/ICCV51070.2023.01807.

[9] M. Qin, W. Li, J. Zhou, H. Wang, and H. Pfister, "LangSplat: 3D Language Gaussian Splatting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024, pp. 20051–20060, doi: 10.1109/CVPR52733.2024.01895.

[10] S. Zhou, H. Chang, S. Jiang, Z. Fan, Z. Zhu, D. Xu, P. Chari, S. You, Z. Wang, and A. Kadambi, "Feature 3DGS: Supercharging 3D Gaussian Splatting to Enable Distilled Feature Fields," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024, pp. 21676–21685, doi: 10.1109/CVPR52733.2024.02048.

[11] M. Ye, M. Danelljan, F. Yu, and L. Ke, "Gaussian Grouping: Segment and Edit Anything in 3D Scenes," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2024, pp. 162–179, doi: 10.1007/978-3-031-73390-1_10.

 This work is licensed under a Creative Commons Attribution 4.0 International License

ISSN (O) 2278-1021, ISSN (P) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511150

[12] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 405–421, doi: 10.1007/978-3-030-58452-8_24.

[13] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant Neural Graphics Primitives with a Multiresolution Hash Encoding," *ACM Trans. Graph.*, vol. 41, no. 4, art. 102, 2022, doi: 10.1145/3528223.3530127.

[14] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning Transferable Visual Models From Natural Language Supervision," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 8748–8763.

[15] B. Li, K. Q. Weinberger, S. J. Belongie, V. Koltun, and R. Ranftl, "Language-driven Semantic Segmentation," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2022.

[16] W. Shen, G. Yang, A. Yu, J. Wong, L. P. Kaelbling, and P. Isola, "Distilled Feature Fields Enable Few-Shot Language-Guided Manipulation," in *Proc. Conf. Robot Learn. (CoRL)*, 2023, pp. 405–424.

[17] C. M. Kim, M. Wu, J. Kerr, K. Goldberg, M. Tancik, and A. Kanazawa, "GARField: Group Anything with Radiance Fields," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024, pp. 21530–21539, doi: 10.1109/CVPR52733.2024.02034.

[18] J.-C. Shi, M. Wang, H.-B. Duan, and S.-H. Guan, "Language Embedded 3D Gaussians for Open-Vocabulary Scene Understanding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024, pp. 5333–5343, doi: 10.1109/CVPR52733.2024.00510.

[19] J. Cen, J. Fang, C. Yang, L. Xie, X. Zhang, W. Shen, and Q. Tian, "Segment Any 3D Gaussians," in *Proc. AAAI Conf. Artif. Intell.*, 2025. arXiv:2312.00860.

[20] Y. Wu, J. Meng, H. Li, C. Wu, Y. Shi, X. Cheng, C. Zhao, H. Feng, E. Ding, J. Wang, and J. Zhang, "OpenGaussian: Towards Point-Level 3D Gaussian-based Open Vocabulary Understanding," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2024, pp. 19114–19138.