# MONOVOC: EFFICIENT OBJECT-CENTRIC OPEN-VOCABULARY 3D GAUSSIAN MAPPING FROM MONOCULAR VIDEO

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Open-vocabulary 3D scene understanding lets users query reconstructed environments in natural language, but existing Gaussian-based methods either store dense per-Gaussian language features or couple semantics to an optimization-heavy reconstruction, both of which scale poorly on ordinary monocular video. We present MonoVoc, an object-centric framework that turns a single monocular walkthrough and its segmentation masks into a compact semantic Gaussian map paired with an object-level language database. MonoVoc attaches three learnable semantic coefficients to each Gaussian of a fixed SLAM backbone, recovers per-Gaussian semantic evidence by windowed deblending restricted to projected Gaussian footprints, and resolves discrete object identities through confidence-gated quantization that combines palette agreement with spatial consistency. Each object is then described once from a set of diverse keyframes and encoded into a single shared embedding, adding semantics on top of reconstruction without a separate language-feature distillation stage and reducing language storage from $O(ND)$ to $O(N+KD)$ for $N$ Gaussians and $K$ objects. On Replica and ScanNet, MonoVoc produces a queryable semantic map in 13-13.5 minutes end to end, against 186 minutes for ObjectGS, using $\approx$160K Gaussians and 14 MB; 84% and 91% less memory than ObjectGS and SceneSplat. Built on HI-SLAM2, MonoVoc inherits its rendering quality and reaches 91.65 mIoU on Replica given ground-truth 2D segmentation, 84%/93% Top-1/Top-3 object retrieval, and 86% accuracy on a controlled scene-grounded QA benchmark, while ObjectGS retains an edge on object removal. A 34-participant study prefers our maps in 52.9% of responses.

## 1 INTRODUCTION

Open-vocabulary 3D scene understanding enables interactive systems to connect natural language with the physical world. Applications in augmented reality (Kato et al., 2023), robotics (Werby et al., 2024a), and inspection (Tasneem & Pieters, 2026) require users to identify objects through descriptions such as "the red toolbox" or "the chair behind the table," beyond a predefined set of categories.

Recent advances in 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) have enabled language-based interaction with reconstructed scenes by incorporating semantic or language features. SceneSplat (Li et al., 2025c) learns open-vocabulary 3D representations, OpenGaussian (Wu et al., 2024) establishes instance-level semantic associations, and ObjectGS (Zhu et al., 2025) introduces object awareness into monocular Gaussian reconstruction. LangSplatV2 (Li et al., 2025a) and 4D LangSplat (Li et al., 2025b) further address query efficiency and dynamic scenes, respectively.

Despite this progress, practical monocular systems must balance semantic capabilities with reconstruction speed and memory efficiency. Gaussian SLAM (Matsuki et al., 2024; Keetha et al., 2024) and language-driven segmentation (Li et al., 2022) provide foundations for this goal. However, integrating them into compact, editable scene representations remains challenging: semantic processing can require additional feature learning or per-scene optimization, while storing dense features across Gaussians increases memory demands. These challenges motivate a lightweight monocular frame-



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

work that organizes Gaussian maps around objects and supports both language-based interaction and queries about their spatial relationships.

We introduce a monocular open-vocabulary scene understanding framework built on a SLAM-generated 3DGS map. Given monocular frames and segmentation masks, the framework learns semantic color parameters alongside the Gaussian representation and derives confidence estimates to guide semantic quantization and downstream processing. Compact object-level embeddings connect the map to language, while geometric information supports queries involving spatial relationships. We organize open-vocabulary scene understanding into four downstream tasks: object removal, object retrieval, question answering, and segmentation. Our framework supports these tasks with improvements in processing speed and memory use. The proposed method section details the individual components and their integration.

## 2 Related Work

Open-vocabulary 3D scene understanding connects geometry with language for localization, segmentation, and interaction. Approaches differ in acquisition, semantic association, and cost, including language fields, semantic mapping, and object-centric or generalizable Gaussian representations.

### 2.1 Language-grounded scene representations

Language fields associate observations with spatial features for queries beyond fixed vocabularies. LERF (Kerr et al., 2023) embeds language features in a NeRF (Mildenhall et al., 2020) and renders text relevancy maps, requiring scene-specific optimization. Gaussian Grouping (Ye et al., 2024) learns object identities for segmentation and editing. LangSplat (Qin et al., 2023) learns compressed language features; LangSplatV2 (Li et al., 2025a) accelerates high-dimensional rendering, and LEGaussians (Shi et al., 2024) quantizes features to reduce storage. 4D LangSplat (Li et al., 2025b) handles dynamic scenes. OpenGaussian (Wu et al., 2024) establishes instance-level 2D–3D associations; LangSurf (Li et al., 2024a) aligns features with surfaces. SLGaussian (Chen et al., 2025) uses sparse views without per-scene optimization, SLAG (Szilagyi et al., 2025) addresses scalable language augmentation, and LUDVIG (Marrie et al., 2025) provides learning-free feature aggregation and refinement on existing Gaussians.

Object-level mapping also supports language-based scene understanding. ConceptGraphs (Gu et al., 2024) associates object features and captions with point-cloud scene graphs; HOV-SG (Werby et al., 2024b) builds hierarchical segment-level language representations. Both use posed RGB-D observations. SGS-SLAM (Li et al., 2024b) integrates semantic colors into Gaussian tracking and mapping; OVO (Martins et al., 2025) associates language features with segments in SLAM maps. These approaches establish related representation choices, but differ from MonoVoc's color recovery and confidence-gated Gaussian assignment. These distinctions concern the complete reconstruction-and-understanding pipeline: object descriptions alone do not specify how semantic identities are recovered, how observations are associated, or how geometric uncertainty affects the resulting Gaussian representation.

A single-video monocular walkthrough follows one uninterrupted camera trajectory; multitrajectory capture combines videos recorded along different paths, potentially increasing coverage. Both provide multiple viewpoints, so multiview processing alone does not imply multitrajectory input. Sparse-view inference and learning-free feature transfer do not establish robustness across an entire reconstruction pipeline. Occlusion and inconsistent 2D semantic predictions affect cross-view agreement, boundaries, and small objects. Fast querying on existing geometry likewise excludes reconstruction costs. Comparisons require consistent observations and explicit accounting for pose estimation, reconstruction, semantic processing, and feature storage.

### 2.2 Object-centric and generalizable 3DGS

Object-centric representations organize Gaussians by entity, whereas generalizable representations learn features transferable across scenes. ObjectGS (Zhu et al., 2025) uses local anchors to generate Gaussians with shared object identities. Joint optimization of the representation and semantic con-

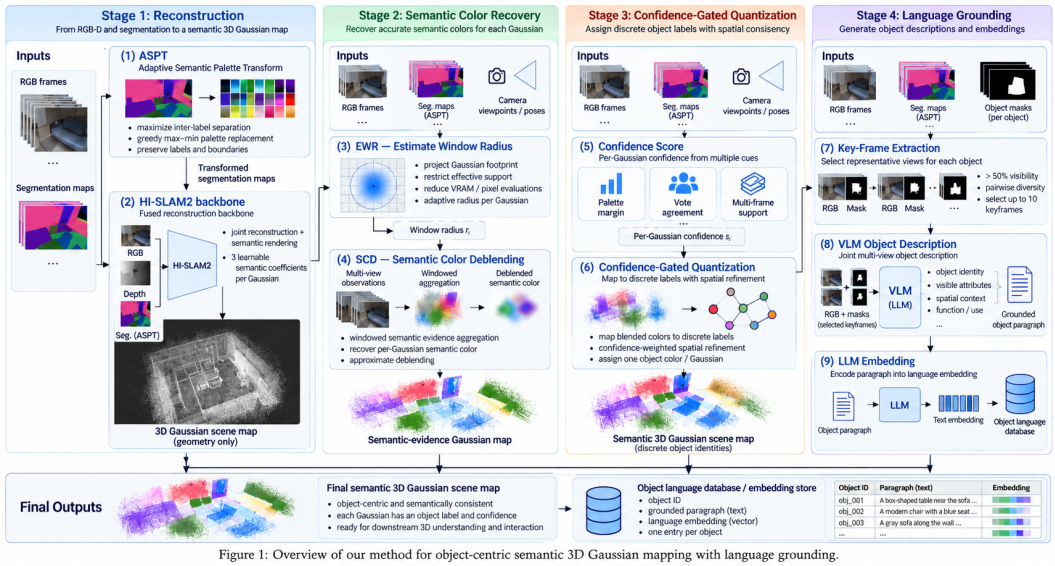


Figure 1: Overview of our method for object-centric semantic 3D Gaussian mapping with language grounding.

Figure 1: Overview of MonoVoc. Four stages connect semantic Gaussian reconstruction, windowed color recovery, confidence-gated object assignment, and language grounding. The outputs are a Gaussian map and a database storing one description and embedding per object.

straints supports object separation and editing, but ties semantic construction to its reconstruction procedure.

SceneSplat (Li et al., 2025c) uses large-scale pretraining to infer semantic features on reconstructed Gaussian scenes without scene-specific feature optimization. It depends on prior training and an available reconstruction, rather than providing a complete monocular reconstruction-and-query pipeline. Transfer across reconstruction methods requires matched evaluation, including training costs and representation size.

MonoVoc targets the gaps like efficient semantic mapping from one uninterrupted monocular video with estimated geometry, limited coverage, and explicit object-level access. After monocular reconstruction, semantic color recovery and confidence-gated assignment connect Gaussians to shared object-level language embeddings. We evaluate ObjectGS and SceneSplat as complementary baselines under this protocol, accounting for their differing requirements across segmentation, retrieval, removal, question answering, runtime, and memory.

## 3 Proposed Method

MonoVoc comprises four stages, illustrated in Figure 1. First, ASPT separates the segmentation palette colors, and our modified HI-SLAM2 backbone reconstructs a Gaussian map with learnable semantic coefficients. Second, EWR determines spatial evaluation windows for SCD, which recovers per-Gaussian semantic color evidence. Third, Confidence-Gated Quantization combines palette agreement with spatial consistency to assign discrete object identities, with per-Gaussian confidence retained for downstream use. Finally, language grounding selects representative frames for each object, generates a description incorporating appearance and spatial context, and encodes it into a shared object-level embedding. The resulting semantic Gaussian map and object language database support object removal, object retrieval, question answering, and segmentation. The following subsections detail each component.

### 3.1 Adaptive Semantic Palette Transform (ASPT)

Adaptive Semantic Palette Transform (ASPT) recolors segmentation maps before they enter our modified HI-SLAM2 backbone, increasing separation between semantic labels to reduce color confusion during subsequent deblending and quantization. Given the input palette, ASPT selects an

equally sized replacement palette from RGB candidates using a greedy max–min rule:

$$\boldsymbol{c}_{k+1} \in \arg\max_{\boldsymbol{c}\in\mathbb{C}\setminus\mathbb{P}_k} \min_{\boldsymbol{u}\in\mathbb{P}_k} \min_{m\in\mathbb{M}} \frac{d_m(\boldsymbol{c},\boldsymbol{u})}{\lambda_m}, \tag{1}$$

where $\mathbb{C} \subseteq \{0,\dots,255\}^3$ is the candidate set, $\mathbb{P}_k$ contains the colors already selected or preserved, and $d_m$ and $\lambda_m > 0$ denote a color dissimilarity and its normalization scale. Optional local refinement further improves separation, after which a one-to-one mapping replaces the original colors consistently across all frames, preserving the supplied labels and mask boundaries. ASPT reduces ambiguity in the color representation, complementing the confidence and spatial information used by Confidence-Gated Quantization. Appendix A explains the overlap mechanism, and Appendix B discusses the motivating color-confusability experiment and palette construction.

### 3.2 HI-SLAM2

We use HI-SLAM2 (Zhang et al., 2025) as a fixed reconstruction backbone, extending its Gaussian map with three learnable semantic coefficients $\boldsymbol{s}_i \in \mathbb{R}^3$ per Gaussian. Each video frame is paired with its ASPT-transformed segmentation map, allowing the system to optimize scene appearance and semantic rendering alongside reconstruction. The semantic coefficients are optimized through alpha compositing: contributions from overlapping Gaussians jointly reproduce each observed segmentation color, so individual coefficients need not correspond to interpretable palette colors. SCD subsequently recovers semantic color evidence, which Confidence-Gated Quantization converts into discrete object assignments for shared object-level embeddings. Appendix C explains this choice of semantic color recovery over dense per-Gaussian language features.

### 3.3 Effective Window Radius (EWR)

Effective Window Radius (EWR) restricts SCD computation to the effective image-space support of each Gaussian, reducing unnecessary pixel evaluations and temporary VRAM usage. Given the reconstructed map, camera poses, and intrinsics, EWR projects Gaussian covariances into sampled frames and estimates footprint radii for visible Gaussian–frame pairs:

$$r_{i,t} = 3\sqrt{\max\left(\sigma^2_{x,i,t}, \sigma^2_{y,i,t}\right)}, \qquad R^\star = \max\left(R_{\min}, \lceil Q_{0.999}(\{r_{i,t}\})\rceil\right), \tag{2}$$

where $\sigma^2_{x,i,t}$ and $\sigma^2_{y,i,t}$ are the projected coordinate variances, $Q_{0.999}$ is the empirical 99.9th percentile, and $R_{\min}$ is a minimum permitted cap. SCD then evaluates each Gaussian within a square window centered on its projection, using radius $\min(\lceil r_{i,t}\rceil, R^\star)$. Sorting visible Gaussians by radius before batching further reduces padding and memory allocation. Appendix D explains the footprint approximation, memory model, and observed agreement with dense evaluation.

### 3.4 Semantic Color Deblending (SCD)

SCD recovers per-Gaussian semantic color evidence by aggregating pixel observations within the windows determined by EWR. Let $p$ index a pixel–frame observation from the input video. With geometry, opacity, and depth ordering fixed, semantic rendering takes the form

$$\boldsymbol{C}_p = \sum_{i=1}^{N} w_{i,p}\boldsymbol{s}_i + T_{\mathrm{bg},p}\boldsymbol{b}, \qquad w_{i,p} = \alpha_{i,p}\prod_{j\prec_p i}(1-\alpha_{j,p}), \tag{3}$$

where $\boldsymbol{s}_i \in \mathbb{R}^3$ contains Gaussian $i$'s learned semantic coefficients, $\alpha_{i,p}$ is its projected opacity, $j \prec_p i$ denotes front-to-back ordering, and $T_{\mathrm{bg},p}\boldsymbol{b}$ is the background contribution. Subtracting this contribution gives $\boldsymbol{y}_p = \boldsymbol{C}_p - T_{\mathrm{bg},p}\boldsymbol{b}$, the semantic evidence used for recovery.

For Gaussian $i$, let $\mathbb{W}_i$ contain the pixel-frame observations inside its EWR-defined windows. SCD computes the normalized weighted estimate

$$\hat{\boldsymbol{s}}_i = \frac{\sum_{p\in\mathbb{W}_i} w_{i,p}\boldsymbol{y}_p}{\sum_{p\in\mathbb{W}_i} w_{i,p}}, \qquad \sum_{p\in\mathbb{W}_i} w_{i,p} > 0. \tag{4}$$

This windowed accumulation concentrates computation on each Gaussian's effective footprint, reducing temporary storage and pixel evaluations. The estimate is an approximate recovery rather than a global inversion. Confidence-Gated Quantization subsequently uses this evidence to determine discrete semantic assignments. Appendix M establishes the estimator's optimization interpretation, recovery conditions, and approximation error.

### 3.5 CONFIDENCE-GATED QUANTIZATION

Confidence-Gated Quantization converts SCD color estimates into discrete object assignments by combining palette proximity with spatial consistency. We construct an undirected 24-nearest-neighbor graph over Gaussian centers and assign bilateral edge weights $a_{ij}$ using covariance-aware spatial distances and normal agreement, with optional color similarity. Only Gaussians with nonzero accumulated support participate; sums below range over these Gaussians and edges between them. For palette colors $\{\boldsymbol{c}_k\}_{k=1}^{K}$, we define

$$\begin{aligned} D_i(k) &= d_{\mathrm{CIEDE2000}}\big(\mathrm{Lab}(\hat{\boldsymbol{s}}_i), \mathrm{Lab}(\boldsymbol{c}_k)\big), \\ E(\boldsymbol{z}) &= \sum_i C_i o_i D_i(z_i) + \lambda \sum_{(i,j)\in\mathcal{E}} a_{ij}\sqrt{C_i C_j}\,\mathbf{1}_{z_i \neq z_j}. \end{aligned} \tag{5}$$

where $o_i$ is Gaussian opacity and $C_i$ is the combined confidence computed from the initial nearest-palette labels $z_i^{(0)}$ using Equation 7. We hold $C_i$, unary costs, edge weights, and $\lambda$ fixed during optimization. Sequential iterated conditional modes (ICM) updates one supported Gaussian at a time using the latest neighbor labels, retaining the current label on ties. Each update is non-increasing in energy; optimization stops when labels stabilize or after 20 sweeps. Assigned Gaussians receive $\boldsymbol{c}_{z_i}$, while unsupported Gaussians remain unassigned. Appendix E gives the preparation and optimization algorithms.

#### 3.5.1 CONFIDENCE SCORE

We measure semantic reliability through palette distinctiveness, agreement across video frames, and observation support. The palette confidence $q_i$ measures how clearly the recovered color favors its nearest palette entry over the strongest alternative. Given the smallest and second-smallest pre-quantization costs, $D_i^{(1)}$ and $D_i^{(2)}$, we define

$$\Delta_i = D_i^{(2)} - D_i^{(1)}, \qquad q_i = \frac{\Delta_i}{\Delta_i + \tau}, \qquad \tau = \mathrm{median}_{i\in\mathcal{I}_+}\,\Delta_i + \epsilon. \tag{6}$$

Here $\mathcal{I}_+$ denotes Gaussians with nonzero accumulated support. Before quantization, $C_i^{\mathrm{vote}}$ measures agreement with the initial nearest-palette label $z_i^{(0)} = \arg\min_k D_i(k)$, while $C_i^{\mathrm{support}}$ measures support distributed across frames through their effective number:

$$\begin{aligned} C_i^{\mathrm{vote}} &= \frac{\sum_{f,p} w_{i,p,f}\,\mathbf{1}_{L_{p,f}=z_i^{(0)}}}{\sum_{f,p} w_{i,p,f} + \epsilon}, \qquad A_{i,f} = \sum_p w_{i,p,f}, \qquad N_i^{\mathrm{eff}} = \frac{\left(\sum_f A_{i,f}\right)^2}{\sum_f A_{i,f}^2 + \epsilon}, \\ C_i^{\mathrm{support}} &= 1 - \exp\!\left(-N_i^{\mathrm{eff}}/\tau_N\right), \qquad C_i = \left(q_i C_i^{\mathrm{vote}} C_i^{\mathrm{support}}\right)^{1/3}. \end{aligned} \tag{7}$$

Here, $L_{p,f}$ is the segmentation label and $w_{i,p,f}$ is the SCD compositing weight within the evaluated windows. The effective count distinguishes evidence distributed across frames from evidence concentrated in a few; it does not directly measure total observation weight. We set $\tau_N = \max(\mathrm{median}_{i\in\mathcal{I}_+} N_i^{\mathrm{eff}}, \epsilon)$. The geometric mean $C_i$ combines palette distinctiveness, vote agreement, and view support. Quantization uses $C_i$ in both its unary costs and symmetric edge gate; $q_i$ is only the palette-margin component of $C_i$. Neither score is recomputed after label updates. Stored $C_i$ therefore describes the initial evidence used for optimization, not agreement with the final label. Unsupported Gaussians receive $C_i = 0$ and $z_i = \perp$ and are excluded from semantic assignment and rendering.

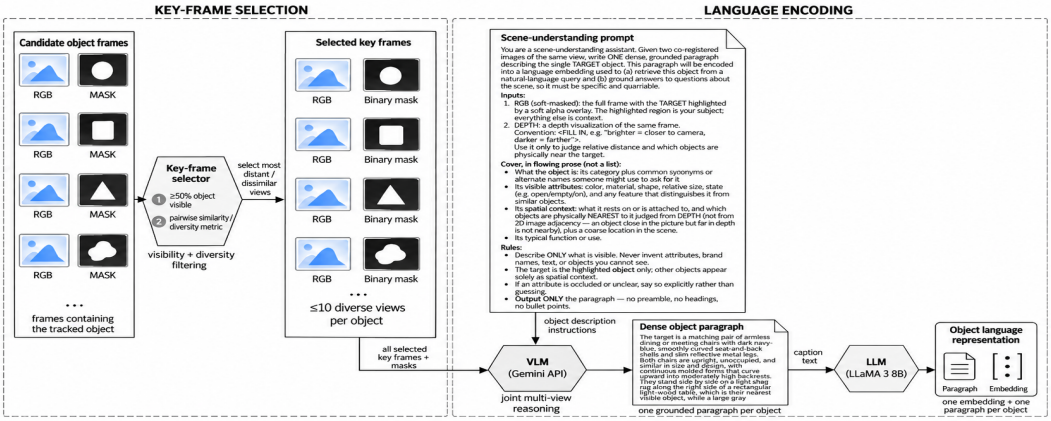



Figure 2: Language grounding through diverse frame selection, joint object description, and object-level text embedding.

### 3.6 Language Grounding

We associate each quantized object with a compact language representation using selected frames from the input video. Candidate frames must provide at least $50\%$ object visibility and are paired with their segmentation masks. We compare candidates using a pairwise similarity measure and select up to ten dissimilar frames per object, favoring complementary observations while reducing redundancy. The masks identify the target object within each frame, allowing its appearance to be interpreted alongside the surrounding scene context.

All selected frames and masks are submitted jointly to Gemini with a scene-understanding prompt, producing one grounded paragraph per object. Beyond identity and appearance, the prompt captures geometric context through descriptions of nearby objects and their positions relative to the target. We encode this paragraph using LLaMA 3 8B and store both the embedding and the description under the object's identity. This incorporates spatial context into the language representation, supporting queries about object relationships alongside appearance-based retrieval. All Gaussians assigned to the object share this representation. Figure 2 illustrates the pipeline, prompt, and an example description.

## 4 Evaluation & Results

### 4.1 Experimental Setup

We evaluate MonoVoc on Replica (Straub et al., 2019) and ScanNet (Dai et al., 2017) for efficiency, rendering, semantic label fidelity, downstream tasks, and human-perceived map quality. Modified HI-SLAM2 optimizes semantic coefficients during reconstruction; SCD and quantization require no additional training. Ground-truth segmentations are used for both supervision and evaluation, so segmentation scores measure label fidelity rather than superiority over differently supervised methods. Section 4 metrics are computed from Gaussian-splat renderings; point rendering with size $0.03$ is used only for the quantization and ASPT ablations in Appendix F. Settings are shared across scenes, with batch size 32 for unique-color extraction. All experiments use one NVIDIA A100 80GB GPU. Bold and underlined scores denote the best and second-best distinct results within comparable settings.

### 4.2 Runtime and Memory

MonoVoc completes the pipeline in 13–13.5 minutes versus ObjectGS's 186 minutes. Excluding reconstruction, it takes 9–9.5 minutes versus SceneSplat's 44 minutes (Figure 3). Its 160,000–166,000 Gaussians occupy 14–14.6 MB, with reported footprints $84\%$ below ObjectGS and $91\%$ below SceneSplat (Figure 5). These savings combine fewer Gaussians and shared embeddings: for

fixed $N$ Gaussians, $K$ objects, and dimension $D$, float32 features require $4ND$ bytes per Gaussian-based representation versus $4KD+4N$ bytes with shared embeddings and uint32 identifiers, excluding geometry and other metadata. Increasing SCD batch size from 3,000 to 20,000 reduces runtime from 90 to 40 seconds while increasing average VRAM from 7,772 to 75,850 MiB (Figure 4).

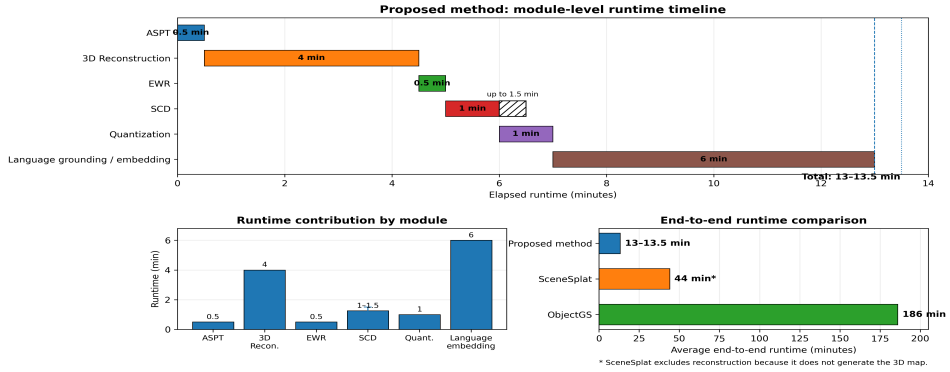



Figure 3: Module runtimes and total processing times. MonoVoc runtimes use the conservative SCD setting $n_{\text{batch}} = 3{,}000$; SceneSplat's reported time excludes reconstruction.

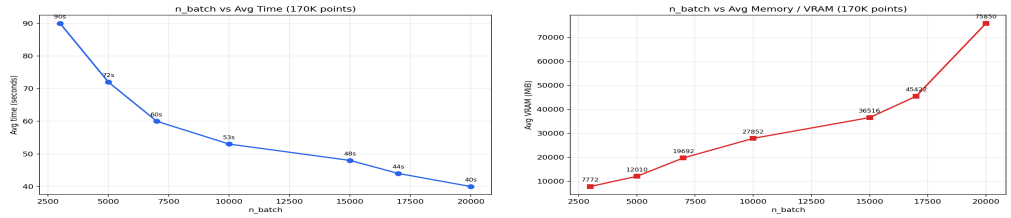



Figure 4: SCD batch-size effects on processing time and average VRAM usage.

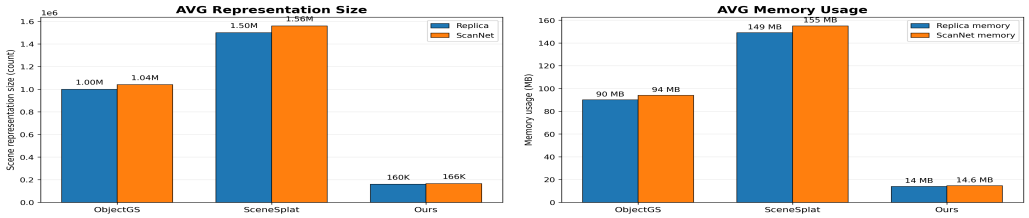



Figure 5: Average Gaussian counts and reported memory footprints on Replica and ScanNet.

### 4.3 Task Performance

**Rendering and segmentation.** We measure rendering using PSNR (Hore & Ziou, 2010), SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018), and segmentation using mIoU, mAcc (Everingham et al., 2010), and Dice (Dice, 1945). Our HI-SLAM2-based reconstruction leads PSNR on both datasets and all rendering metrics on ScanNet; ObjectGS leads Replica LPIPS. These results characterize reconstruction, not isolated semantic-module gains. With GT segmentation inputs, MonoVoc leads Replica segmentation scores and trails ObjectGS by 1.40 points in ScanNet mIoU (Table 1). These measure label-rendering fidelity under differing input supervision, rather than matched segmentation prediction. Appendix G provides examples.

**Object retrieval.** MonoVoc achieves 84% Top-1 and 93% Top-3 accuracy, exceeding SceneSplat by eight and seven percentage points, respectively (Table 2). These metrics test whether the annotated target appears among the highest-ranked objects (Song & Soleymani, 2019). Soft masking outperforms hard masking for both evaluated embedding variants. Figure 6 shows the object returned for the prompt "rug"; Appendix H details queries, category-level results, and further examples.

Table 1: Rendering and segmentation quality. Arrows indicate preferred directions; segmentation scores are percentages.

(a) Rendering quality

| Model | PSNR↑ | SSIM↑ | LPIPS↓ |
|---|---|---|---|
| **Replica** | | | |
| Ours | **40.71** | **0.98** | 0.03 |
| ObjectGS | 40.26 | **0.98** | **0.02** |
| Gaussian Grouping | 39.52 | 0.97 | 0.05 |
| **ScanNet** | | | |
| Ours | **30.27** | **0.93** | **0.13** |
| ObjectGS | 30.14 | 0.92 | 0.14 |
| Gaussian Grouping | 28.35 | 0.92 | 0.16 |

(b) Segmentation quality

| Model | mIoU↑ | mAcc↑ | Dice↑ |
|---|---|---|---|
| **Replica** | | | |
| Ours | **91.65** | **96.18** | **94.80** |
| ObjectGS | 87.39 | 92.65 | 90.39 |
| Gaussian Grouping | 81.42 | 90.12 | 89.31 |
| SceneSplat | 84.68 | 91.64 | 88.52 |
| **ScanNet** | | | |
| Ours | 92.98 | 94.58 | 93.03 |
| ObjectGS | **94.38** | **97.07** | **95.48** |
| Gaussian Grouping | 85.78 | 89.39 | 86.92 |
| SceneSplat | 77.28 | 80.27 | 78.98 |

Table 2: Illustrative object retrieval accuracy (%). Results are computed over 630 queries (42 objects × 15 query categories). Higher is better.

| Method | Variant | Top-1 | Top-3 |
|---|---|---|---|
| CLIP | Soft mask | 65.1 | 77.9 |
| | Hard mask | 57.9 | 70.0 |
| Perception Encoder | Soft mask | 73.0 | 85.1 |
| | Hard mask | 66.0 | 79.0 |
| ObjectGS | Grounded-SAM | 62.1 | 74.0 |
| SceneSplat | – | 76.0 | 86.0 |
| MonoVoc | – | **84.0** | **93.0** |

**Text prompt:** `rug`



Figure 6: Object returned for the prompt "rug".

**Question answering.** We construct a controlled, scene-grounded benchmark for selected Replica and ScanNet scenes using annotations and object-level representations; it is not a dataset-provided QA benchmark. MonoVoc achieves 86% accuracy versus 79.8% for Qwen2.5-VL-3B-Instruct and leads all six reported metrics (Table 3). Appendix I specifies question families, reference-based scoring, and benchmark-specific paraphrase and constraint metrics.

Table 3: Illustrative question-answering results on our controlled scene-grounded benchmark. QA accuracy is evaluated over 630 questions. Paraphrase consistency is evaluated over 42 paraphrase groups, Count MAE over 42 counting questions, and constraint accuracy over 84 constraint questions. Semantic similarity and judge score are averaged over all applicable questions.

| Method | Variant | QA Acc. ↑ | Semantic similarity↑ | Judge score↑ | Paraphrase consistency↑ | Count MAE↓ | Constraint Acc.↑ |
|---|---|---|---|---|---|---|---|
| CLIP | Soft mask | 61.9 | 0.78 | 3.57 | 71.4 | 0.95 | 52.4 |
| | Hard mask | 55.1 | 0.72 | 3.24 | 64.3 | 1.12 | 44.0 |
| Perception Encoder | Soft mask | 69.4 | 0.84 | 4.02 | 81.0 | 0.74 | 64.3 |
| | Hard mask | 63.0 | 0.80 | 3.71 | 73.8 | 0.90 | 54.8 |
| ObjectGS | Grounded-SAM | 58.1 | 0.75 | 3.45 | 66.7 | 0.71 | 50.0 |
| SceneSplat | – | 74.1 | 0.87 | 4.17 | 85.7 | 0.57 | 72.6 |
| Qwen2.5-VL-3B-Instruct | – | 79.8 | 0.90 | 4.39 | 88.1 | 0.48 | 81.0 |
| MonoVoc | – | **86.0** | **0.94** | **4.67** | **92.9** | **0.33** | **90.5** |

**Object removal.** MonoVoc outperforms SceneSplat across all five removal metrics, achieving 90.8% target edit coverage and 18.1% edit leakage. ObjectGS performs best overall, exceeding MonoVoc by 2.6 percentage points in TEC and by 0.031 in NT-SSIM (Table 4). Figure 7 illustrates a representative removal example. Appendix J defines the mask-guided metrics, which require no ground-truth edited scenes.

Table 4: Object removal results averaged over 15 target-object removal cases. Each metric is computed independently for each case and then averaged across cases. Arrows indicate preferred directions.

| Model | TEC↑ | NTPE↓ | ELR↓ | BSR↓ | NT-SSIM↑ |
|---|---|---|---|---|---|
| SceneSplat | 87.1% | 0.117 | 24.4% | 19.2% | 0.761 |
| ObjectGS | **93.4%** | **0.069** | **14.7%** | **10.9%** | **0.843** |
| MonoVoc | 90.8% | 0.084 | 18.1% | 13.7% | 0.812 |

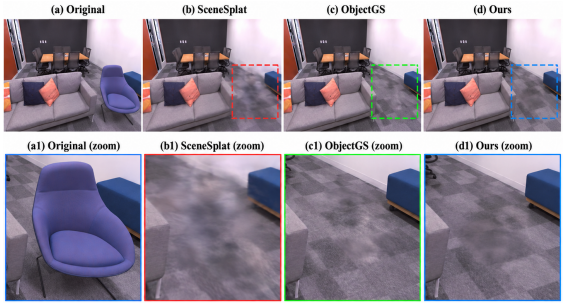



Figure 7: Chair removal comparison with enlarged views of the edited region.

Table 5: Overall ratings on a five-point scale.

| Method | Mean ± SD |
|---|---|
| MonoVoc | **4.11** ± 0.28 |
| ObjectGS | 3.94 ± 0.34 |
| SceneSplat | 3.80 ± 0.32 |

Table 6: Overall participant preferences.

| Method | Votes | Share |
|---|---|---|
| MonoVoc | **18/34** | **52.9%** |
| ObjectGS | 9/34 | 26.5% |
| SceneSplat | 6/34 | 17.6% |
| No clear preference | 1/34 | 2.9% |

### 4.4 Human Evaluation

We asked 34 participants to rate scene structure, object distinguishability, cross-view consistency, visual clarity, and usefulness on five-point scales. MonoVoc received the highest overall mean (4.11, versus 3.94 for ObjectGS and 3.80 for SceneSplat) and 52.9% of preference votes (Tables 5 and 6). Appendix K provides all the details involved including results.

These results support compact semantic mapping with competitive task performance and favorable human ratings. Appendix C further discusses semantic color recovery versus dense per-Gaussian embeddings.

## 5 Discussion

MonoVoc builds on a fixed, modified HI-SLAM2 backbone while supporting compatible segmenters and customizable embedding models or procedures. Segmentation noise remains a limitation (Appendix L). We evaluate alternative embeddings and recommend our proposed configuration as the reference. Semantic inference uses a confidence-guided approximation to account for overlapping Gaussians without solving the full global problem. Organizing open-vocabulary understanding into object removal, object retrieval, question answering, and segmentation enables separate assessment of complementary capabilities. The framework also supports future investigation of richer semantic and spatial embeddings, improved reconstruction and rendering, applying our evaluation protocol to new methods, and adaptation of individual modules to other pipelines through targeted ablations.

## 6 Conclusion

We presented MonoVoc, a framework for organizing monocular 3D Gaussian maps into compact, interpretable objects connected to language. It brings together semantic mapping, spatial context, and object-level interaction, supporting retrieval, question answering, removal, and segmentation while balancing scene quality with computational efficiency.

Our findings support object-centered representations as a practical basis for accessible 3D scene understanding. Reliable segmentation remains important, particularly near ambiguous boundaries. MonoVoc's modular design and evaluation across complementary tasks provide a foundation for improving semantic reliability and extending language-guided interaction with reconstructed environments.

### AI USE STATEMENT

In this work, we used large language models solely for writing, editing, and paraphrasing assistance to improve the clarity and readability of the manuscript, and generative AI to create conceptual illustrations. These illustrations explain ideas and mechanisms; they do not represent experimental outputs or measured results. AI assistance in manuscript preparation was not used to develop the proposed methodology, design or conduct experiments, or formulate scientific claims.

Their use did not alter the research contributions or originality of the work. We reviewed all AI-assisted text for accuracy and consistency with our methodology and experimental findings, revising it where necessary to preserve the intended technical meaning. We take full responsibility for the final content of this paper, including all AI-assisted text, scientific claims, and reported results.

### ETHICS STATEMENT

Our human evaluation compared three methods to assess visual qualities of reconstructed 3D Gaussian maps that may not be fully captured by numerical metrics. Participants consented to participate and were informed that their responses would be used for the paper's evaluation. They received a brief introduction to the relevant concepts and reconstructed spaces. Method identities were concealed using anonymous labels A, B, and C. Output presentation order was randomized across question sets and participants. The evaluation took place in a closed session without external lookups. Names were the only identifying information collected and were stored in a separate spreadsheet accessible only to the research team. No email addresses or recordings were collected, and results are reported anonymously. The study protocol, including participant consent and privacy safeguards, was reviewed and approved by the supervising researcher.

### REPRODUCIBILITY STATEMENT

The main paper describes the MonoVoc pipeline, experimental setup, and evaluation on the public Replica and ScanNet datasets. The appendix provides additional methodological details and evaluation results. Appendix F documents the quantization and ASPT experiments, including the custom quantization variants (Appendix F.4), rendered-image evaluation (Appendix F.6), and internal diagnostics (Appendix F.7). A standalone pseudo implementation of Confidence-Gated Quantization is also provided in the appendix E to make the algorithm's implementation explicit. We will publicly release the source code, including implementations of the proposed methods, upon acceptance, organized into readable, modular files to facilitate reproduction and reuse.

### ACKNOWLEDGMENTS

## REFERENCES

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2425–2433, 2015.

Kangjie Chen, BingQuan Dai, Minghan Qin, Dongbin Zhang, Peihao Li, Yingshuang Zou, and Haoqian Wang. Slgaussian: Fast language gaussian splatting in sparse views. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 3047–3056, 2025.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

Lee R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3): 297–302, 1945. doi: 10.2307/1932409.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031, 2021. doi: 10.1162/tacl_a_00410.

Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. doi: 10.1007/s11263-009-0275-4.

Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5021–5028. IEEE, 2024.

Alain Hore and Djemel Ziou. Image quality metrics: PSNR vs. SSIM. *Proceedings of the 20th International Conference on Pattern Recognition (ICPR)*, pp. 2366–2369, 2010. doi: 10.1109/ICPR.2010.579.

Shunya Kato, Shuhei Kurita, Chenhui Chu, and Sadao Kurohashi. ARKitSceneRefer: Text-based localization of small objects in diverse real-world 3d indoor scenes. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 784–799. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.56.

Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat, track & map 3d gaussians for dense rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023. doi: 10.1145/3592433.

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*, pp. 19729–19739, 2023.

Boyi Li, Kilian Q. Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations*, 2022.

Hao Li, Roy Qin, Zhengyu Zou, Diqi He, Bohan Li, Bingquan Dai, Dingewn Zhang, and Junwei Han. Langsurf: Language-embedded surface gaussians for 3d scene understanding. *arXiv preprint arXiv:2412.17635*, 2024a. URL https://arxiv.org/abs/2412.17635.

Mingrui Li, Shuhong Liu, Heng Zhou, Guohao Zhu, Na Cheng, Tianchen Deng, and Hongyu Wang. Sgs-slam: Semantic gaussian splatting for neural dense slam. In *European Conference on Computer Vision*, pp. 163–179. Springer, 2024b.

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 (NeurIPS)*, 2025a. URL https://arxiv.org/abs/2507.07136.

Wanhua Li, Renping Zhou, Jiawei Zhou, Yingwei Song, Johannes Herter, Minghan Qin, Gao Huang, and Hanspeter Pfister. 4D langsplat: 4D language Gaussian splatting via multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22001–22011, June 2025b.

Yue Li, Qi Ma, Runyi Yang, Huapeng Li, Mengjiao Ma, Bin Ren, Nikola Popovic, Nicu Sebe, Ender Konukoglu, Theo Gevers, Luc Van Gool, Martin R. Oswald, and Danda Pani Paudel. Scenesplat: Gaussian splatting-based scene understanding with vision-language pretraining. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4961–4972, October 2025c.

Juliette Marrie, Romain Menegaux, Michael Arbel, Diane Larlus, and Julien Mairal. Ludvig: Learning-free uplifting of 2d visual features to gaussian splatting scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Tomas Berriel Martins, Martin R. Oswald, and Javier Civera. Open-vocabulary online semantic mapping for slam. *IEEE Robotics and Automation Letters*, 2025.

Hidenobu Matsuki, Riku Murai, Paul H. J. Kelly, and Andrew J. Davison. Gaussian splatting slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pp. 405–421. Springer, 2020.

Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. *arXiv preprint arXiv:2312.16084*, 2023.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 3982–3992. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1410.

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*, pp. 5333–5343, 2024.

Yale Song and Mohammad Soleymani. Polysemous visual-semantic embedding for cross-modal retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1979–1988, 2019. doi: 10.1109/CVPR.2019.00208.

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. URL `https://arxiv.org/abs/1906.05797`.

Laszlo Szilagyi, Francis Engelmann, and Jeannette Bohg. Slag: Scalable language-augmented gaussian splatting. *IEEE Robotics and Automation Letters*, 2025.

Osama Tasneem and Roel Pieters. Human–robot collaborative visual inspection with large language models. *Robotics and Computer-Integrated Manufacturing*, 98:103154, 2026. doi: 10.1016/j.rcim.2025.103154.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. doi: 10.1109/TIP.2003.819861.

Abdelrhman Werby, Chenguang Huang, Martin Büchner, Abhinav Valada, and Wolfram Burgard. Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation. In *Robotics: Science and Systems (RSS)*, 2024a.

Abdelrhman Werby, Chenguang Huang, Martin Büchner, Abhinav Valada, and Wolfram Burgard. Hierarchical Open-Vocabulary 3D Scene Graphs for Language-Grounded Robot Navigation. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024b. doi: 10.15607/RSS.2024.XX.077.

Cort J. Willmott and Kenji Matsuura. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, 30:79–82, 2005. doi: 10.3354/cr030079.

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. In *Advances in Neural Information Processing Systems*, volume 37, pp. 19114–19138. Curran Associates, Inc., 2024. doi: 10.52202/079017-0604.

Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *ECCV*, 2024.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018. doi: 10.1109/CVPR.2018.00068.

Wei Zhang, Qing Cheng, David Skuddis, Niclas Zeller, Daniel Cremers, and Norbert Haala. HI-SLAM2: Geometry-aware gaussian SLAM for fast monocular scene reconstruction. *IEEE Transactions on Robotics*, 41:6478–6493, October 2025. doi: 10.1109/TRO.2025.3626627. URL https://doi.org/10.1109/TRO.2025.3626627. Originally available as arXiv:2411.17982.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

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 (ICCV)*, pp. 8350–8360, October 2025.

## A SEMANTIC OVERLAP IN GAUSSIAN RENDERING

Semantic overlap arises because projected Gaussian footprints have spatial extent and contribute jointly to rendered pixels. Near an object boundary, a footprint can influence pixels on both sides, even when its center lies within one object. Figure 8 illustrates typical overlap configurations.

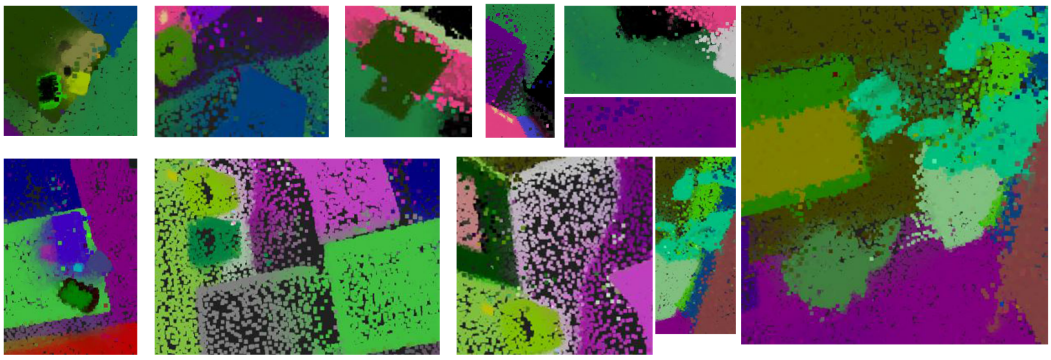


Figure 8: Typical configurations in which projected Gaussian footprints overlap across semantic regions.

For fixed geometry and opacity, the background-subtracted semantic observation at pixel $p$ is

$$\boldsymbol{y}_p = \sum_{i=1}^{N} w_{i,p} \boldsymbol{s}_i, \tag{8}$$

where $\boldsymbol{y}_p, \boldsymbol{s}_i \in \mathbb{R}^3$ denote the observed pixel color and Gaussian semantic color, respectively, and $w_{i,p} \geq 0$ is the alpha-compositing weight. A rendered color therefore generally reflects several Gaussian contributions rather than an individual semantic assignment, as illustrated in Figure 9.

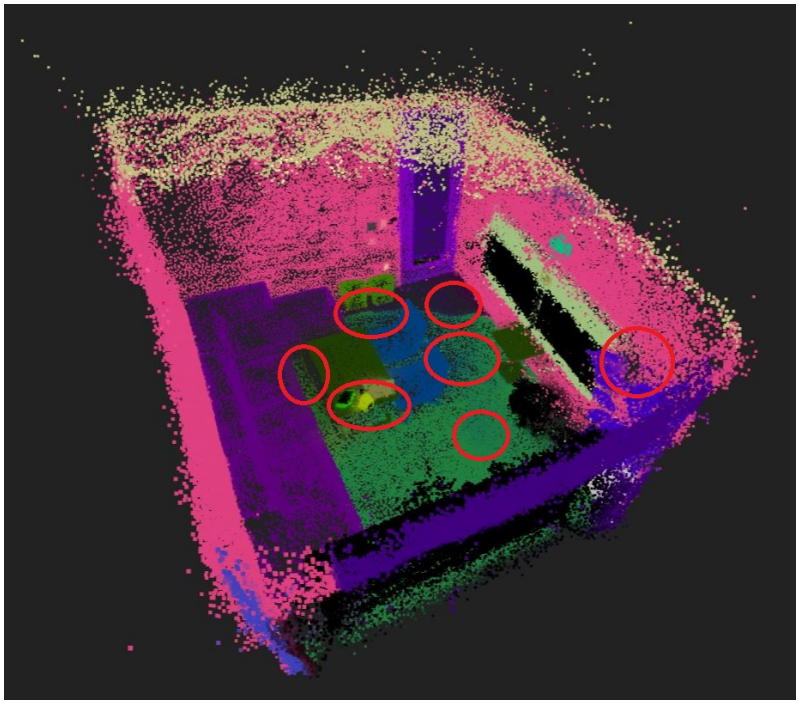


Figure 9: Illustration of semantic overlap during deblending. Shared contributions to rendered pixels complicate the recovery of individual Gaussian colors.

**Coupled semantic recovery.** Stacking $M$ pixel observations from the input video gives $\boldsymbol{Y} = \boldsymbol{W}\boldsymbol{S}$, with $\boldsymbol{Y} \in \mathbb{R}^{M\times 3}$, $\boldsymbol{W} \in \mathbb{R}^{M\times N}$, and $\boldsymbol{S} \in \mathbb{R}^{N\times 3}$. A global least-squares formulation would couple Gaussian colors through the normal equations:

$$\boldsymbol{W}^\top \boldsymbol{W}\boldsymbol{S} = \boldsymbol{W}^\top \boldsymbol{Y}, \qquad \left(\boldsymbol{W}^\top \boldsymbol{W}\right)_{i,j} = \sum_{p=1}^{M} w_{i,p} w_{j,p}. \tag{9}$$

The off-diagonal terms describe shared influence on the same observations. This formulation explains the coupling; it is not the update rule of our approximate SCD procedure. Independently aggregating observed colors can mix evidence from neighboring semantic regions, and weak contributions can accumulate over many pixels and frames from the same video.

**Complementary roles of ASPT and quantization.** ASPT increases separation between valid semantic colors before reconstruction, making them less confusable during recovery. It does not change the soft support of Gaussian footprints or eliminate their overlap. Confidence-Gated Quantization subsequently combines color evidence, confidence estimates, and spatial relationships to guide discrete semantic assignment. ASPT thus improves the input encoding, while quantization resolves assignments using information beyond color proximity alone.

## B Color-Confusable Palettes and ASPT

**Motivating experiment.** The color-confusable setting assigns distinct but visually similar segmentation colors to different objects or object parts. Figure 10 illustrates these inputs, and Figure 11 shows a corresponding semantic rendering. These figures illustrate the motivating setting rather than a quantitative comparison of the current modules.

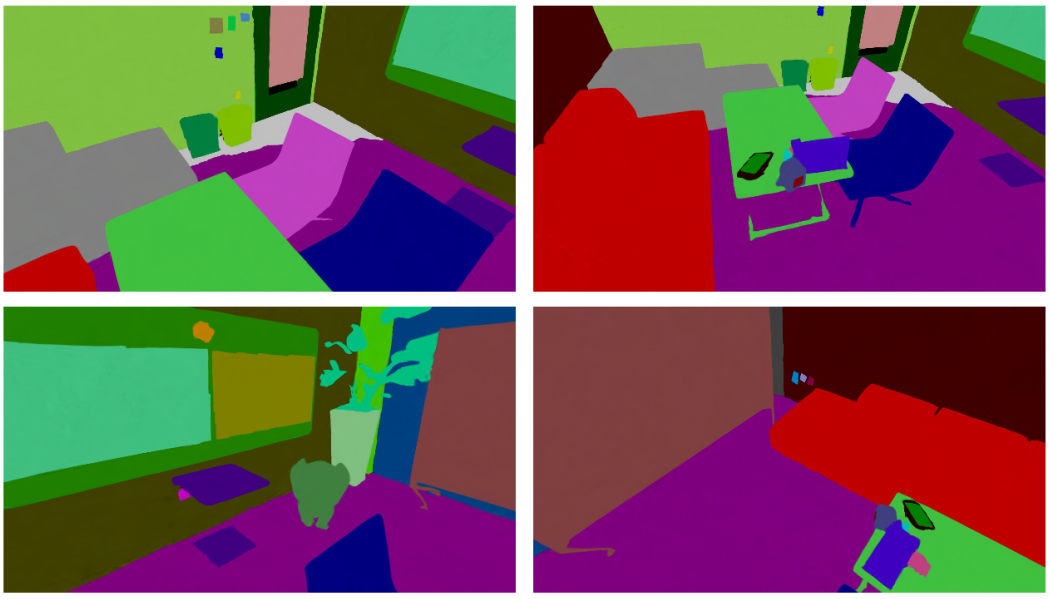


Figure 10: Color-confusable segmentation inputs. Distinct objects or object parts receive similar colors, reducing separation between their semantic identifiers.

When valid palette entries are close, small changes in recovered color can make the associated identity ambiguous, particularly where Gaussian contributions overlap. Appendix F evaluates ASPT's effect on semantic rendering error. The transform changes only the palette encoding, retaining the supplied segmentation regions and their label correspondence. Confidence-Gated Quantization complements this transformation by incorporating confidence and spatial evidence when assigning Gaussian semantics.

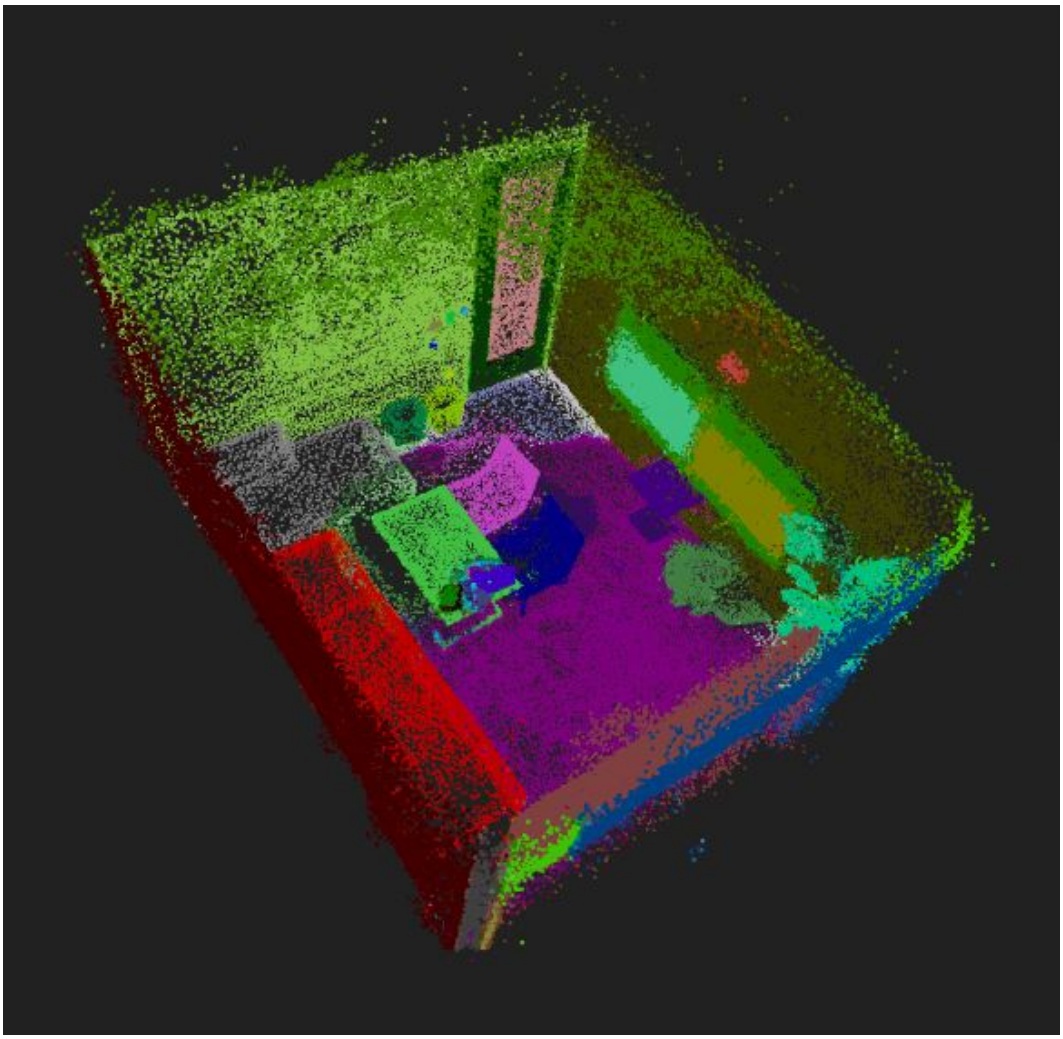


Figure 11: Example semantic Gaussian rendering for the color-confusable setting, illustrating the challenge of recovering distinct object assignments from similar palette colors.

**Palette construction.** ASPT extracts the discrete RGB palette from the input segmentation images or loads a supplied palette. It constructs candidate colors from an RGB grid supplemented with randomly sampled colors. User-designated colors can remain fixed and participate in the separation criterion. Without fixed colors, the procedure selects an initial seed, favoring high chroma and intermediate lightness when Lab conversion is available. It then applies Equation 1 until the required palette size is reached. Optional refinement replaces individual mutable colors when doing so improves their minimum normalized dissimilarity to the remaining palette. This greedy construction with local refinement does not guarantee a globally optimal palette.

**Dissimilarities and label preservation.** The supplied implementation defaults to RGB, CIELAB, CMC, CIEDE2000, and OKLab comparisons, with normalization scales $255\sqrt{3}$, $150$, $100$, $100$, and $0.70$, respectively; the default CMC parameters are $l : c = 2 : 1$. These comparisons govern palette selection and do not specify the downstream quantization rule. After selection, replacement colors are matched one-to-one to the original palette, using Lab-based assignment by default. The same mapping is applied to every frame, preserving existing label distinctions without introducing new object identities.

**Pairwise separation and mixture ambiguity.** ASPT controls pairwise palette separation, but overlapping Gaussian contributions produce convex or sub-convex mixtures of semantic colors

(Equation 108). Pairwise separation does not guarantee separation between a palette entry and mixtures of other entries: even a two-color mixture can approach an unrelated label, and higher-order overlap introduces further ambiguity. Confidence-Gated Quantization uses confidence and spatial evidence to mitigate these residual ambiguities, without guaranteeing their elimination. Explicit separation from mixture segments or convex hulls is a possible refinement beyond the current ASPT objective.

## C Semantic Colors versus Per-Gaussian Embeddings

**Separating assignment from language representation.** Semantic colors encode segmentation identities, while language embeddings describe the associated objects. Our reconstruction uses three semantic coefficients per Gaussian, supervised through rendered segmentation maps. Because alpha compositing mixes these coefficients, they need not coincide with valid palette entries. SCD recovers color evidence, and Confidence-Gated Quantization combines this evidence with confidence and spatial information to obtain discrete assignments. Each assigned object then shares one language embedding across its Gaussians.

**Alternative identity representations.** Three-channel encoding is a compact design choice, not a requirement for object-level language storage. Learned identity features, such as those used by Gaussian Grouping (Ye et al., 2024), and explicit $K$-dimensional label logits provide alternatives. Compact identity features can also have a fixed dimension independent of $K$, while explicit logits allocate a separate channel to each label. Both could subsequently share language embeddings by object. Our choice retains three-channel segmentation supervision and direct palette visualization, at the cost of color ambiguity that motivates ASPT, SCD, and quantization. We do not claim that RGB encoding is inherently more accurate than these alternatives. In our reported comparisons, Gaussian Grouping obtains lower rendering and segmentation scores than MonoVoc on both datasets (Table 1). These results support the quality of our complete configuration, although differences in reconstruction and semantic supervision prevent attributing the gap solely to the identity representation. Compact learned identity features therefore remain a relevant alternative, without demonstrating an advantage in our evaluated settings.

**Storage and computation as object count increases.** For $N$ Gaussians, $K$ objects, and language dimension $D$, dense per-Gaussian language vectors require $\mathcal{O}(ND)$ storage. Our representation requires $\mathcal{O}(N + KD)$ for constant-sized Gaussian semantics, object identifiers, and shared embeddings. Explicitly storing $K$ logits per Gaussian instead requires $\mathcal{O}(NK)$ assignment storage; fixed-dimensional identity features require $\mathcal{O}(Nd_{\mathrm{id}})$, excluding decoder parameters. These comparisons concern explicit Gaussian representations, not the parameter count of an implicit neural field. Although our per-Gaussian semantic dimension stays fixed, evaluating every Gaussian against all palette entries requires $\mathcal{O}(NK)$ color comparisons. Processing $B$ Gaussians at a time limits the corresponding distance buffer to $\mathcal{O}(BK)$.

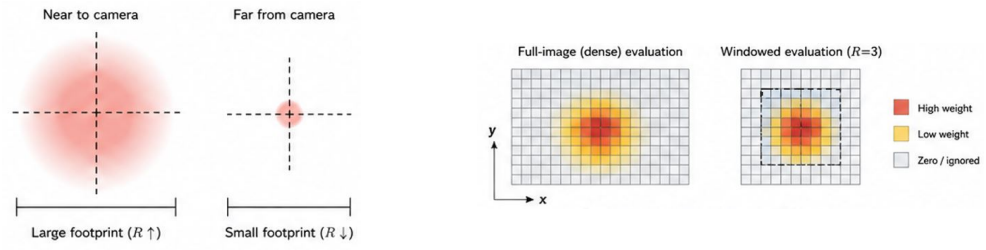
**Palette separation as object count increases.** Let $\mathcal{P}_K = \{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$ contain normalized RGB colors in $[0, 1]^3$, and define

$$d_{\min}(\mathcal{P}_K) = \min_{j \neq k} \|\mathbf{c}_j - \mathbf{c}_k\|_2. \tag{10}$$

For $K \geq 2$, partitioning the RGB cube into $m^3 < K$ equal subcubes places at least two colors in one subcube. Choosing $m = \lfloor (K-1)^{1/3} \rfloor$ therefore gives

$$d_{\min}(\mathcal{P}_K) \leq \frac{\sqrt{3}}{\lfloor (K-1)^{1/3} \rfloor} = \mathcal{O}(K^{-1/3}). \tag{11}$$

Thus no palette construction can maintain a fixed positive minimum RGB separation for arbitrarily many labels. For Euclidean nearest-palette assignment, recovery error below $d_{\min}/2$ is sufficient to preserve a generating label. The decreasing separation bound therefore constrains the attainable worst-case robustness margin. This argument characterizes RGB geometry; it is not a quantitative CIEDE2000 accuracy bound or a prediction of measured mIoU.




(a) Projected footprint.

(b) Windowed evaluation.

Figure 12: Motivation for EWR. (a) For a fixed Gaussian size, proximity to the camera increases its projected footprint. (b) Windowed evaluation concentrates computation around the projected center, omitting contributions outside the selected window.

**Mixture ambiguity and practical scope.** Pairwise separation alone does not prevent a mixture of several palette colors from approaching another valid entry. Under the compositing model, recovered evidence can lie between semantic colors, with additional attenuation from residual transparency (Appendix M). ASPT improves pairwise separation but does not guarantee separation from these mixtures. Confidence and spatial regularization provide additional evidence for assignment; the optional chroma guard addresses a specific neutral-color failure mode rather than general palette crowding. Actual accuracy depends on local overlap, observation quality, and palette geometry as well as $K$. Our experiments do not establish accuracy scaling across object counts. The analysis therefore identifies a tradeoff: fixed per-Gaussian channel count and shared language storage, with increasing palette comparison cost and potentially reduced color robustness as the number of labels grows.

## D Windowed Evaluation for Semantic Color Deblending

Figure 12 illustrates how camera distance affects footprint size and how spatial windows restrict evaluation to the Gaussian's effective support.

**Gaussian footprints.** A projected Gaussian centered at $\boldsymbol{\mu} \in \mathbb{R}^2$ with covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{2\times 2}$ has spatial kernel

$$g(\boldsymbol{x}) = \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right). \tag{12}$$

Its contribution decreases with distance from the projected center. Evaluating every Gaussian across the full image therefore spends computation on many weak contributions. EWR uses a square window based on three projected coordinate standard deviations to concentrate SCD evaluation around each footprint. Windowing changes the spatial support evaluated while retaining the SCD weighting and accumulation procedure.

**Estimating and capping the radius.** EWR constructs each Gaussian's 3D covariance from its scale and rotation, transforms it into camera coordinates, and projects it into image space using the camera intrinsics. Positive-depth and image-overlap tests identify eligible Gaussian–frame pairs in sampled frames. Their radii form the empirical distribution used in Equation 2. Large projected footprints, including those produced by Gaussians close to the camera, can otherwise dominate window allocation. The percentile-based cap controls this cost while retaining the estimated footprint radius for most sampled pairs. The percentile describes footprint coverage in the sampled distribution, rather than a bound on the discarded semantic contribution.

**Batching and memory.** Visible Gaussians are sorted by radius so that similarly sized footprints are processed together. Each batch uses the largest capped radius among its members, avoiding excessive padding of small footprints alongside much larger ones. For a batch of $B$ Gaussians with allocated radius $R_b$, the dominant temporary window storage is approximately

$$M_{\text{window}} \approx B(2R_b+1)^2 b, \qquad b \approx 56 \text{ bytes}, \tag{13}$$

where $b$ is the implementation-specific storage per Gaussian–pixel cell. Dense evaluation instead allocates corresponding storage proportional to $BHW$ for an image of height $H$ and width $W$. The window model describes the dominant temporary allocation; total peak VRAM also includes the Gaussian map and other buffers.

**Agreement with dense evaluation.** The supplied dense–windowed comparison reports a maximum output difference of approximately $10^{-11}$, indicating close numerical agreement for the evaluated run. This measures agreement with the particular reference implementation used; it does not establish equivalence to evaluation of untruncated Gaussian footprints. Interpreting this difference requires distinguishing support truncation shared by both implementations from the additional restriction introduced by EWR and its percentile-based radius cap.

# E Confidence-Gated Quantization Algorithm

The procedure is divided into preparation (Algorithm 1) and label optimization (Algorithm 2). Preparation computes combined confidence from Equation 7 using the initial labels $z_i^{(0)}$. These confidence scores remain fixed throughout optimization. Algorithm 2 uses sequential ICM with in-place updates and tie preservation; it does not use simultaneous label updates. The margin confidence $q_i$ enters only through the combined score $C_i$.

Gaussians with zero accumulated support $W_i = \sum_{p \in \mathcal{W}_i} w_{i,p} = 0$ retain their geometry but receive $z_i = \perp$ and $C_i = 0$. Their recovered colors remain undefined, and they are excluded from palette assignment, label updates, and semantic rendering. Confidence normalization statistics are computed only over supported Gaussians.

Normal signs are propagated along a maximum spanning forest, prioritizing absolute normal agreement. This ensures nonnegative agreement on forest edges while retaining the signed bilateral weighting in Equation 31; agreement on every remaining graph edge is not guaranteed.

Here $\mathcal{N}_+(i)$ denotes neighbors in $\mathcal{E}_+$, and $N$ is the total Gaussian count. Algorithm 2 is applied even when preparation returns early: an empty supported graph leaves initial or unassigned labels unchanged. Supported Gaussians with $C_i = 0$ retain their initial labels through the tie-preserving update rule. Stored confidence scores describe the initial evidence used for quantization and are not recomputed from the final assignments.

# F Ablation Study on Quantization Algorithms and the Effect of ASPT

This study examines how continuous semantic colors recovered by SCD should be converted into discrete object assignments, and how ASPT affects this conversion. We first compare independent nearest-palette assignment under several color dissimilarities, then evaluate six custom spatial quantization variants. We additionally examine a chroma guard motivated by erroneous assignments to neutral palette entries. Evaluations cover semantic and instance segmentation, each with and without ASPT, using shared rendered-image error measures and complementary diagnostics of the Gaussian label structure.

## F.1 Experimental Design

The four settings are instance segmentation with ASPT, instance segmentation without ASPT, semantic segmentation with ASPT, and semantic segmentation without ASPT. Each setting provides an SCD Gaussian map and its corresponding segmentation palette. Within a setting, the custom methods use the same input Gaussians and base color costs for a given distance metric. Graph-based diagnostics are evaluated on a common graph, including when a method uses a different graph internally.

The custom sweep contains 608 configurations: 152 per setting. For each setting, four distance choices and two chroma-guard modes produce eight nearest-palette baselines and $4 \times 2 \times 3 \times 6 = 144$

**Algorithm 1** Confidence-Gated Quantization: Preparation

**Require:** Centers $\boldsymbol{x}_i$, covariances $\boldsymbol{\Sigma}_i$, opacities $o_i$, supports $W_i$; recovered colors $\hat{\boldsymbol{s}}_i$ for $W_i > 0$; palette $\{\boldsymbol{c}_k\}_{k=1}^{K}$ with $K \geq 2$; segmentation observations and SCD weights
**Ensure:** Supported set $\mathcal{I}_+$, labels $z_i$, fixed confidences $C_i$, unary costs $U_i(k)$, margins $\Delta_i$, graph $\mathcal{E}_+$, and oriented normals $\boldsymbol{n}_i$

$\mathcal{I}_+ \leftarrow \{i : W_i > 0\}$
Set $z_i \leftarrow \perp$ and $C_i \leftarrow 0$ for $i \notin \mathcal{I}_+$
$\mathcal{E}_+ \leftarrow \varnothing$
**if** $\mathcal{I}_+ = \varnothing$ **then**
    **return**
**end if**
Compute CIEDE2000 costs $D_i(k)$ for $i \in \mathcal{I}_+$
$z_i^{(0)} \leftarrow \arg\min_k D_i(k)$; $z_i \leftarrow z_i^{(0)}$
Compute $\Delta_i$, $\tau$, and $q_i$ using Equation 6, with statistics over $\mathcal{I}_+$
Compute $C_i$ using Equation 7, with $z_i^{(0)}$ in the vote term
$U_i(k) \leftarrow C_i o_i D_i(k)$; hold $C_i$ fixed
Build unique undirected edges $\mathcal{E}$ from 24-nearest-neighbor relations between all centers ▷ Use all available neighbors if fewer exist
$\mathcal{E}_+ \leftarrow \{(i,j) \in \mathcal{E} : i, j \in \mathcal{I}_+\}$
**if** $\mathcal{E}_+ = \varnothing$ **then**
    **return**
**end if**
Obtain unit normals $\tilde{\boldsymbol{n}}_i$ from stored normals, or the smallest-eigenvalue covariance eigenvectors
$\boldsymbol{n}_i \leftarrow \tilde{\boldsymbol{n}}_i$
Construct a maximum spanning forest of $\mathcal{E}_+$ with priorities $|\tilde{\boldsymbol{n}}_i^\top \tilde{\boldsymbol{n}}_j|$
**for all** trees in the forest **do**
    Choose the smallest-index vertex as root
    **for all** non-root vertices $j$ in parent-before-child order **do**
        Let $i$ be the parent of $j$
        **if** $\boldsymbol{n}_i^\top \boldsymbol{n}_j < 0$ **then**
            $\boldsymbol{n}_j \leftarrow -\boldsymbol{n}_j$
        **end if**
    **end for**
**end for**

Table 7: Configuration grid for the custom quantization study. Defaults shown here describe the sweep implementation.

| Factor | Values |
|---|---|
| Segmentation setting | Semantic, instance |
| ASPT | Enabled, disabled |
| Unary dissimilarity | RGB, Lab, CMC, CIEDE2000 |
| Custom method | A, B, C, D, E, F |
| Regularization multiplier $\beta$ | $1, 2, 4$ |
| Chroma guard | Enabled, disabled |
| Base neighborhood size | 24 |
| Local-scale neighborhood size | 8 |
| Adaptive-distance cutoff | 2 |
| Gaussian/palette chroma thresholds | $12/12$ |
| Sequential ICM sweeps | Up to 20 |
| Alpha-expansion sweeps | Up to 2 |

spatial-method configurations. The historical comparison additionally contains 15 nearest-color variants per setting, giving 60 legacy results.

The sweep uses sequential, in-place ICM for methods A–D. Option B uses the combined confidence $C_i$ from Equation 7, computed against the initial labels and held fixed throughout optimization, as in Appendix E. The comparisons evaluate this combined gating configuration against the other quantization variants; they do not separately isolate the contributions of its margin, vote, and support factors.

**Algorithm 2** Confidence-Gated Quantization: Optimization

**Require:** Outputs of Algorithm 1; centers $\boldsymbol{x}_i$, covariances $\boldsymbol{\Sigma}_i$, opacities $o_i$, supported recovered colors $\hat{\boldsymbol{s}}_i$, palette $\{\boldsymbol{c}_k\}$; optional supplied $\lambda$
**Ensure:** Final labels $z_i$, fixed confidences $C_i$, and palette colors for assigned Gaussians

**if** $\mathcal{E}_+ \neq \varnothing$ **then**
  **for all** $(i,j) \in \mathcal{E}_+$ **do**
    $\eta_{ij} \leftarrow 10^{-9}\,\mathrm{tr}(\boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j) + \epsilon$
    $\delta_{ij}^2 \leftarrow (\boldsymbol{x}_i - \boldsymbol{x}_j)^\top (\boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j + \eta_{ij}\boldsymbol{I})^{-1}(\boldsymbol{x}_i - \boldsymbol{x}_j)$
  **end for**
  $\sigma_x^2 \leftarrow \mathrm{median}_{(i,j)\in\mathcal{E}_+}\,\delta_{ij}^2 + \epsilon$; $\sigma_n \leftarrow 0.3$
  For $(i,j) \in \mathcal{E}_+$, set $a_{ij} \leftarrow \exp\!\left(-\dfrac{\delta_{ij}^2}{2\sigma_x^2+\epsilon} - \dfrac{1-\mathrm{clip}(\boldsymbol{n}_i^\top \boldsymbol{n}_j, -1, 1)}{2\sigma_n^2+\epsilon}\right)$
  **if** optional color weighting is enabled **then**
    $h_{ij} \leftarrow \|\,\mathrm{Lab}(\hat{\boldsymbol{s}}_i) - \mathrm{Lab}(\hat{\boldsymbol{s}}_j)\|_2^2$ for $(i,j) \in \mathcal{E}_+$
    $\sigma_c^2 \leftarrow \mathrm{median}_{(i,j)\in\mathcal{E}_+}\,h_{ij} + \epsilon$
    $a_{ij} \leftarrow a_{ij}\exp\!\big(-h_{ij}/(2\sigma_c^2+\epsilon)\big)$
  **end if**
  $\tilde{a}_{ij} \leftarrow a_{ij}\sqrt{C_i C_j}$ for $(i,j) \in \mathcal{E}_+$
  Unless supplied, set $\lambda \leftarrow \dfrac{2\,\mathrm{median}_{i\in\mathcal{I}_+}(C_i o_i \Delta_i)}{\max\!\left(\frac{2}{N}\sum_{(i,j)\in\mathcal{E}_+}\tilde{a}_{ij}, \epsilon\right)}$
  **for** $t = 1, \ldots, 20$ **do**
    Shuffle the supported Gaussian update order
    **for all** $i \in \mathcal{I}_+$ in that order **do**
      $z_i \leftarrow \arg\min_k \left[U_i(k) + \lambda \sum_{j\in\mathcal{N}_+(i)} \tilde{a}_{ij}\,\mathbf{1}_{k\neq z_j}\right]$ ▷ Update in place; retain current label on ties
    **end for**
    **if** no labels changed **then**
      **break**
    **end if**
  **end for**
**end if**
Store $z_i$ and $C_i$ for every Gaussian, and $\boldsymbol{c}_{z_i}$ only for $i \in \mathcal{I}_+$

### F.2 Independent Nearest-Palette Quantization

Let $\hat{\mathbf{s}}_i$ denote Gaussian $i$'s recovered semantic color and let $\{\mathbf{c}_k\}_{k=1}^K$ be the valid palette. For a color dissimilarity $d$, independent quantization assigns

$$D_i(k) = d(\hat{\mathbf{s}}_i, \mathbf{c}_k), \qquad z_i^{(0)} \in \arg\min_k D_i(k), \qquad \mathbf{s}_i^{\mathrm{quant}} = \mathbf{c}_{z_i^{(0)}}. \tag{14}$$

Every Gaussian is processed independently. This minimizes the sum of unary costs, but introduces no preference for spatial agreement or preservation of coherent object regions. The assignment is computed in batches by evaluating a Gaussian-to-palette distance matrix and selecting each row's minimum.

**RGB and Lab distances.** RGB quantization uses Euclidean distance in the input RGB coordinates:

$$d_{\mathrm{RGB}}(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\|_2. \tag{15}$$

The shared custom-method implementation converts normalized RGB values to floating-point CIELAB and uses

$$d_{\mathrm{Lab}}(\mathbf{u}, \mathbf{v}) = \|\,\mathrm{Lab}(\mathbf{u}) - \mathrm{Lab}(\mathbf{v})\|_2. \tag{16}$$

The earlier nearest-color script instead computes Euclidean distance in OpenCV's eight-bit Lab encoding. Approximately, this encoding represents $(L^*, a^*, b^*)$ as $(255L^*/100, a^*+128, b^*+128)$, with integer quantization. Consequently, its distance is

$$d_{\mathrm{Lab8}}(\mathbf{u}, \mathbf{v}) = \|\,\mathrm{Lab8}(\mathbf{u}) - \mathrm{Lab8}(\mathbf{v})\|_2, \tag{17}$$

which weights lightness differently from floating-point CIELAB. These two implementations therefore should not be treated as identical Lab baselines.

**Lightness–chroma–hue differences.** For two Lab colors $\boldsymbol{\ell}_1 = (L_1, a_1, b_1)$ and $\boldsymbol{\ell}_2 = (L_2, a_2, b_2)$, define

$$\begin{aligned} C_r &= \sqrt{a_r^2 + b_r^2}, \qquad r \in \{1, 2\}, \\ \Delta L &= L_1 - L_2, \qquad \Delta C = C_1 - C_2, \\ \Delta H &= \sqrt{\max((a_1 - a_2)^2 + (b_1 - b_2)^2 - \Delta C^2, 0)}. \end{aligned} \tag{18}$$

CMC and CIE94 rescale these components using color-dependent weighting functions.

**CMC.** The CMC implementation uses

$$d^2_{\mathrm{CMC}} = \left(\frac{\Delta L}{l S_L}\right)^2 + \left(\frac{\Delta C}{c S_C}\right)^2 + \left(\frac{\Delta H}{S_H}\right)^2, \tag{19}$$

with default $l = 2$ and $c = 1$. For the reference color $(L_r, a_r, b_r)$, let $C_r = \sqrt{a_r^2 + b_r^2}$ and $h_r = \mathrm{atan2}(b_r, a_r)$ expressed in degrees in $[0, 360)$. The weighting functions are

$$\begin{aligned} S_L &= \begin{cases} 0.511, & L_r < 16, \\ \dfrac{0.040975 L_r}{1 + 0.01765 L_r}, & L_r \geq 16, \end{cases} \\ S_C &= \frac{0.0638 C_r}{1 + 0.0131 C_r} + 0.638, \\ F &= \sqrt{\frac{C_r^4}{C_r^4 + 1900}}, \\ T &= \begin{cases} 0.56 + |0.2 \cos(h_r + 168^\circ)|, & 164^\circ \leq h_r \leq 345^\circ, \\ 0.36 + |0.4 \cos(h_r + 35^\circ)|, & \text{otherwise}, \end{cases} \\ S_H &= S_C(FT + 1 - F). \end{aligned} \tag{20}$$

The historical nearest-color script uses the palette color as the reference, whereas the shared custom-method utilities use the Gaussian color. Since these weights depend on the reference, the two CMC implementations need not produce identical assignments.

**CIE94.** CIE94 uses

$$d^2_{94} = \left(\frac{\Delta L}{k_L}\right)^2 + \left(\frac{\Delta C}{k_C(1 + K_1 C_1)}\right)^2 + \left(\frac{\Delta H}{k_H(1 + K_2 C_1)}\right)^2. \tag{21}$$

In the supplied implementation, $C_1$ is the Gaussian's chroma. The graphic-arts setting uses $(k_L, k_C, k_H, K_1, K_2) = (1, 1, 1, 0.045, 0.015)$; the textile setting changes $(k_L, K_1, K_2)$ to $(2, 0.048, 0.014)$. This option belongs to the historical nearest-color comparison, rather than the four-metric custom sweep.

**CIEDE2000.** CIEDE2000 introduces corrected chroma and hue coordinates and an interaction term between them. For $r \in \{1, 2\}$, define

$$\begin{aligned} \overline{C} &= \frac{C_1 + C_2}{2}, \\ g &= \frac{1}{2}\left(1 - \sqrt{\frac{\overline{C}^{\,7}}{\overline{C}^{\,7} + 25^7}}\right), \\ a'_r &= (1 + g) a_r, \qquad C'_r = \sqrt{(a'_r)^2 + b_r^2}, \\ h'_r &= \mathrm{atan2}(b_r, a'_r) \quad \text{in } [0, 360^\circ). \end{aligned} \tag{22}$$

Let $\Delta h'$ be the wrapped difference $h'_2 - h'_1$ in $[-180^\circ, 180^\circ]$, and let $\overline{h}'$ be the corresponding wrapped mean hue. The evaluated differences are

$$\Delta L' = L_2 - L_1, \qquad \Delta C' = C'_2 - C'_1, \qquad \Delta H' = 2\sqrt{C'_1 C'_2}\sin(\Delta h'/2). \tag{23}$$

With $\overline{L}' = (L_1 + L_2)/2$ and $\overline{C}' = (C_1' + C_2')/2$, the scale factors are

$$\begin{aligned} T &= 1 - 0.17\cos(\overline{h}' - 30^\circ) + 0.24\cos(2\overline{h}') \\ &\quad + 0.32\cos(3\overline{h}' + 6^\circ) - 0.20\cos(4\overline{h}' - 63^\circ), \\ S_L &= 1 + \frac{0.015(\overline{L}' - 50)^2}{\sqrt{20 + (\overline{L}' - 50)^2}}, \\ S_C &= 1 + 0.045\overline{C}', \\ S_H &= 1 + 0.015\overline{C}'T, \\ R_T &= -2\sqrt{\frac{\overline{C}'^7}{\overline{C}'^7 + 25^7}}\sin(2\Delta\theta), \\ \Delta\theta &= 30^\circ \exp\left[-\left(\frac{\overline{h}' - 275^\circ}{25^\circ}\right)^2\right]. \end{aligned} \tag{24}$$

With unit parametric factors, the distance is

$$d_{00} = \sqrt{\left(\frac{\Delta L'}{S_L}\right)^2 + \left(\frac{\Delta C'}{S_C}\right)^2 + \left(\frac{\Delta H'}{S_H}\right)^2 + R_T\frac{\Delta C'}{S_C}\frac{\Delta H'}{S_H}}. \tag{25}$$

The shared implementation includes small numerical stabilizers. The supplied routines use their angle-wrapping branches directly; they do not explicitly implement all standard zero-chroma hue branches. Thus exact implementation equivalence should be checked before treating independently implemented CIEDE2000 baselines as interchangeable.

**DIN99-labeled transforms.** The historical script provides two logarithmic Lab transforms, selected by its `din99` and `din99o` options. Writing $\theta = 16^\circ$, their implemented chromatic transform is

$$\begin{aligned} e &= a\cos\theta + b\sin\theta, \\ f &= 0.7(-a\sin\theta + b\cos\theta), \\ G &= \sqrt{e^2 + f^2}, \qquad h = \operatorname{atan2}(f, e), \\ C_{99} &= \frac{\log(1 + 0.045G)}{0.045}, \\ a_{99} &= C_{99}\cos h, \qquad b_{99} = C_{99}\sin h. \end{aligned} \tag{26}$$

The lightness branches in that script are

$$L_{99} = \begin{cases} 325.22\log(1 + 0.0036L), & \texttt{din99}, \\ 105.509\log(1 + 0.0158L), & \texttt{din99o}. \end{cases} \tag{27}$$

Nearest-color assignment uses Euclidean distance between the resulting $(L_{99}, a_{99}, b_{99})$ coordinates. These equations document the implemented variants; the option names alone do not establish equivalence to the standardized DIN99 and DIN99o transforms.

**Appearance-space and additional historical variants.** The function named `cam16ucs` in the supplied nearest-color script actually requests a CAM02-UCS conversion. Its implemented distance is therefore

$$d_{\text{CAM02-UCS}}(\mathbf{u}, \mathbf{v}) = \|\phi_{\text{CAM02-UCS}}(\mathbf{u}) - \phi_{\text{CAM02-UCS}}(\mathbf{v})\|_2, \tag{28}$$

where $\phi$ is the conversion used by the color-space library.

The historical result log additionally reports CAM16-LCD, CAM16-SCD, CAM16-UCS, OKLab, OKLCH, and variants named `OKLAB_SEG` and `OKLCH_SEG`. We retain these experimental names in the result tables. Their scores are reported independently of the eight-option script described above.

### F.3 Shared Formulation for the Custom Methods

The custom variants combine color evidence with relationships between nearby Gaussians. Let $\mathbf{x}_i$, $\mathbf{\Sigma}_i$, $o_i$, and $\mathbf{n}_i$ denote Gaussian center, covariance, opacity, and normal. The loader decodes stored log-scales by exponentiation and opacity logits by a sigmoid, and constructs

$$\mathbf{\Sigma}_i = \mathbf{R}_i \operatorname{diag}(\boldsymbol{\sigma}_i^2)\mathbf{R}_i^\top. \tag{29}$$

Usable stored normals are normalized; otherwise the smallest covariance eigenvector provides a normal estimate. Before signed normal agreement is evaluated, the sign ambiguity is resolved by the orientation procedure described below and in Appendix E.

**Neighborhood graph and edge weights.** The base graph contains the unique undirected union of Euclidean 24-nearest-neighbor relations. For each edge $(i, j)$, the covariance-aware squared distance is

$$\begin{aligned} \delta_{ij}^2 &= (\mathbf{x}_i - \mathbf{x}_j)^\top (\mathbf{\Sigma}_i + \mathbf{\Sigma}_j + \eta_{ij}\mathbf{I})^{-1} (\mathbf{x}_i - \mathbf{x}_j), \\ \eta_{ij} &= 10^{-9} \operatorname{tr}(\mathbf{\Sigma}_i + \mathbf{\Sigma}_j) + \epsilon. \end{aligned} \tag{30}$$

Normals are obtained from stored values or, when unavailable, from the smallest-eigenvalue covariance eigenvectors. Before computing bilateral weights, we propagate their orientation along a maximum spanning forest of the retained neighborhood graph, using absolute normal agreement as the edge priority. Each child normal is flipped when necessary to agree with its oriented parent. The resulting unit normals $\mathbf{n}_i$ enter the signed weighting below:

$$\begin{aligned} a_{ij}^{\mathrm{SN}} &= \exp\left[-\frac{\delta_{ij}^2}{2\sigma_x^2 + \epsilon} - \frac{1 - \operatorname{clip}(\mathbf{n}_i^\top \mathbf{n}_j, -1, 1)}{2\sigma_n^2 + \epsilon}\right], \\ a_{ij}^{\mathrm{SNC}} &= a_{ij}^{\mathrm{SN}} \exp\left[-\frac{\|\boldsymbol{\ell}_i - \boldsymbol{\ell}_j\|_2^2}{2\sigma_c^2 + \epsilon}\right], \end{aligned} \tag{31}$$

where $\boldsymbol{\ell}_i = \mathrm{Lab}(\hat{\boldsymbol{s}}_i)$. Absolute normal agreement is used only to construct the orientation forest; the bilateral weights retain signed normal agreement. Orientation propagation ensures nonnegative agreement on forest edges, but does not guarantee it on every remaining graph edge. The automatic scales are

$$\sigma_x^2 = \operatorname{median}_{(i,j)\in\mathcal{E}} \delta_{ij}^2 + \epsilon, \qquad \sigma_c^2 = \operatorname{median}_{(i,j)\in\mathcal{E}} \|\boldsymbol{\ell}_i - \boldsymbol{\ell}_j\|_2^2 + \epsilon, \tag{32}$$

with $\sigma_n = 0.3$ and $\epsilon = 10^{-12}$ in the shared utilities. The superscripts SN and SNC indicate spatial–normal and spatial–normal–color weighting, respectively.

**Energy and automatic regularization.** For unary costs $U_i(k)$ and nonnegative symmetric edge weights $b_{ij}$, the common weighted-Potts objective is

$$E(\mathbf{z}) = \sum_i U_i(z_i) + \lambda \sum_{(i,j)\in\mathcal{E}} b_{ij} \mathbf{1}[z_i \neq z_j]. \tag{33}$$

The label-disagreement penalty is uniform across distinct palette entries; color and geometry influence the unary terms and edge weights. If $\lambda$ is not supplied, the code sets

$$\lambda = \beta \frac{\operatorname{median}_i \left(U_i^{(2)} - U_i^{(1)}\right)}{\max\left(\frac{2}{N}\sum_{(i,j)\in\mathcal{E}} b_{ij}, \epsilon\right)}, \tag{34}$$

where $U_i^{(1)}$ and $U_i^{(2)}$ are the two smallest unary costs and $\beta \in 1, 2, 4$ is the swept multiplier. This balances a typical unary margin against the average incident edge weight. The resulting $\lambda$ is method-dependent because the unary costs and edge weights can differ. Statistics are computed over supported Gaussians and their retained edges, with $N$ denoting the supported count here. For Option B, $U_i^{(2)} - U_i^{(1)} = C_i o_i \Delta_i$, $b_{ij} = \tilde{a}_{ij}$, and $\mathcal{E} = \mathcal{E}_+$. Thus, choosing $\beta = 2$ reduces Equation 34 exactly to the default $\lambda$ setting in Algorithm 2.

**Sequential label updates.** Methods A–D use sequential ICM, updating one supported Gaussian at a time using the latest neighbor labels:

$$z_i \leftarrow \arg\min_k \left[ U_i(k) + \lambda \sum_{j \in \mathcal{N}(i)} b_{ij} \mathbf{1}[k \neq z_j] \right]. \tag{35}$$

The update order is shuffled at each sweep, and labels are updated in place. The current label is retained when it minimizes the local cost, including ties. Confidence, unary costs, edge weights, and $\lambda$ remain fixed, so each update is non-increasing in energy. Optimization stops when no labels change or after 20 sweeps. This procedure matches Algorithm 2 and does not guarantee a global optimum.

### F.4 Custom Quantization Variants

The six variants adapt spatial labeling and component-processing ideas to the Gaussian semantic assignment problem. They vary the unary reliability, graph structure, edge weights, or optimization procedure.

**A: Weighted-Potts regularization.** Option A uses the original unary costs and spatial–normal weights:

$$U_i^A(k) = D_i(k), \qquad b_{ij}^A = a_{ij}^{\mathrm{SN}}. \tag{36}$$

It provides the reference spatial regularizer against which confidence gating is compared. Although some script comments describe it as spatial-only, the executable configuration includes normal agreement. Its purpose is to reduce fragmented assignments without explicit color gating on edges.

**B: Confidence-gated quantization.** Option B uses the combined confidence $C_i = (q_i C_i^{\mathrm{vote}} C_i^{\mathrm{support}})^{1/3}$ from Equation 7. Its palette-margin component is

$$\Delta_i = D_i^{(2)} - D_i^{(1)}, \qquad q_i = \frac{\Delta_i}{\Delta_i + \tau}, \qquad \tau = \mathrm{median}_{i \in \mathcal{I}_+} \Delta_i + \epsilon. \tag{37}$$

Vote confidence is evaluated against the initial nearest-palette label $z_i^{(0)}$. The resulting $C_i$ is held fixed throughout sequential ICM, with costs

$$U_i^B(k) = C_i o_i D_i(k), \qquad b_{ij}^B = a_{ij}^{\mathrm{SN}} \sqrt{C_i C_j}. \tag{38}$$

This is the spatial–normal configuration of Equation 5 and the preparation and optimization algorithms in Appendix E. Confidence and opacity scale unary costs, while the symmetric confidence gate scales spatial regularization. These terms do not guarantee preservation of confident labels or correction of uncertain ones. For a supported Gaussian with $C_i = 0$, both unary and incident gated costs vanish; tie preservation retains its initial label. Unsupported Gaussians remain unassigned.

**C: Scale-adaptive neighborhoods.** Option C adapts the neighborhood to local point density. Let

$$s_i = \mathrm{median}_{j \in \mathcal{N}_8(i)} \|\mathbf{x}_i - \mathbf{x}_j\|_2, \qquad \widetilde{d}_{ij} = \frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2}{\sqrt{s_i s_j} + \epsilon}. \tag{39}$$

Candidate edges are retained when $\widetilde{d}_{ij} < \tau_s$, with $\tau_s = 2$ in the sweep. Spatial distances and bilateral scales are recomputed on the retained graph $\mathcal{E}_C$. The energy uses

$$U_i^C(k) = D_i(k), \qquad b_{ij}^C = a_{ij,C}^{\mathrm{SN}} \sqrt{q_i q_j}, \qquad (i,j) \in \mathcal{E}_C. \tag{40}$$

Unlike Option B, this variant leaves unary costs unscaled and uses only the margin component $q_i$ in its edge gate. The standalone script defaults to 32 candidate neighbors; the shared comparison and sweep use 24.

**D: Bilateral CRF–Potts regularization.** Option D uses

$$U_i^D(k) = D_i(k), \qquad b_{ij}^D = a_{ij}^{\mathrm{SNC}}. \tag{41}$$

It adapts appearance-sensitive Gaussian edge weighting associated with bilateral CRF formulations, but operates on the sparse Gaussian neighborhood graph with the label-update procedures above. It is not a fully connected CRF with mean-field inference. The color term can weaken smoothing across true semantic boundaries, but can also preserve isolated color errors because those errors themselves create strong color differences.

**E: Alpha-expansion graph cuts.** Option E uses the same unscaled unary costs and spatial–normal weights as A, but replaces single-node updates with alpha-expansion moves. For a candidate label $\alpha$, define

$$z_i'(\mathbf{x}) = \begin{cases} \alpha, & x_i = 1, \\ z_i, & x_i = 0, \end{cases} \qquad \mathbf{x} \in \{0,1\}^N. \tag{42}$$

Each move minimizes $E(\mathbf{z}'(\mathbf{x}))$ over this binary choice using a graph cut, and is accepted only if it decreases the energy. Labels are visited over a bounded number of expansion sweeps.

Each binary expansion move can be solved exactly for the nonnegative weighted-Potts model, but the complete multi-label procedure is an approximate optimizer. The limited sweep budget does not establish convergence to an expansion-local optimum. When PyMaxflow is unavailable, the scripts use an explicitly named ICM fallback; such runs should not be interpreted as graph-cut results.

**F: Confidence-aware component cleanup.** Option F begins with nearest-palette labels and constructs connected components using graph edges whose endpoints share a label. A component $\mathcal{C}$ is considered for relabeling only if

$$|\mathcal{C}| < 20, \qquad \overline{q}_{\mathcal{C}} = \frac{1}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} q_i < 0.4. \tag{43}$$

Let $\partial\mathcal{C}$ contain edges leaving the component, and define the boundary support for label $k$ as

$$v_{\mathcal{C}}(k) = \sum_{\substack{i \in \mathcal{C},\, j \notin \mathcal{C} \\ (i,j) \in \mathcal{E}}} a_{ij}^{\mathrm{SN}} \mathbf{1}[z_j = k]. \tag{44}$$

Processing continues only when boundary neighbors exist and

$$\max_k v_{\mathcal{C}}(k) \geq \frac{1}{2} \sum_k v_{\mathcal{C}}(k). \tag{45}$$

Among the current and neighboring labels, the component receives the label minimizing

$$J_{\mathcal{C}}(k) = \sum_{i \in \mathcal{C}} D_i(k) + \lambda \sum_{\substack{i \in \mathcal{C},\, j \notin \mathcal{C} \\ (i,j) \in \mathcal{E}}} a_{ij}^{\mathrm{SN}} \mathbf{1}[k \neq z_j]. \tag{46}$$

Components are identified initially and processed in sequence. Small components with sufficiently high mean confidence remain unchanged. This is a selective cleanup heuristic, not a guarantee that every thin or small object is preserved.

### F.5 Neutral-Color Artifacts and the Chroma Guard

A motivating failure mode was the assignment of some chromatic Gaussian colors to neutral palette entries. Such assignments can originate in the nearest-color costs or spread through spatial regularization. Since the colors encode semantic identities, a neutral assignment can correspond to the wrong object even when it is close under a perceptual dissimilarity. The diagnostic script additionally checks whether quantization is reading the intended SCD color fields rather than unrelated appearance coefficients.

To study this behavior, we introduce a chroma-guard variant. For Lab coordinates $(L, a, b)$, define chroma as

$$\chi(L, a, b) = \sqrt{a^2 + b^2}. \tag{47}$$

A palette entry is treated as neutral if $\chi(\mathrm{Lab}(\mathbf{c}_k)) < \tau_{\mathrm{neutral}}$. For Gaussians whose recovered chroma is at least $\tau_{\mathrm{guard}}$, the guarded costs are

$$D_i^g(k) = \begin{cases} M, & \chi(\mathrm{Lab}(\hat{\mathbf{s}}_i)) \geq \tau_{\mathrm{guard}}, \\ & \chi(\mathrm{Lab}(\mathbf{c}_k)) < \tau_{\mathrm{neutral}}, \\ D_i(k), & \text{otherwise}, \end{cases} \tag{48}$$

with $M = 10^6$ and both thresholds set to 12 in the sweep. The guard is applied only when the palette contains both neutral and non-neutral entries. Otherwise, the original costs are retained.

Guard-enabled configurations initialize labels and compute $q_i$ from $D_i^g$, then substitute these costs into the selected custom method. Option B also evaluates vote confidence against these initial labels and combines it with $q_i$ and support confidence to obtain fixed $C_i$. Low-chroma Gaussians retain access to all palette entries. The implementation uses a finite penalty rather than a hard feasibility constraint; in Option B, this penalty is also multiplied by $C_i o_i$. The guard therefore discourages neutral assignments but does not guarantee the correct semantic label or preserve every legitimate neutral-colored label.

### F.6 Shared Rendered-Image Evaluation

We compare the rendered semantic images against their corresponding reference segmentations using exact RGB matching. References are resized to the target rendering dimensions, with nearest-neighbor interpolation as the default to preserve discrete colors. ASPT and non-ASPT evaluations require references expressed in the corresponding palette, or an equivalent mapping back to common semantic labels.

For frame $f$, let $R_f(p)$ and $I_f(p)$ denote reference and rendered RGB values. Define

$$V_f(p) = \mathbf{1}[I_f(p) \neq (255, 255, 255)], \qquad e_f(p) = V_f(p)\mathbf{1}[I_f(p) \neq R_f(p)]. \tag{49}$$

Here RGB inequality means that at least one channel differs. Let

$$n_f = |\mathcal{P}_f|, \qquad v_f = \sum_p V_f(p), \qquad m_f = \sum_p e_f(p). \tag{50}$$

The four reported percentage errors are

$$\begin{aligned} E_{\text{all}}^{\text{pool}} &= 100\frac{\sum_f m_f}{\sum_f n_f}, \quad E_{\text{valid}}^{\text{pool}} = 100\frac{\sum_f m_f}{\sum_f v_f}, \\ E_{\text{all}}^{\text{mean}} &= \frac{100}{F}\sum_f \frac{m_f}{n_f}, \quad E_{\text{valid}}^{\text{mean}} = \frac{100}{F}\sum_f \frac{m_f}{v_f}. \end{aligned} \tag{51}$$

The implementation assigns zero to a ratio with a zero denominator. Pooled errors weight frames by their pixel counts, whereas mean errors weight frames equally. For equal image dimensions, the pooled and mean all-pixel errors coincide, as they do in the supplied results.

White target pixels are excluded from mismatch counting in both versions; the all-pixel version nevertheless retains them in its denominator. Consequently, neither metric fully penalizes missing coverage, and a change in the white-pixel mask can affect comparisons. Exact RGB agreement also measures color fidelity rather than semantic correctness independently of rendering and palette encoding. These errors are therefore reported separately from mIoU and the graph diagnostics.

### F.7 Diagnostics of the Quantized Gaussian Map

The custom-method diagnostics measure how spatial processing changes color fidelity and label structure. They complement rendered-image errors but do not independently establish semantic correctness. For fair within-setting comparisons, all methods are evaluated using the same base graph and the same unmodified unary matrix for the selected distance metric.

**Unary fidelity and assignment changes.** For final labels $z_i$ and initial labels $z_i^{(0)}$,

$$\begin{aligned} \overline{U} &= \frac{1}{N}\sum_i D_i(z_i), \\ U_{\text{med}} &= \text{median}_i\, D_i(z_i), \\ \overline{U}_0 &= \frac{1}{N}\sum_i D_i(z_i^{(0)}), \\ A_{\text{change}} &= \frac{1}{N}\sum_i \mathbf{1}[z_i \neq z_i^{(0)}]. \end{aligned} \tag{52}$$

The first two describe assigned-color cost, while the last two provide a reference and measure departure from the initial assignment. With the chroma guard enabled, $z_i^{(0)}$ is the guarded argmin; the sweep still evaluates its cost using the unmodified $D_i$. Thus the field named `mean_unary_argmin` need not equal the unrestricted minimum average unary cost in guarded configurations.

**Fragmentation and neighborhood agreement.** Let $\mathfrak{C}(\mathbf{z})$ be the connected components of the common graph after retaining only same-label edges. We report

$$\begin{aligned} N_{\text{comp}} &= |\mathfrak{C}(\mathbf{z})|, \\ N_{\text{small}} &= \sum_{\mathcal{C}\in\mathfrak{C}(\mathbf{z})} \mathbf{1}[|\mathcal{C}| < 20], \\ A_{\text{neighbor}} &= \frac{1}{|\mathcal{E}|} \sum_{(i,j)\in\mathcal{E}} \mathbf{1}[z_i = z_j]. \end{aligned} \tag{53}$$

The small-component count corresponds to the code's `n_isolated` field; these components need not be singletons or geometrically isolated. Lower fragmentation and higher agreement indicate smoother labeling, but over-merging distinct objects can produce the same trends.

**Label usage and neutral assignments.** The number of used palette entries and the fraction of neutral assignments are

$$\begin{aligned} K_{\text{used}} &= |\{z_i : i = 1, \dots, N\}|, \\ F_{\text{neutral}} &= \frac{1}{N} \sum_i \mathbf{1}[\chi(\text{Lab}(\mathbf{c}_{z_i})) < \tau_{\text{neutral}}]\,. \end{aligned} \tag{54}$$

Neither quantity has a universally preferred direction. Unused labels can indicate missing objects, while neutral assignments may be correct when the palette intentionally contains neutral semantic colors.

**Composite diagnostic score.** The sweep constructs an equal-weight score from $\overline{U}$, $N_{\text{small}}$, $A_{\text{neighbor}}$, and $F_{\text{neutral}}$. For diagnostic $x$ over configurations in one setting, let

$$r_x = \begin{cases} x_{\max} - x_{\min}, & x_{\max} > x_{\min}, \\ 1, & \text{otherwise}, \end{cases} \qquad \widehat{x} = \frac{x - x_{\min}}{r_x}. \tag{55}$$

Then

$$S_{\text{diag}} = \frac{1}{4}\left[(1 - \widehat{\overline{U}}) + (1 - \widehat{N}_{\text{small}}) + \widehat{A}_{\text{neighbor}} + (1 - \widehat{F}_{\text{neutral}})\right]. \tag{56}$$

This score encodes a heuristic preference for fidelity, smoothness, fewer small components, and fewer neutral assignments. Its normalization depends on the configurations included in the sweep. Moreover, raw unary costs from RGB, Lab, CMC, and CIEDE2000 have different scales; pooled min–max normalization does not remove that distinction. We therefore use rendered-image error, rather than this composite score, to identify the best configurations reported below.

### F.8 Comparison with Independent Quantization

Confidence-Gated Quantization with CIEDE2000 achieves the lowest reported overall all-pixel error in all four settings. Its best errors are $5.498\%$ and $5.694\%$ for instance segmentation with and without ASPT, and $4.442\%$ and $4.507\%$ for semantic segmentation. Relative to the best historical nearest-color method in each setting, the absolute reductions are approximately $0.67$, $0.53$, $0.36$, and $0.42$ percentage points, respectively (Table 8).

The result supports selecting Option B with CIEDE2000 among the evaluated configurations. It does not imply that every B configuration outperforms every alternative: the full ranked plot shows substantial variation with metric, guard, and regularization settings. The historical and shared-code baselines also differ in some color-space implementations, so within-sweep comparisons provide the cleaner assessment of adding spatial and confidence terms to a common unary cost.

Table 8: Best custom and historical nearest-color results in each setting. Values are overall all-pixel error percentages; lower is better. Custom values follow the three-decimal summary, while historical values retain the precision of the result log.

| Setting | Best custom | Error | Best historical | Error |
|---|---|---|---|---|
| Instance + ASPT | B / CIEDE2000 | **5.498** | CIEDE2000 | 6.1715 |
| Instance, no ASPT | B / CIEDE2000 | **5.694** | RGB | 6.2282 |
| Semantic + ASPT | B / CIEDE2000 | **4.442** | CMC | 4.8028 |
| Semantic, no ASPT | B / CIEDE2000 | **4.507** | CMC | 4.9300 |

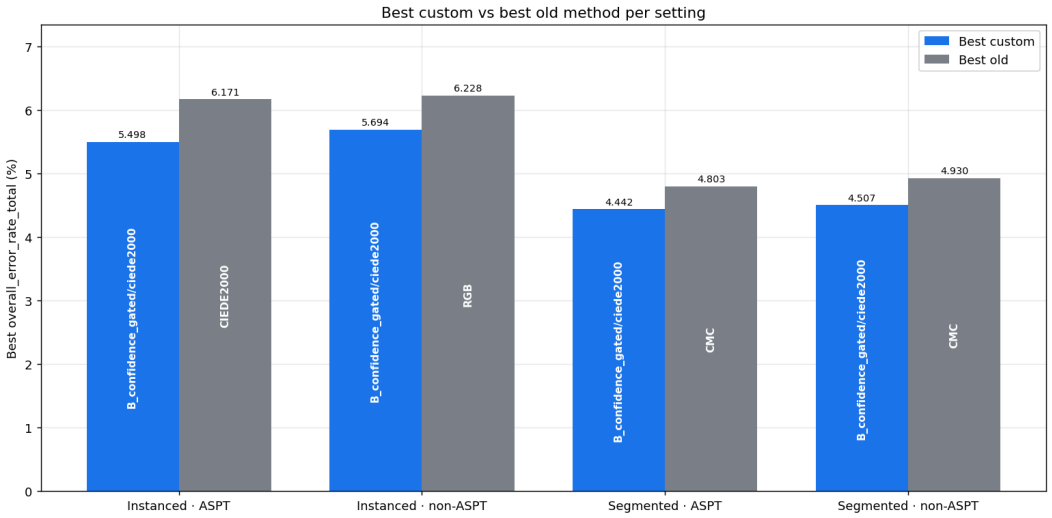



Figure 13: Best custom quantizer versus the best historical nearest-color method in each setting. Confidence B-Gated with CIEDE2000 has the lowest reported all-pixel error in all four comparisons.

**Historical results.** Tables 9, 10, and 11 report all 15 historical variants across the four settings. Their pooled and mean all-pixel errors are identical in the supplied logs, so Table 9 represents both quantities (sweep run on one scene of the REPLICA dataset).

### F.9 Effect of ASPT

ASPT improves the best observed result in both segmentation families. The minimum instance-segmentation error decreases from $5.694\%$ to $5.498\%$, an absolute reduction of $0.196$ percentage points. The minimum semantic-segmentation error decreases from $4.507\%$ to $4.442\%$, a reduction of $0.065$ percentage points. Both comparisons select Option B with CIEDE2000.

Figures 15 and 16 compare the best configuration within each method separately for ASPT and non-ASPT. These comparisons characterize the performance attainable under each setting, rather than isolating ASPT with every remaining hyperparameter fixed. Improvements are clearer for instance segmentation; several semantic-method differences are small at the displayed precision.

The historical results also show that palette transformation does not improve every fixed distance variant. For example, RGB and Lab all-pixel errors increase with ASPT in both segmentation families. Thus ASPT should be assessed jointly with the quantizer: its benefit in the selected MonoVoc configuration does not imply a universal improvement for every color-space assignment rule.

### F.10 Relationship Between Structural Diagnostics and Image Error

To examine whether the internal diagnostics track the shared image errors, we calculate Pearson correlations across the $608$ configurations. For configuration-level measurements $x_t$ and $y_t$, Pearson

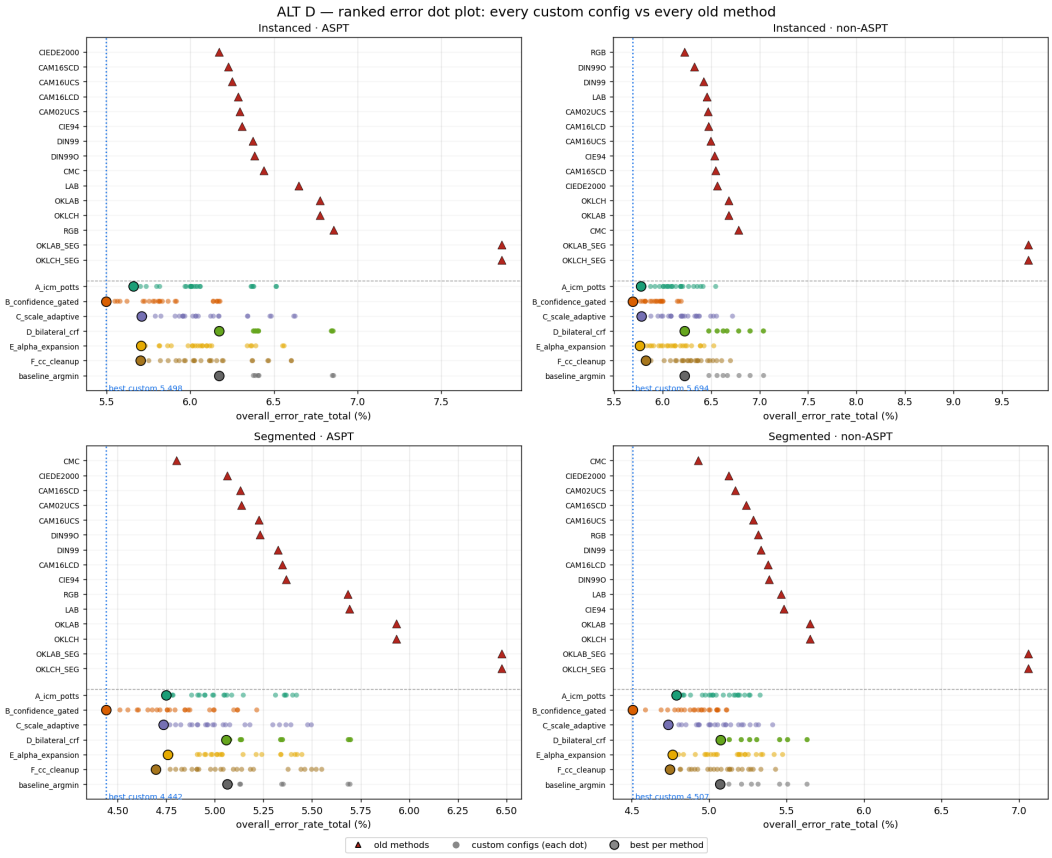



Figure 14: Configuration-level error comparison across the four settings. Triangles denote historical nearest-color methods; dots denote configurations from the custom sweep. Enlarged circles mark each method's minimum, and the vertical dotted line marks the best custom result.

correlation is

$$r(x, y) = \frac{\sum_t (x_t - \overline{x})(y_t - \overline{y})}{\sqrt{\sum_t (x_t - \overline{x})^2} \sqrt{\sum_t (y_t - \overline{y})^2}}. \tag{57}$$

We additionally calculate Spearman correlation with overall all-pixel error:

$$\rho_s(x, y) = r(\mathrm{rank}(x), \mathrm{rank}(y)), \tag{58}$$

using average ranks for ties. Spearman correlation measures monotonic association, including associations that are not linear; it does not detect every form of nonlinear dependence.

**Spatial agreement and fragmentation.** Higher neighbor agreement is associated with lower all-pixel error ($r = -0.757$, $\rho_s = -0.742$), while more components and small components are associated with higher error. The per-setting plots provide complementary views of these relationships (Figures 17 and 18). The best configurations do not simply maximize smoothness or minimize component count, supporting the use of a balance between color evidence and spatial regularization.

**Unary costs and the composite score.** Final mean unary cost has little pooled linear association with all-pixel error ($r = 0.058$), and its Spearman correlation is also small ($\rho_s = -0.050$). Initial unary cost and the fraction of changed labels similarly show weak associations. The composite diagnostic score is more aligned with lower error, but only moderately so ($r = -0.320$, $\rho_s = -0.329$). These observations reinforce the distinction between minimizing a color-space cost and obtaining an accurate semantic rendering. Cross-metric unary scales also limit interpretation of their pooled correlations.

Table 9: Historical nearest-color quantizers: pooled and mean all-pixel error (%). These two measures coincide in the supplied results. I and S denote instance and semantic segmentation; best entries per column are bold.

| Variant | I + ASPT | I, no ASPT | S + ASPT | S, no ASPT |
|---|---|---|---|---|
| CAM02UCS | 6.2970 | 6.4682 | 5.1380 | 5.1672 |
| CAM16LCD | 6.2867 | 6.4751 | 5.3474 | 5.3809 |
| CAM16SCD | 6.2260 | 6.5462 | 5.1317 | 5.2391 |
| CAM16UCS | 6.2502 | 6.4944 | 5.2272 | 5.2832 |
| CIE94 | 6.3077 | 6.5349 | 5.3674 | 5.4820 |
| CIEDE2000 | **6.1715** | 6.5621 | 5.0637 | 5.1273 |
| CMC | 6.4386 | 6.7808 | **4.8028** | **4.9300** |
| DIN99 | 6.3737 | 6.4237 | 5.3251 | 5.3319 |
| DIN99O | 6.3831 | 6.3260 | 5.2315 | 5.3855 |
| LAB | 6.6489 | 6.4597 | 5.6933 | 5.4637 |
| OKLAB | 6.7746 | 6.6805 | 5.9321 | 5.6508 |
| OKLAB_SEG | 7.8596 | 9.7680 | 6.4734 | 7.0582 |
| OKLCH | 6.7746 | 6.6805 | 5.9321 | 5.6508 |
| OKLCH_SEG | 7.8596 | 9.7680 | 6.4734 | 7.0582 |
| RGB | 6.8568 | **6.2282** | 5.6851 | 5.3157 |

Table 10: Historical nearest-color quantizers: pooled non-white-pixel error (%). Lower is better.

| Variant | I + ASPT | I, no ASPT | S + ASPT | S, no ASPT |
|---|---|---|---|---|
| CAM02UCS | 7.8106 | 8.2061 | 6.3780 | 6.4035 |
| CAM16LCD | 7.7978 | 8.2149 | 6.6380 | 6.6683 |
| CAM16SCD | 7.7226 | 8.3050 | 6.3702 | 6.4927 |
| CAM16UCS | 7.7525 | 8.2394 | 6.4887 | 6.5473 |
| CIE94 | 7.8238 | 8.2907 | 6.6628 | 6.7936 |
| CIEDE2000 | **7.6550** | 8.3252 | 6.2858 | 6.3541 |
| CMC | 7.9863 | 8.6026 | **5.9620** | **6.1096** |
| DIN99 | 7.9058 | 8.1496 | 6.6103 | 6.6076 |
| DIN99O | 7.9174 | 8.0257 | 6.4941 | 6.6741 |
| LAB | 8.2471 | 8.1952 | 7.0673 | 6.7710 |
| OKLAB | 8.4030 | 8.4754 | 7.3637 | 7.0028 |
| OKLAB_SEG | 9.7489 | 12.3924 | 8.0357 | 8.7470 |
| OKLCH | 8.4030 | 8.4754 | 7.3637 | 7.0028 |
| OKLCH_SEG | 9.7489 | 12.3924 | 8.0357 | 8.7470 |
| RGB | 8.5049 | **7.9015** | 7.0572 | 6.5875 |

**Label count and neutral assignments.** The strongest pooled associations involve label count ($r = 0.889$) and neutral-assignment fraction ($r = -0.877$). However, the corresponding plots contain distinct groups with different label inventories and neutral fractions. Pooling semantic and instance settings can therefore produce strong associations partly through differences between the settings themselves. These results do not establish that discarding labels or increasing neutral assignments improves semantic correctness, nor do they establish that the chroma guard should always be enabled.

All correlations are descriptive summaries of a parameter sweep, not independent scene-level trials or causal effects. The four image-error columns also share mismatch counts, and two coincide numerically here. Similar correlations across those columns consequently do not constitute four independent confirmations. Also correlation sign in near zero values are not stable and both are consistent with no association.

### F.11 Complete Diagnostic Plots

The following plots retain all nine diagnostic comparisons. Each contains four panels corresponding to the image-error definitions in Equation 51. Gray points represent configurations pooled across the

Table 11: Historical nearest-color quantizers: mean per-frame non-white-pixel error (%). Lower is better.

| Variant | I + ASPT | I, no ASPT | S + ASPT | S, no ASPT |
|---|---|---|---|---|
| CAM02UCS | 7.6050 | 7.9362 | 6.2177 | 6.2660 |
| CAM16LCD | 7.5941 | 7.9422 | 6.4744 | 6.5216 |
| CAM16SCD | 7.5207 | 8.0343 | 6.2131 | 6.3535 |
| CAM16UCS | 7.5493 | 7.9679 | 6.3293 | 6.4053 |
| CIE94 | 7.6161 | 8.0042 | 6.5023 | 6.6457 |
| CIEDE2000 | **7.4530** | 8.0482 | 6.1363 | 6.2147 |
| CMC | 7.7756 | 8.3334 | **5.8255** | **5.9652** |
| DIN99 | 7.6942 | 7.8806 | 6.4485 | 6.4607 |
| DIN99O | 7.7043 | 7.7610 | 6.3359 | 6.5275 |
| LAB | 8.0301 | 7.9226 | 6.8921 | 6.6161 |
| OKLAB | 8.1869 | 8.1939 | 7.1640 | 6.8356 |
| OKLAB_SEG | 9.4661 | 11.9566 | 7.8570 | 8.5602 |
| OKLCH | 8.1869 | 8.1939 | 7.1640 | 6.8356 |
| OKLCH_SEG | 9.4661 | 11.9566 | 7.8570 | 8.5602 |
| RGB | 8.2769 | **7.6214** | 6.8720 | 6.4341 |

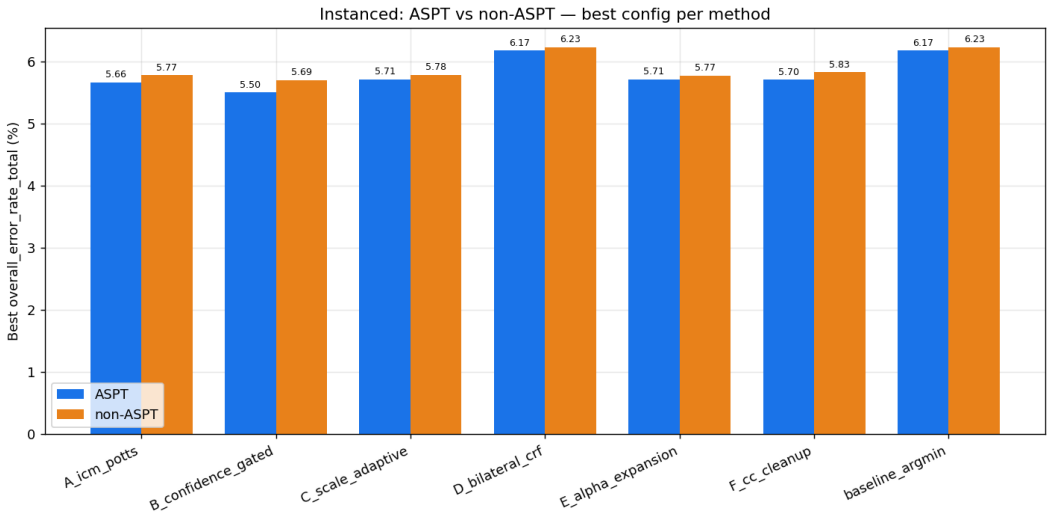



Figure 15: Instance-segmentation comparison with and without ASPT. Each bar is the minimum all-pixel error over that method's evaluated configurations, with configurations selected independently in the two settings.

four settings, and the blue point identifies the minimum for the panel's error measure. In these pooled plots, the lowest errors occur in the semantic-segmentation ASPT setting.

**Summary.** The evaluated configurations favor Confidence B-Gated Quantization with CIEDE2000 over independent nearest-color assignment, and ASPT further improves its best reported result in both segmentation settings. Spatial diagnostics help explain differences between configurations, but neither unary fidelity, smoothness, neutral fraction, nor the composite score replaces evaluation against semantic observations. The chroma guard remains an experimental variant whose usefulness must be judged from the resulting assignments and image errors, rather than from gray suppression alone.

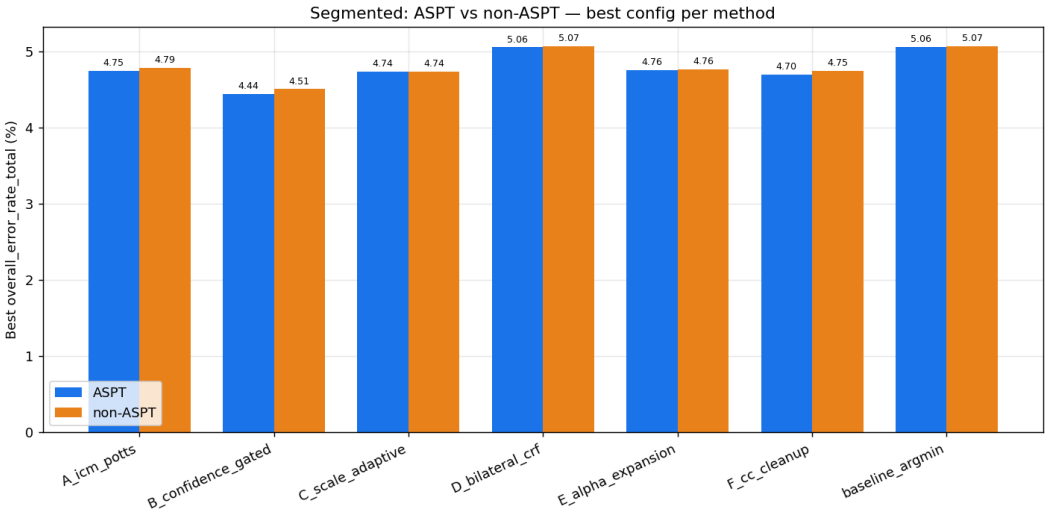



Figure 16: Semantic-segmentation comparison with and without ASPT. Confidence B-Gated obtains the lowest error in both settings; some other differences are small at the displayed precision.

Table 12: Pearson correlations between internal diagnostics and rendered-image errors across all 608 configurations. Pool and Mean denote pooled and per-frame averaging; All and NW denote all-pixel and non-white denominators.

| **Diagnostic** | **Pool–All** | **Pool–NW** | **Mean–All** | **Mean–NW** |
|---|---|---|---|---|
| Labels used | $+0.889$ | $+0.892$ | $+0.889$ | $+0.889$ |
| Neutral-assignment fraction | $-0.877$ | $-0.882$ | $-0.877$ | $-0.878$ |
| Neighbor agreement | $-0.757$ | $-0.746$ | $-0.757$ | $-0.748$ |
| Component count | $+0.692$ | $+0.677$ | $+0.692$ | $+0.681$ |
| Small-component count | $+0.670$ | $+0.656$ | $+0.670$ | $+0.660$ |
| Composite score | $-0.320$ | $-0.305$ | $-0.320$ | $-0.308$ |
| Changed-assignment fraction | $-0.131$ | $-0.121$ | $-0.131$ | $-0.122$ |
| Initial mean unary cost | $+0.089$ | $+0.073$ | $+0.089$ | $+0.076$ |
| Final mean unary cost | $+0.058$ | $+0.044$ | $+0.058$ | $+0.047$ |

## G Qualitative Reconstruction and Segmentation Examples

Figure 28 presents an example semantic Gaussian map, while Figure 29 shows four ScanNet reconstructions. Colors visualize semantic assignments rather than scene appearance.

## H Object Retrieval Evaluation

**Query construction.** We select 42 objects and construct 15 short retrieval queries per object, spanning lexical, categorical, functional, visual, contextual, and contrastive descriptions. The reported evaluation contains 630 queries. Short queries emphasize object ranking rather than free-form answer generation. Table 14 illustrates the query types for chairs; reference targets are scene-specific.

**Query matching.** Each query is encoded in the same embedding space as the candidate object representations. For cosine-based matching, query embedding $\mathbf{e}_q$ and object embedding $\mathbf{e}_k$ receive score

$$s(q,k) = \frac{\mathbf{e}_q^\top \mathbf{e}_k}{\|\mathbf{e}_q\|_2 \|\mathbf{e}_k\|_2}. \tag{59}$$

Objects are ranked by decreasing similarity, and the first one or three candidates define the Top-1 and Top-3 results. MonoVoc uses LLaMA-derived text representations; CLIP and Perception Encoder use their corresponding text and visual embedding spaces.

Table 13: Spearman correlations with pooled all-pixel error across all 608 configurations.

| **Diagnostic** | **Spearman $\rho_s$** |
|---|---|
| Labels used | $+0.859$ |
| Neutral-assignment fraction | $-0.753$ |
| Neighbor agreement | $-0.742$ |
| Component count | $+0.663$ |
| Small-component count | $+0.637$ |
| Composite score | $-0.329$ |
| Changed-assignment fraction | $-0.139$ |
| Final mean unary cost | $-0.050$ |
| Initial mean unary cost | $-0.048$ |

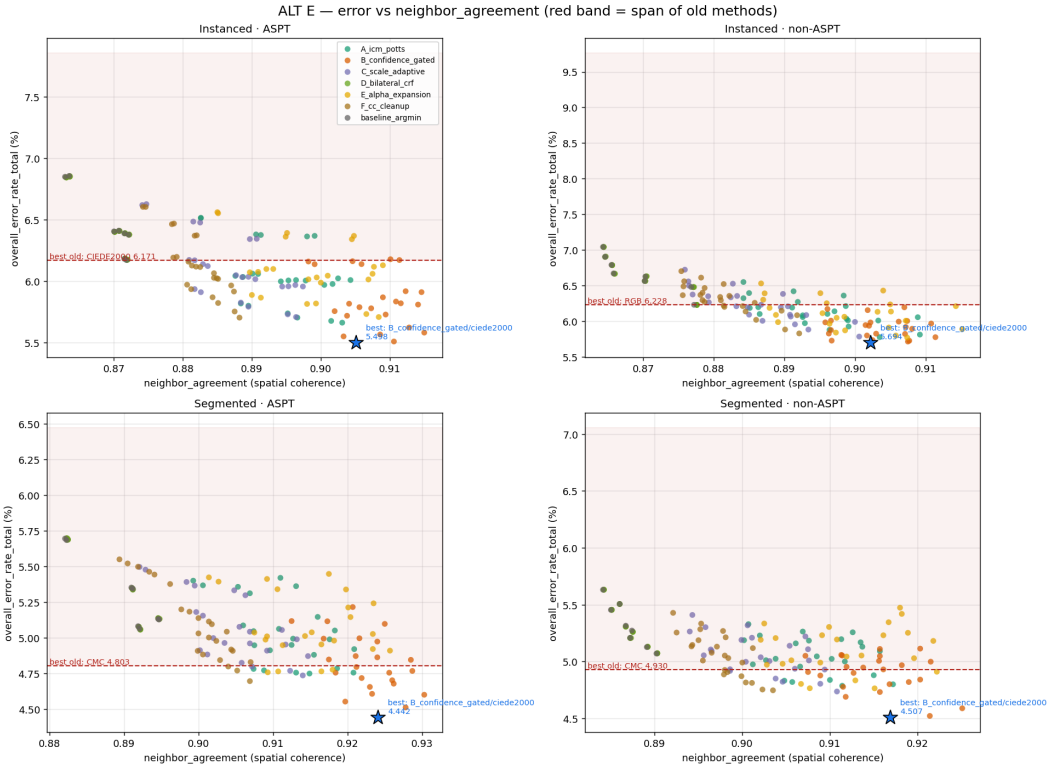



Figure 17: All-pixel error versus neighbor-label agreement within each setting. Stars mark the minimum-error custom configuration; dashed lines and shaded bands indicate the best historical error and historical error range.

**Retrieval output.** After ranking object embeddings by query similarity, we resolve each selected database entry to its semantic palette color. This identifier selects the corresponding Gaussians in the map, allowing the retrieved object to be isolated and rendered. Retrieval therefore returns an object assignment and its visual output, rather than a generated text answer.

**Hard and soft masking.** The CLIP and Perception Encoder variants differ in the visual context retained when extracting object embeddings. For RGB image $I$ and binary target mask $M$, hard masking uses $I_{\text{hard}} = M \odot I$, preserving the target's appearance and setting the background to black. Soft masking uses

$$I_{\text{soft}} = \big[M + \rho(1 - M)\big] \odot I, \qquad 0 < \rho < 1, \tag{60}$$

preserving the target while attenuating surrounding content. Thus soft masking retains contextual evidence, whereas hard masking isolates the target. These operations modify visual inputs, not query text.

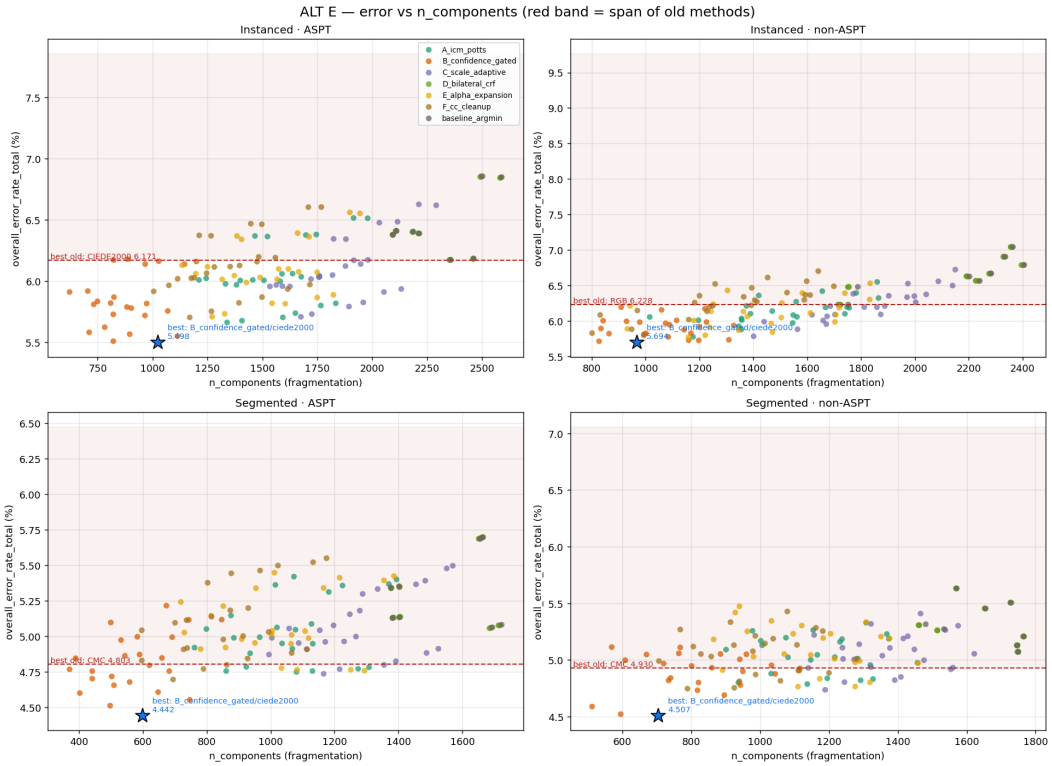



Figure 18: All-pixel error versus same-label component count within each setting. The plots relate fragmentation to image error while retaining the distinction between configurations and method families.

**Retrieval metrics.** Let $g_n$ be the single annotated target for query $n$, and $\mathcal{R}_n^{(k)}$ its top-$k$ retrieved candidates. Following rank-based cross-modal retrieval evaluation (Song & Soleymani, 2019), we report

$$\mathrm{Acc@}\,k = \frac{1}{N_\mathrm{r}} \sum_{n=1}^{N_\mathrm{r}} \mathbf{1}\left[g_n \in \mathcal{R}_n^{(k)}\right], \qquad k \in \{1, 3\}. \tag{61}$$

With one ground-truth target per query, this criterion is equivalent to Recall@$k$. Percentages are obtained by multiplying by 100. The candidate pool and reference targets must be identical across the compared configurations.

**Results by query category.** Tables 15 and 16 report category-level accuracy. MonoVoc achieves the strictly highest Top-1 accuracy in 14 of the 15 categories and ties SceneSplat on the remaining contrastive-phrase category. Its largest Top-1 gains over SceneSplat are 12.0 percentage points on both affordance and semantic-paraphrase queries (81.0% versus 69.0%), with similarly strong gains on functional and spatial/contextual descriptions. For Top-3 retrieval, MonoVoc attains the highest accuracy, including ties, in all 15 categories.

# I Controlled Scene-Grounded Question Answering

**Benchmark construction.** We construct a controlled QA benchmark for selected Replica and ScanNet scenes by manually inspecting the input scene data and writing scene-grounded questions and reference answers. Neither question construction nor reference-answer creation uses MonoVoc's outputs. The benchmark covers 42 objects with 15 questions per object, yielding 630 questions across five families: direct grounding, affordance and function, scenario reasoning, spatial relations, and counting or constraint-based composition. It evaluates whether the methods' representations

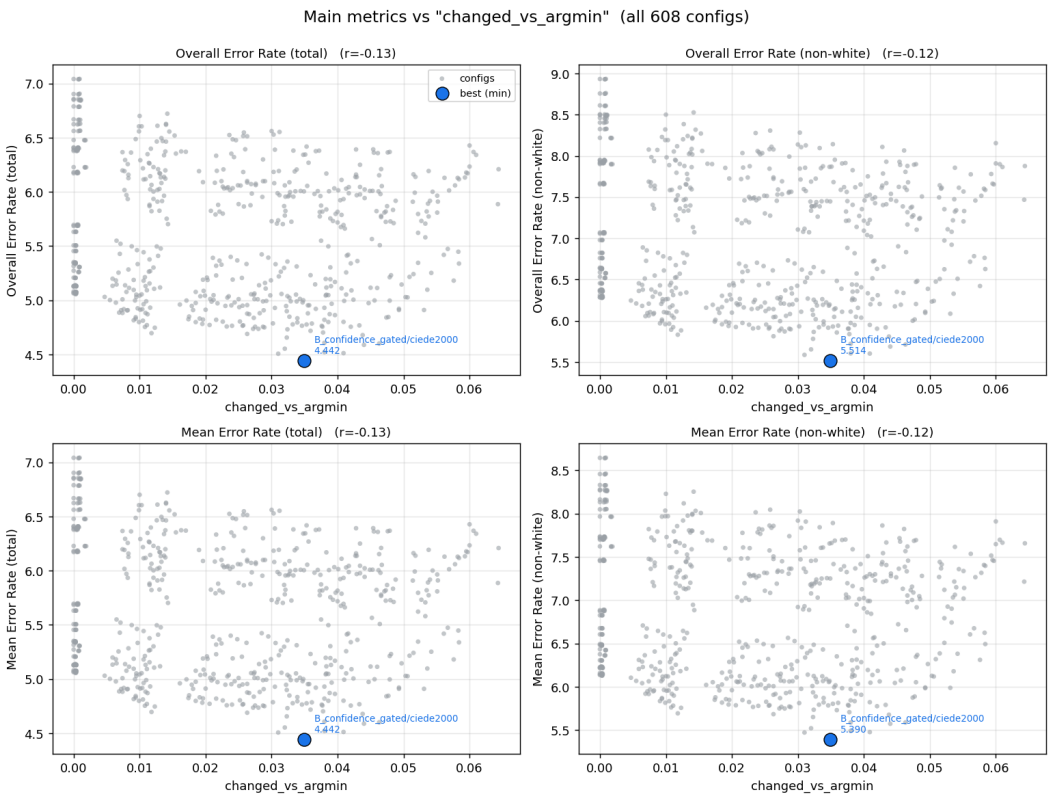



Figure 19: Image errors versus the fraction of assignments changed from the initial argmin. The number of changed labels alone has a weak association with final error.

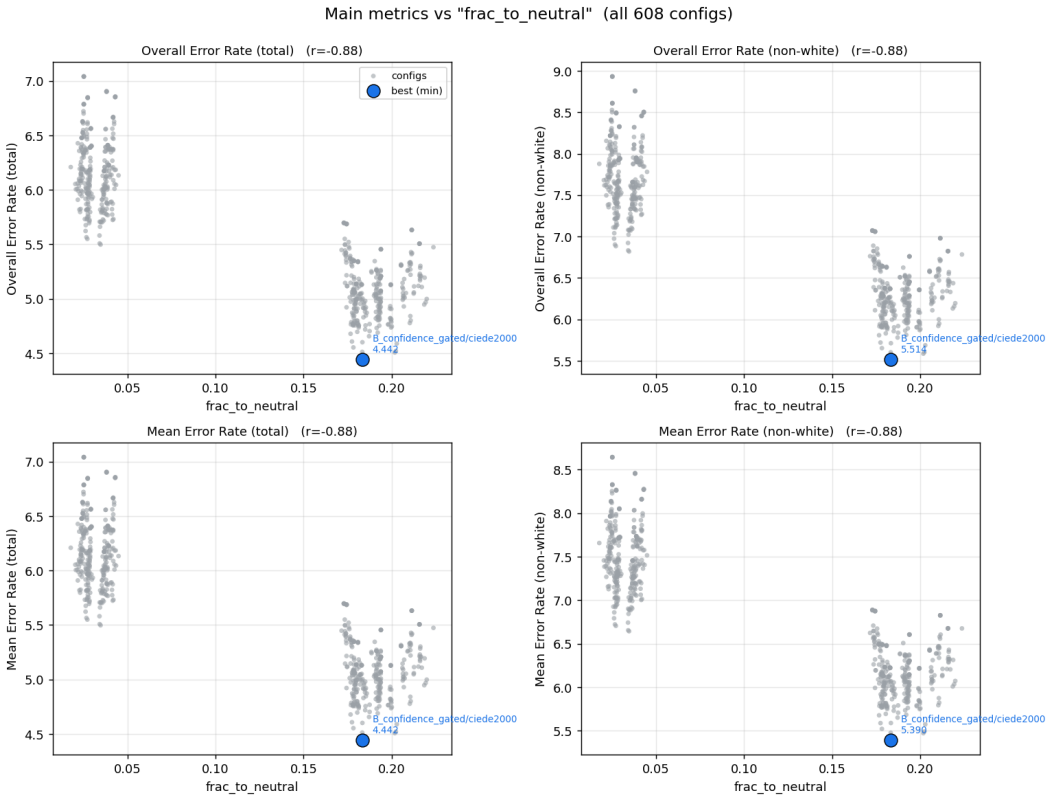



Figure 20: Image errors versus neutral-assignment fraction. The strong pooled negative correlation spans distinct setting-dependent groups and should not be interpreted as evidence that neutral assignments are universally beneficial.

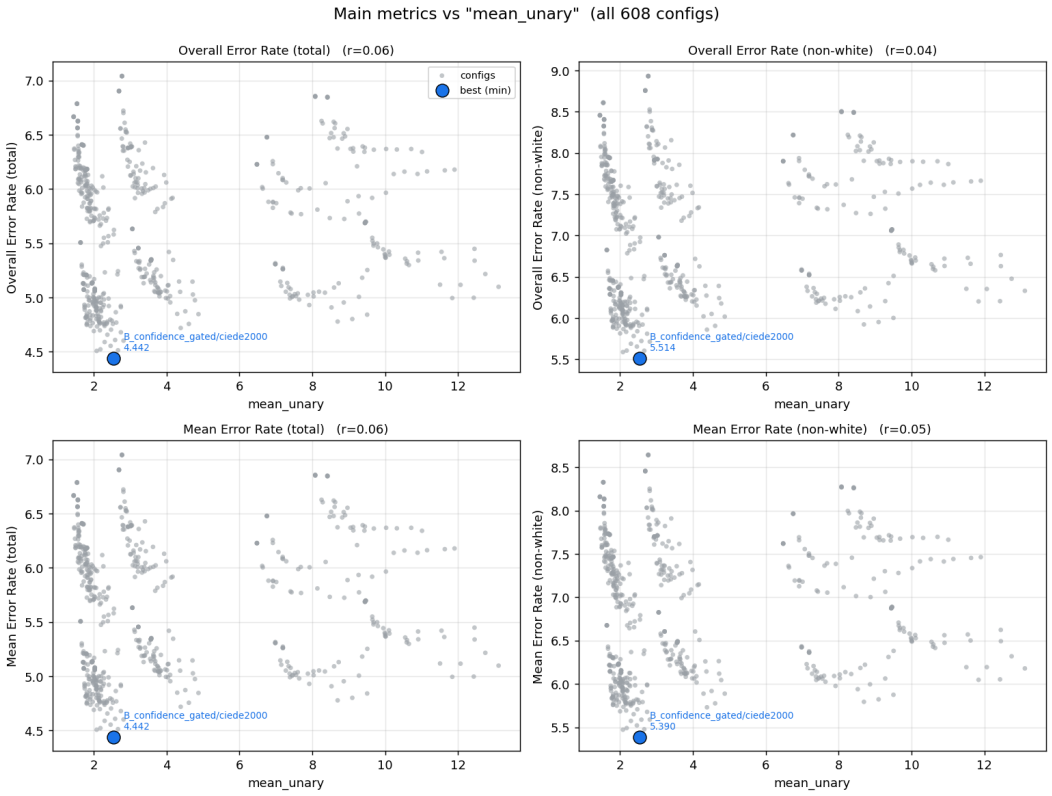



Figure 21: Image errors versus final mean unary cost. Costs are measured in the selected color dissimilarity, so their numerical scales differ across metric families.

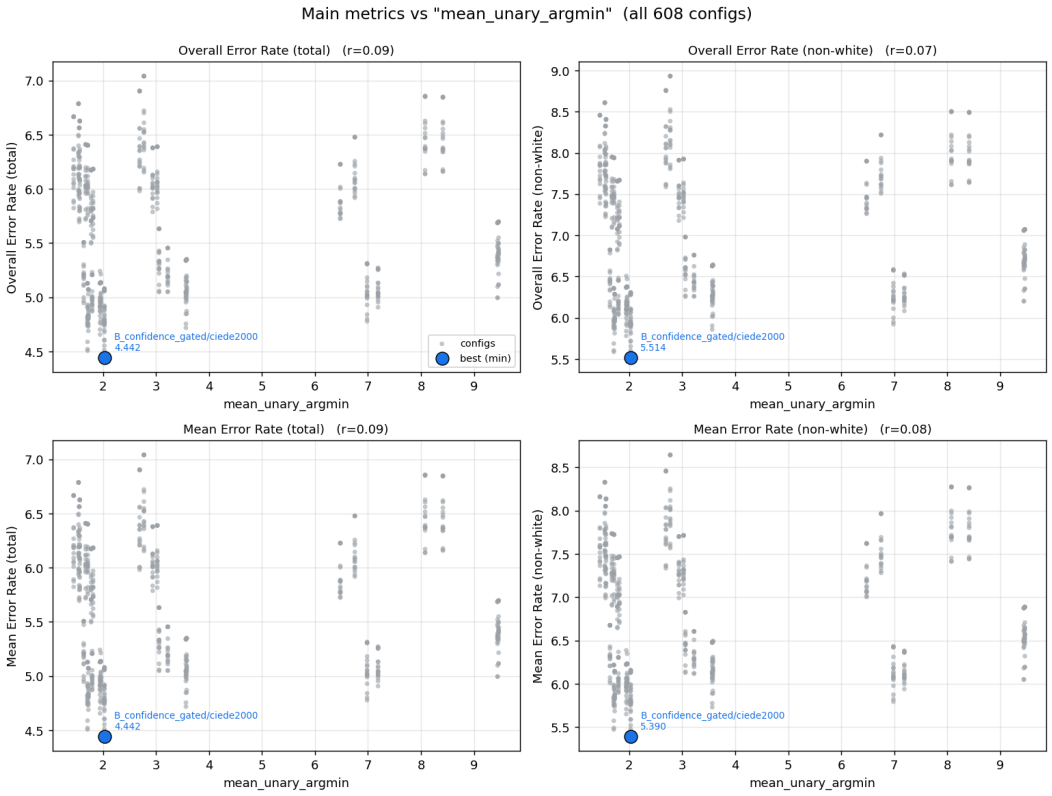



Figure 22: Image errors versus initial-assignment mean unary cost. Configurations sharing an initialization can have different final errors after spatial processing.

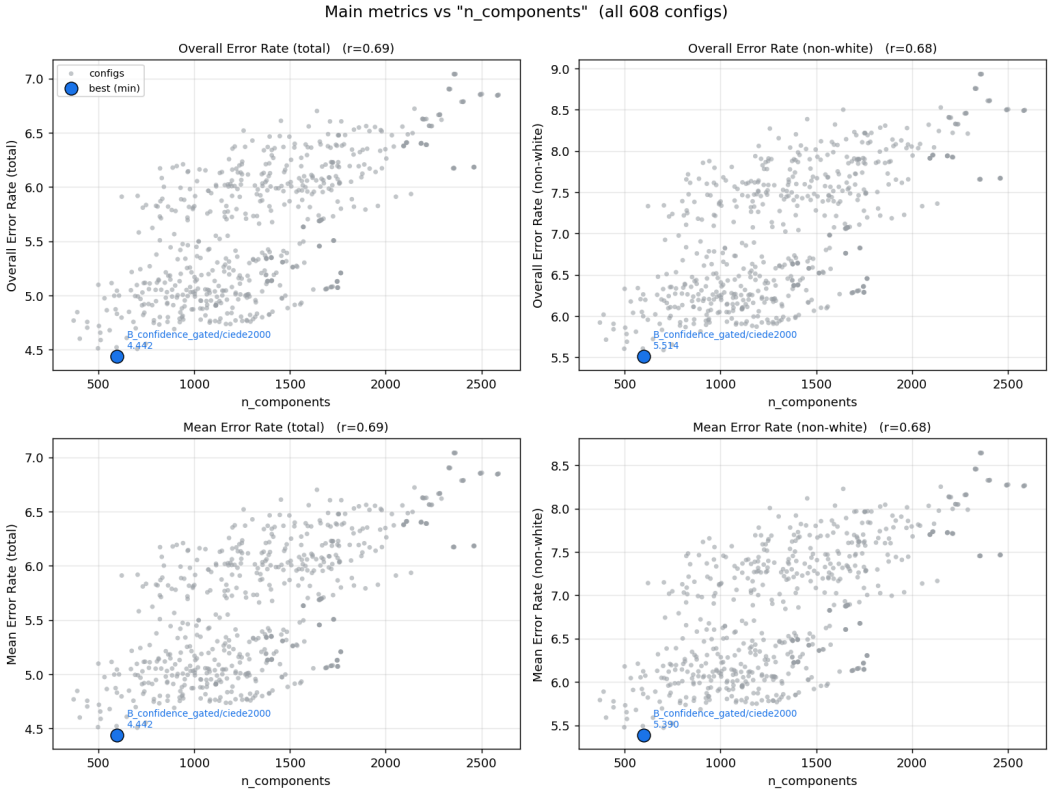



Figure 23: Image errors versus connected-component count. More fragmented labelings are associated with higher pooled error, although component count also depends on the segmentation setting and label inventory.

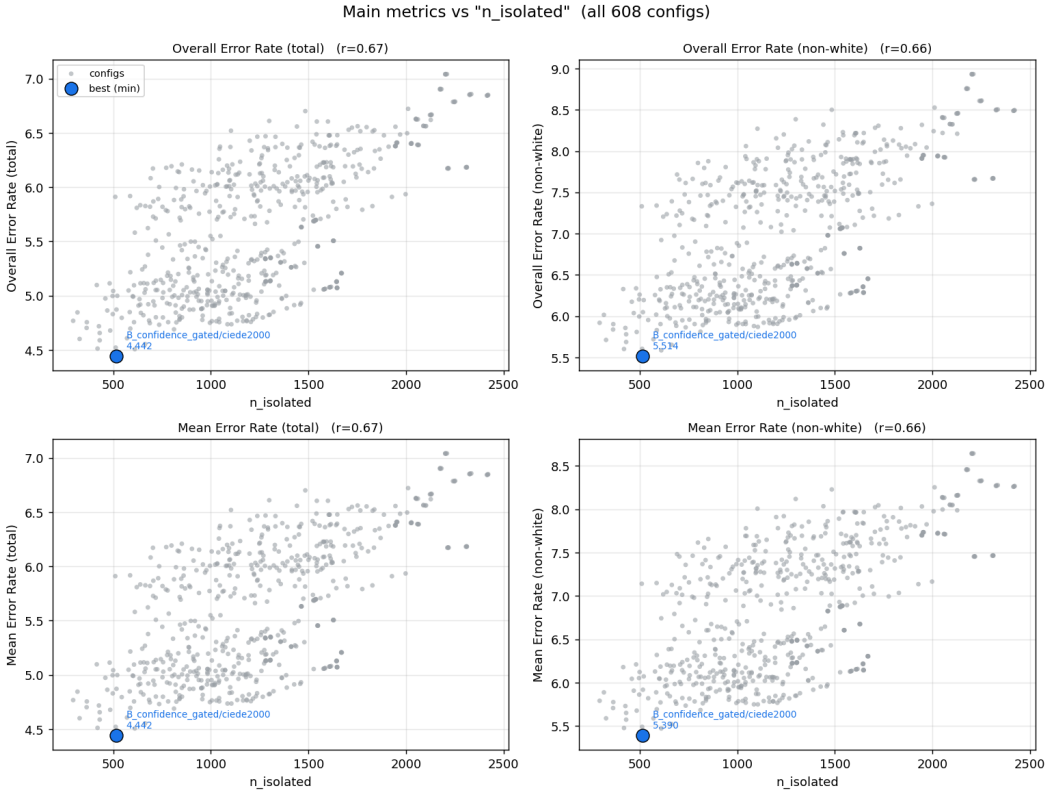



Figure 24: Image errors versus the number of same-label components containing fewer than 20 Gaussians. This diagnostic measures small fragments without establishing whether each fragment is erroneous.

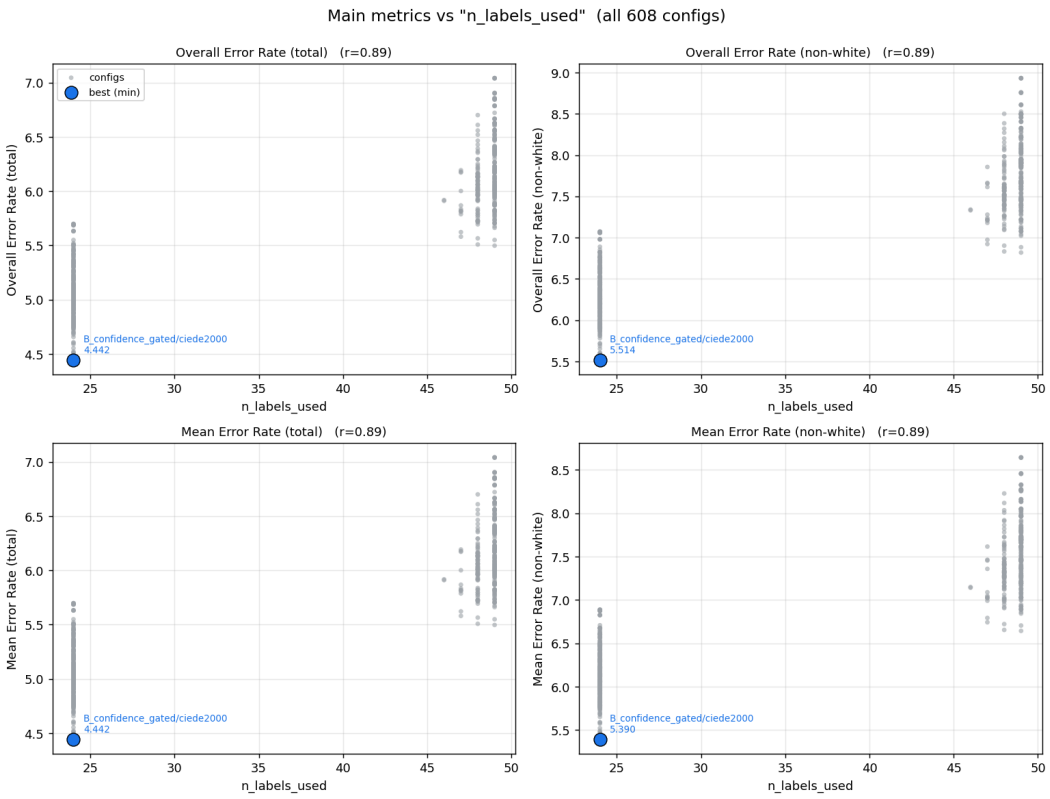



Figure 25: Image errors versus the number of palette labels used. The separated label-count groups highlight the influence of pooling semantic and instance settings.

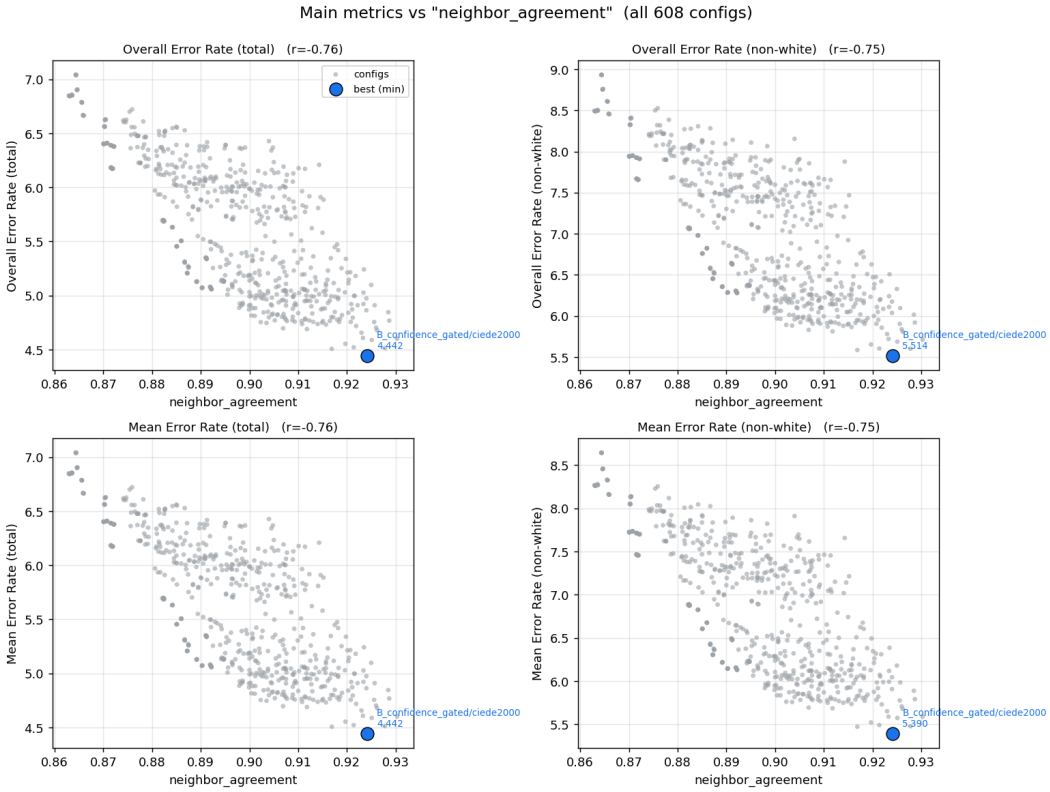



Figure 26: Image errors versus neighbor-label agreement. Higher agreement is associated with lower pooled error, but agreement alone cannot distinguish coherent objects from excessive label merging.

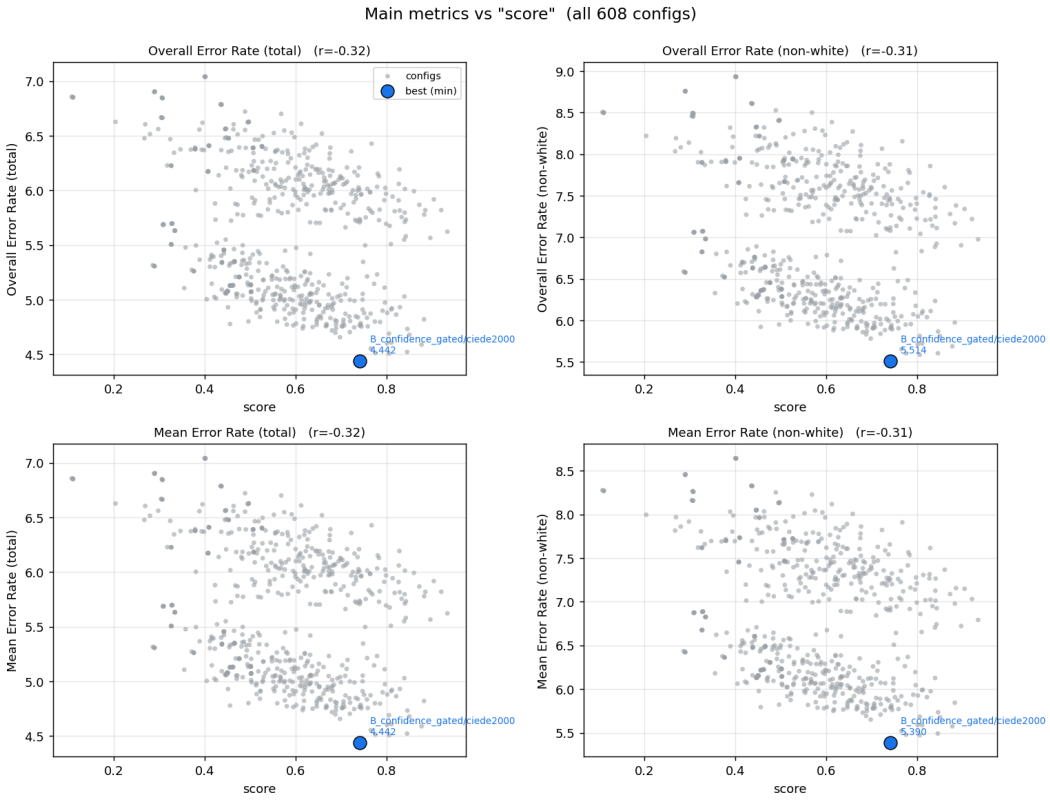



Figure 27: Image errors versus the composite diagnostic score. The score has a moderate negative association with error, and its maximum does not coincide with the minimum-error configuration.

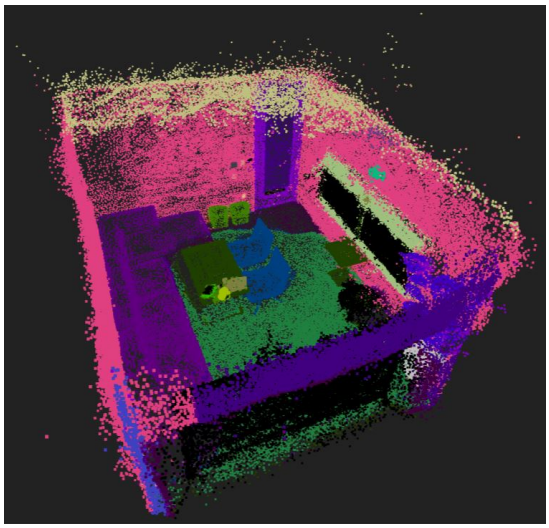
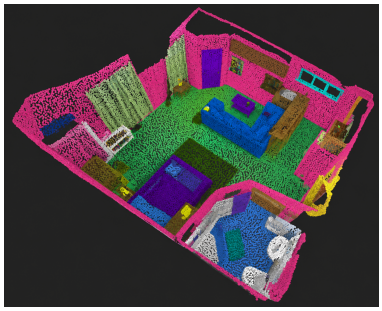
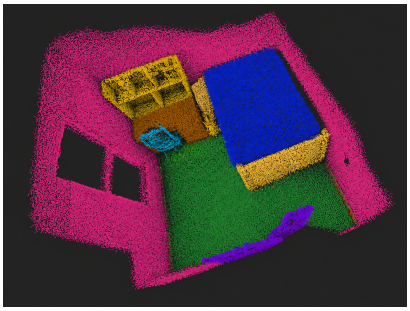


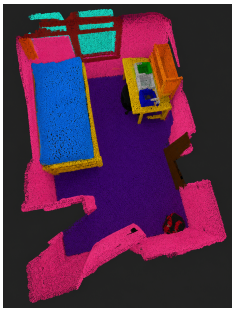
Figure 28: Example semantic Gaussian map with color-coded assignments.



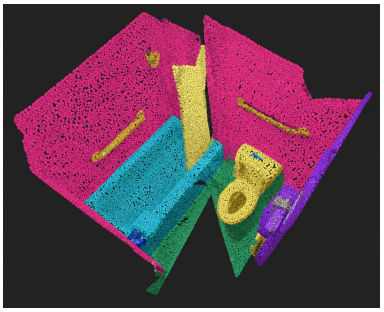
(a) ScanNet example 1.



(b) ScanNet example 2.



(c) ScanNet example 3.



(d) ScanNet example 4.

Figure 29: Qualitative semantic reconstructions on ScanNet, showing color-coded regions across different indoor scenes.

support downstream reasoning against independently authored references; it is not an official QA benchmark supplied with either dataset.

**Embedding and masking configurations.** Object matching follows the embedding-similarity procedure in Appendix H. CLIP and Perception Encoder use the same hard- and soft-masked visual inputs defined there. These representations provide object evidence for QA; the answer-generation procedure is distinct from embedding retrieval.

Table 14: Example retrieval queries for chairs. Each row represents one query category used in constructing the evaluation set.

| # | Query | Category |
|---|---|---|
| 1 | chair | Exact class |
| 2 | chairs | Plural |
| 3 | seat | Synonym |
| 4 | seating furniture | Category |
| 5 | desk chair | Contextual category |
| 6 | single-person seat | Functional description |
| 7 | place to sit | Affordance |
| 8 | sitting furniture | Functional phrase |
| 9 | seat with backrest | Visual / structural attribute |
| 10 | furniture with a seat | Attribute + category |
| 11 | work chair | Context / function |
| 12 | personal seat | Semantic paraphrase |
| 13 | something to sit on | Natural-language affordance |
| 14 | chair near desk | Spatial / contextual phrase |
| 15 | non-sofa seating | Contrastive phrase |

Table 15: Illustrative Top-1 retrieval accuracy (%) by query category. Each category contains exactly 42 queries, one per object; therefore every entry is computed as $100n/42$ for an integer $n$. PE denotes Perception Encoder.

| Query category | CLIP Soft | CLIP Hard | PE Soft | PE Hard | ObjectGS | SceneSplat | MonoVoc |
|---|---|---|---|---|---|---|---|
| Exact class | 90.5 | 83.3 | 97.6 | 92.9 | 95.2 | 97.6 | **100.0** |
| Plural | 88.1 | 81.0 | 95.2 | 88.1 | 92.9 | 95.2 | **97.6** |
| Synonym | 78.6 | 71.4 | 85.7 | 78.6 | 76.2 | 85.7 | **90.5** |
| Category | 71.4 | 64.3 | 78.6 | 71.4 | 66.7 | 81.0 | **88.1** |
| Contextual category | 69.0 | 61.9 | 76.2 | 69.0 | 66.7 | 78.6 | **85.7** |
| Functional description | 61.9 | 54.8 | 69.0 | 61.9 | 54.8 | 71.4 | **83.3** |
| Affordance | 57.1 | 50.0 | 64.3 | 57.1 | 50.0 | 69.0 | **81.0** |
| Functional phrase | 59.5 | 52.4 | 69.0 | 61.9 | 54.8 | 71.4 | **83.3** |
| Visual / structural attribute | 66.7 | 59.5 | 76.2 | 69.0 | 66.7 | 76.2 | **83.3** |
| Attribute + category | 66.7 | 59.5 | 73.8 | 66.7 | 64.3 | 76.2 | **83.3** |
| Context / function | 61.9 | 57.1 | 71.4 | 64.3 | 57.1 | 73.8 | **83.3** |
| Semantic paraphrase | 57.1 | 50.0 | 66.7 | 59.5 | 50.0 | 69.0 | **81.0** |
| Natural-language affordance | 54.8 | 47.6 | 61.9 | 54.8 | 47.6 | 66.7 | **78.6** |
| Spatial / contextual phrase | 50.0 | 40.5 | 59.5 | 52.4 | 50.0 | 64.3 | **76.2** |
| Contrastive phrase | 42.9 | 35.7 | 50.0 | 42.9 | 38.1 | **64.3** | **64.3** |

**Question families and references.** Table 17 gives an illustrative chair-based set with three questions per family. References must reflect the particular scene: spatial answers depend on the observed layout, and counting answers depend on annotated instances. Broad questions such as "Where can I sit?" can admit other seating objects, so the chair examples do not define universally exclusive answers. Reference construction must account for all valid scene-grounded responses.

**QA accuracy.** Following answer-correctness evaluation in VQA (Antol et al., 2015), we use a binary scene-grounded adaptation. For generated answer $a_n$ and reference answer $a_n^\star$,

$$\mathrm{QAAcc} = \frac{1}{N_\mathrm{q}} \sum_{n=1}^{N_\mathrm{q}} \mathbf{1}[a_n \text{ is semantically correct against } a_n^\star]\,. \tag{62}$$

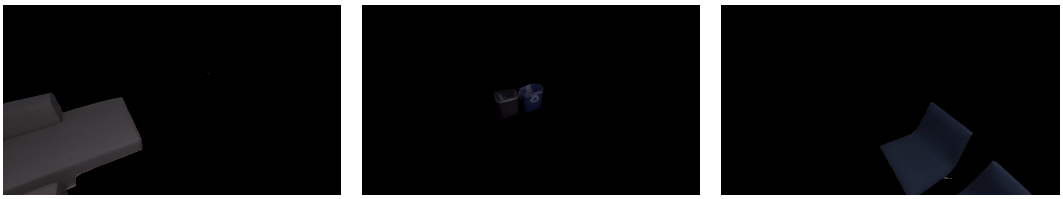
This differs from the original VQA consensus-based scoring rule.

**Semantic answer similarity.** Following sentence-embedding similarity evaluation (Reimers & Gurevych, 2019), we compare generated and reference answers using a fixed sentence encoder $f$:

$$\mathrm{SemSim} = \frac{1}{N_\mathrm{q}} \sum_{n=1}^{N_\mathrm{q}} \frac{f(a_n)^\top f(a_n^\star)}{\|f(a_n)\|_2 \|f(a_n^\star)\|_2}. \tag{63}$$

Table 16: Illustrative Top-3 retrieval accuracy (%) by query category. Each category contains 42 queries; entries are computed directly from integer success counts as $100n/42$. Best scores, including ties, are bold.

| Query category | CLIP Soft | CLIP Hard | PE Soft | PE Hard | ObjectGS | SceneSplat | MonoVoc |
|---|---|---|---|---|---|---|---|
| Exact class | 95.2 | 90.5 | **100.0** | 95.2 | **100.0** | **100.0** | **100.0** |
| Plural | 92.9 | 88.1 | **100.0** | 92.9 | 95.2 | 97.6 | **100.0** |
| Synonym | 85.7 | 81.0 | 92.9 | 88.1 | 85.7 | 92.9 | **95.2** |
| Category | 81.0 | 73.8 | 88.1 | 83.3 | 76.2 | 88.1 | **92.9** |
| Contextual category | 81.0 | 71.4 | 88.1 | 81.0 | 76.2 | 88.1 | **92.9** |
| Functional description | 76.2 | 66.7 | 81.0 | 76.2 | 66.7 | 81.0 | **92.9** |
| Affordance | 71.4 | 61.9 | 78.6 | 71.4 | 61.9 | 81.0 | **90.5** |
| Functional phrase | 73.8 | 64.3 | 81.0 | 76.2 | 66.7 | 81.0 | **90.5** |
| Visual / structural attribute | 78.6 | 71.4 | 85.7 | 81.0 | 76.2 | 85.7 | **92.9** |
| Attribute + category | 78.6 | 69.0 | 85.7 | 78.6 | 76.2 | 85.7 | **92.9** |
| Context / function | 76.2 | 69.0 | 83.3 | 78.6 | 69.0 | 83.3 | **92.9** |
| Semantic paraphrase | 73.8 | 64.3 | 81.0 | 73.8 | 66.7 | 81.0 | **90.5** |
| Natural-language affordance | 69.0 | 61.9 | 76.2 | 71.4 | 61.9 | 78.6 | **90.5** |
| Spatial / contextual phrase | 69.0 | 57.1 | 76.2 | 69.0 | 66.7 | 78.6 | **88.1** |
| Contrastive phrase | 66.7 | 59.5 | 78.6 | 69.0 | 64.3 | 88.1 | **92.9** |



(a) Sofa. (b) Bins. (c) Chairs.



(d) Plant. (e) Plant pot.

Figure 30: Additional examples of retrieved object.

The same evaluator is used across configurations. This measures semantic proximity rather than independently establishing factual correctness.

**LLM judge score.** Inspired by rubric-based LLM evaluation protocols, a fixed judge assesses correctness, relevance, and completeness on a five-point scale:

$$\text{Judge} = \frac{1}{N_{\text{q}}} \sum_{n=1}^{N_{\text{q}}} J(u_n, a_n, a_n^{\star}), \qquad J(\cdot) \in \{1, \ldots, 5\}, \tag{64}$$

where $u_n$ denotes the question. The judge and rubric remain identical across evaluated methods.

We use ChatGPT to assess generated answers against the manually authored scene-grounded references. The research team additionally reviews the judgments for correctness. The judge assesses answers after generation and does not construct the reference answers. ChatGPT provides an initial reference-based assessment of the generated answers. The research team reviews all assessments and corrects erroneous judgments against the input scene data and manually authored references. Reported judge scores therefore reflect human-reviewed assessments rather than unverified automated judgments.

Table 17: Illustrative QA templates for chairs. Expected answer forms are resolved against the particular scene.

| # | Question | Expected answer form |
|---|---|---|
| **Direct / lexical grounding** | | |
| 1 | Where are the chairs in the room? | Chair object(s) and location |
| 2 | Show me the seating chairs. | Chair object(s) |
| 3 | Locate the chairs for me. | Chair object(s) and location |
| **Affordance / functional reasoning** | | |
| 4 | Where can I sit? | Valid seating object(s) |
| 5 | What can I use if I need a seat? | Valid seating object(s) |
| 6 | Which furniture can support me while I am sitting? | Valid seating furniture |
| **Scenario-based reasoning** | | |
| 7 | I have been standing for a long time. Where can I rest? | Suitable object(s) |
| 8 | A guest has arrived and needs somewhere to sit. What should I offer them? | Suitable seating object(s) |
| 9 | I want to work at the desk for a while. What should I sit on? | Appropriate chair |
| **Spatial / relational QA** | | |
| 10 | Is there a chair near the desk? | Yes/no and supporting object |
| 11 | Where is the chair relative to the desk? | Scene-grounded relation |
| 12 | Which chair is closest to the desk? | Specific chair |
| **Counting / constraints / composition** | | |
| 13 | How many chairs are in the room? | Integer |
| 14 | Is there somewhere to sit other than the sofa or bed? | Yes/no and eligible object(s) |
| 15 | I need a personal seat near the desk, but I do not want to use the sofa. What should I use? | Object satisfying all conditions |

**Paraphrase consistency.** Inspired by invariance under meaning-preserving reformulations (Elazar et al., 2021), we define a benchmark-specific group-level criterion:

$$\mathrm{PC} = \frac{1}{N_{\mathrm{p}}} \sum_{g=1}^{N_{\mathrm{p}}} \mathbf{1}\left[ \begin{array}{c} \text{all questions in paraphrase group } g \\ \text{receive equivalent and correct answers} \end{array} \right]. \tag{65}$$

Each group contains semantically equivalent questions about the same scene. Membership in the same QA family alone does not imply paraphrase equivalence.

**Count error.** For the $N_{\mathrm{c}}$ counting questions, we report mean absolute error (Willmott & Matsuura, 2005) between predicted counts $\hat{c}_n$ and reference counts $c_n$:

$$\mathrm{CountMAE} = \frac{1}{N_{\mathrm{c}}} \sum_{n=1}^{N_{\mathrm{c}}} |\hat{c}_n - c_n|. \tag{66}$$

**Constraint satisfaction accuracy.** Motivated by strict verifiable-instruction evaluation (Zhou et al., 2023), we define a scene-grounded all-constraints criterion. For question $n$ with conditions $\{\kappa_{n,j}\}_{j=1}^{m_n}$,

$$\mathrm{ConstraintAcc} = \frac{1}{N_{\mathrm{con}}} \sum_{n=1}^{N_{\mathrm{con}}} \prod_{j=1}^{m_n} \mathbf{1}[a_n \models \kappa_{n,j}]. \tag{67}$$

An answer passes only if every specified positive, spatial, and exclusion condition is satisfied. This is our benchmark-specific adaptation, rather than the original IFEval metric.

**Answer generation.** QA uses retrieval followed by language generation. We compare the question embedding with the stored object embeddings to identify relevant objects. Information from the selected object and related database entries is then provided to LLaMA 3 8B to generate a textual answer.

Table 18: Illustrative QA accuracy (%) by question family. Each family contains 126 questions (42 objects × 3 questions). PE denotes Perception Encoder; Qwen denotes Qwen2.5-VL-3B-Instruct.

| QA family | CLIP Soft | CLIP Hard | PE Soft | PE Hard | ObjectGS | SceneSplat | Qwen | Ours |
|---|---|---|---|---|---|---|---|---|
| Direct / lexical grounding | 88.9 | 83.3 | 94.4 | 89.7 | 92.1 | 93.7 | 95.2 | **96.8** |
| Affordance / functional reasoning | 63.5 | 56.3 | 73.0 | 65.1 | 54.0 | 77.8 | 88.1 | **92.1** |
| Scenario-based reasoning | 57.1 | 50.8 | 66.7 | 59.5 | 46.0 | 73.0 | 84.1 | **89.7** |
| Spatial / relational QA | 59.5 | 53.2 | 69.8 | 62.7 | 57.9 | 75.4 | 77.8 | **85.7** |
| Counting / constraints / composition | 40.5 | 31.7 | 42.9 | 38.1 | 40.5 | 50.8 | 54.0 | **65.9** |
| **Overall** | 61.9 | 55.1 | 69.4 | 63.0 | 58.1 | 74.1 | 79.8 | **86.0** |

**Results by question family.** MonoVoc achieves the highest accuracy in all five reported families (Table 18). Its spatial accuracy reaches $85.7\%$, compared with $77.8\%$ for Qwen2.5-VL, while counting, constraint, and compositional accuracy reaches $65.9\%$ versus $54\%$. The latter remains the most challenging family for every evaluated configuration.

## J Object Removal Evaluation

**Evaluation protocol.** We evaluate object removal over $C = 15$ target-object removal cases. Each case contains aligned before- and after-edit renderings from the same camera viewpoints. Without ground-truth edited scenes, we evaluate removal through target-region changes and preservation of surrounding content.

For removal case $c$, let $\mathbf{I}_{c,f}^{\text{before}}(p)$ and $\mathbf{I}_{c,f}^{\text{after}}(p)$ denote RGB values rendered at the same pixel $p$ in frame $f$ before and after editing, and let $M_{c,f}(p)$ denote the corresponding target mask. The edit magnitude is

$$\Delta_{c,f}(p) = \left\|\mathbf{I}_{c,f}^{\text{before}}(p) - \mathbf{I}_{c,f}^{\text{after}}(p)\right\|_1 . \tag{68}$$

Each metric is first computed independently for every removal case. The value reported in Table 4 is the case-wise mean:

$$\overline{m} = \frac{1}{C}\sum_{c=1}^{C} m_c, \qquad C = 15, \tag{69}$$

where $m_c$ denotes the corresponding TEC, NTPE, ELR, BSR, or NT-SSIM score for case $c$.

Without ground-truth edited scenes, we evaluate removal through target-region changes and preservation of surrounding content. For camera frame $f$, let $\mathbf{I}_f^{\text{before}}(p)$ and $\mathbf{I}_f^{\text{after}}(p)$ denote RGB values rendered at the same pixel $p$ before and after editing. The binary mask $M_f(p)$ identifies the target in the original frame. We define the edit magnitude as

$$\Delta_f(p) = \left\|\mathbf{I}_f^{\text{before}}(p) - \mathbf{I}_f^{\text{after}}(p)\right\|_1 . \tag{70}$$

The following mask-guided measures summarize where changes occur and how strongly they affect non-target regions. They assess editing behavior without a clean edited reference; they do not independently establish the correctness of newly exposed surfaces.

**Target Edit Coverage (TEC).** TEC measures the fraction of target pixels whose appearance changes beyond a threshold $\delta$:

$$\text{TEC} = \frac{\sum_{f,p} M_f(p)\,\mathbf{1}[\Delta_f(p) > \delta]}{\sum_{f,p} M_f(p) + \epsilon}. \tag{71}$$

Higher coverage indicates that more of the visible target region was affected. This is a proxy for removal coverage, rather than proof of successful deletion: other appearance changes can also increase TEC.

**Non-Target Preservation Error (NTPE).** NTPE measures the average RGB change outside the target mask:

$$\text{NTPE} = \frac{\sum_{f,p}(1 - M_f(p))\,\Delta_f(p)}{\sum_{f,p}(1 - M_f(p)) + \epsilon}. \tag{72}$$

Lower values indicate better preservation of the originally visible non-target region. Its magnitude depends on the RGB scale used in Equation 70.

**Edit Leakage Ratio (ELR).** ELR measures the proportion of total edit magnitude outside the target:

$$\mathrm{ELR} = \frac{\sum_{f,p}(1 - M_f(p))\,\Delta_f(p)}{\sum_{f,p}\Delta_f(p) + \epsilon}. \tag{73}$$

Unlike NTPE, which averages change over non-target pixels, ELR expresses how much of the overall edit extends beyond the mask. Lower values indicate more concentrated editing.

**Boundary Spill Ratio (BSR).** To allow a small boundary tolerance, let $M_f^{+r}$ be the target mask dilated by $r$ pixels. We measure spill beyond this expanded region as

$$\mathrm{BSR} = \frac{\sum_{f,p}(1 - M_f^{+r}(p))\,\Delta_f(p)}{\sum_{f,p}\Delta_f(p) + \epsilon}. \tag{74}$$

Lower values indicate less change outside the target and its boundary neighborhood. Since dilation includes the original mask, $\mathrm{BSR} \leq \mathrm{ELR}$ under the same evaluation conditions.

**Non-Target SSIM (NT-SSIM).** We adapt structural similarity (Wang et al., 2004) by applying the same non-target mask to both renderings and averaging over $F$ frames:

$$\text{NT-SSIM} = \frac{1}{F}\sum_{f=1}^{F}\mathrm{SSIM}\big(\mathbf{I}_f^{\mathrm{before}} \odot (1 - M_f), \mathbf{I}_f^{\mathrm{after}} \odot (1 - M_f)\big)\,. \tag{75}$$

Here, the mask is broadcast over RGB channels. Higher scores indicate greater structural preservation in the masked-image comparison. Because conventional SSIM uses local windows, this implementation also includes shared zero-valued target regions and boundary effects; it is not a strict average over exclusively non-target SSIM windows.

**Joint interpretation.** TEC, ELR, and BSR are reported as percentages, while NTPE and NT-SSIM retain their native scales. The metrics are interpreted jointly: making no edit yields low leakage and high preservation but zero target coverage. Qualitative comparisons complement these scores by showing residual appearance and local reconstruction quality in the edited regions.

**Additional examples.** Figures 31–33 compare SceneSplat, ObjectGS, and MonoVoc on small-table and vase removal. Each example includes the original scene, edited renderings, and enlarged views of the target region.

## K Human Evaluation of Map Quality

We conducted a human evaluation with 34 participants to examine how clearly the 3D Gaussian maps produced by MonoVoc, ObjectGS, and SceneSplat communicate scene content. All participants rated all three methods on five questions using integer scores from 1 to 5, with higher scores indicating a more favorable assessment. They also selected an overall preferred method or "no clear preference."

**Participant recruitment.** Participants were recruited through flyers and in-person outreach. Individuals chose to participate based on their interest, yielding a self-selected volunteer sample rather than a randomly sampled population. The findings therefore describe the participating group and should not be interpreted as representative of a broader population.

### K.1 Questionnaire and Rating Anchors

The questionnaire assessed scene structure, object distinguishability, consistency across viewpoints, visual clutter, and usefulness for scene understanding. The questions and rating anchors were:

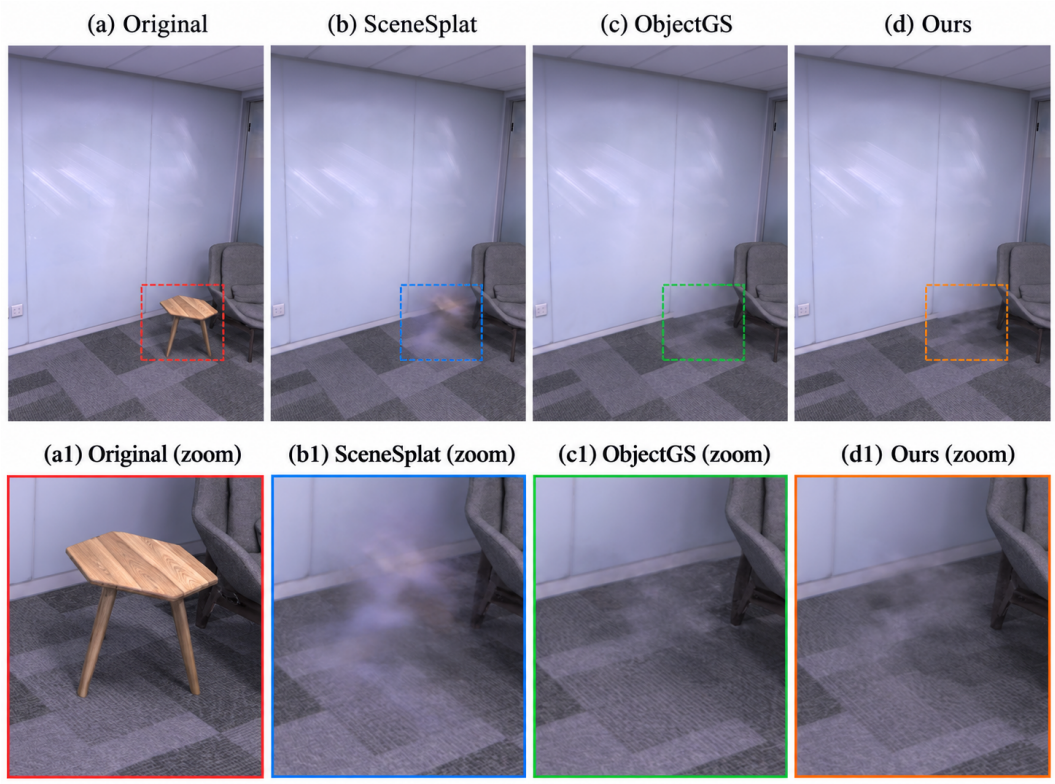



Figure 31: Small-table removal comparison. Enlarged views show the edited floor region and neighboring chair.

1. **Q1: Scene structure preservation.** How well does this 3D representation preserve the important structure and major objects of the original scene?

   *Ratings:* 1—Very poor: important scene structure or objects are difficult to recognize; 2—Poor: several important elements are unclear or missing; 3—Acceptable: the main scene structure is understandable; 4—Good: most important structures and objects are clearly preserved; 5—Excellent: scene structure and major objects are immediately recognizable.

2. **Q2: Object distinguishability.** How easily can you distinguish the major objects as separate entities in this 3D representation?

   *Ratings:* 1—Very difficult; 2—Difficult; 3—Moderately clear; 4—Easy; 5—Very easy.

3. **Q3: Cross-view consistency.** How consistently can you recognize the same objects across different viewpoints of the scene?

   *Ratings:* 1—Very inconsistent: objects are difficult to associate between views; 2—Inconsistent; 3—Moderately consistent; 4—Consistent; 5—Very consistent: object identity is clear across all views.

4. **Q4: Clarity versus visual clutter.** How well does this representation balance preserving useful scene information while avoiding unnecessary visual clutter?

   *Ratings:* 1—Very poor: either excessive clutter or important information is lost; 2—Poor; 3—Acceptable balance; 4—Good balance: a clear scene with little unnecessary clutter; 5—Excellent balance: compact and clear while retaining important scene content.

5. **Q5: Usefulness for understanding the scene.** Considering scene structure, object distinguishability, consistency, and visual clarity, how useful is this representation for quickly understanding the 3D scene?

   *Ratings:* 1—Not useful; 2—Slightly useful; 3—Moderately useful; 4—Very useful; 5—Extremely useful.

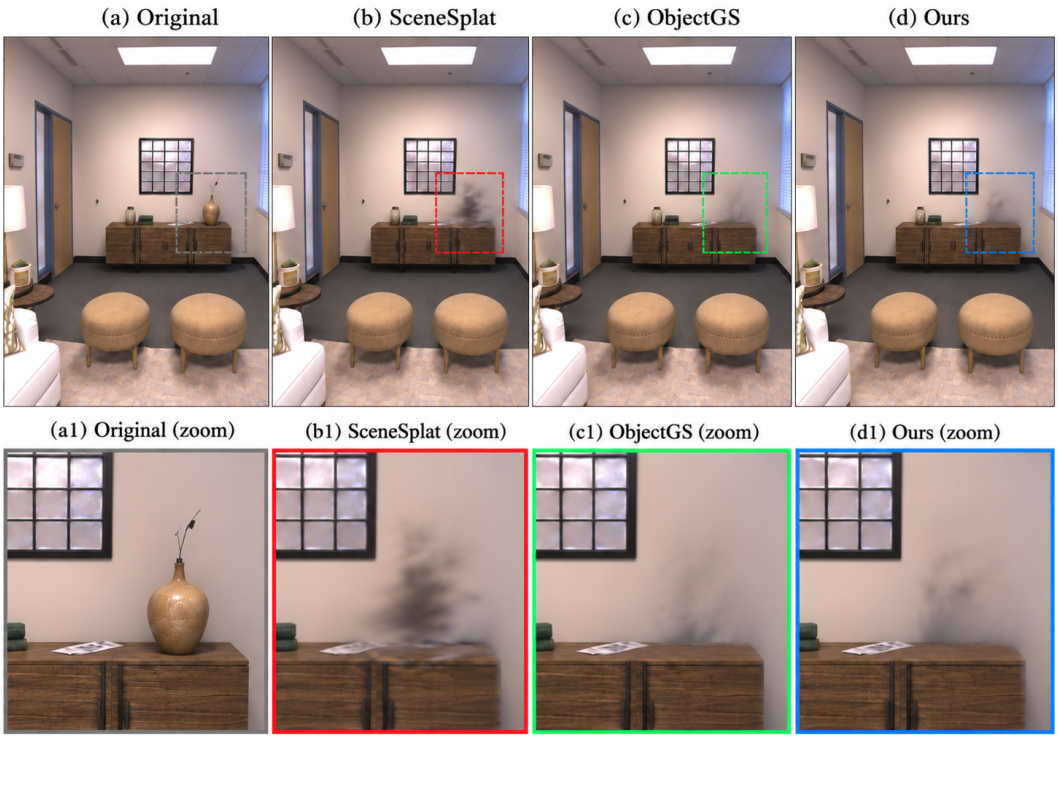



Figure 32: Vase removal comparison, showing the edited region above the cabinet.

**Anonymous and randomized presentation.** Participants viewed outputs under anonymous labels A, B, and C without being told their corresponding method identities. Presentation order was randomized across question sets and participants. Ratings and preferences were recorded using these labels and mapped to methods during analysis. Identity concealment and randomization were intended to reduce method-name and presentation-order biases. Visual differences were retained because perceived map quality was the subject of evaluation. Identity concealment does not guarantee that participants could not infer which method produced an output.

### K.2 Aggregation and Detailed Results

Let $x_{r,m,q}$ denote participant $r$'s rating of method $m$ on question $q$. Each participant's overall score is the mean of their five ratings:

$$\bar{x}_{r,m} = \frac{1}{5}\sum_{q=1}^{5} x_{r,m,q}. \tag{76}$$

The overall method mean and sample standard deviation are computed across the 34 participant-level scores:

$$\begin{aligned} \bar{x}_m &= \frac{1}{34}\sum_{r=1}^{34} \bar{x}_{r,m}, \\ s_m &= \sqrt{\frac{1}{33}\sum_{r=1}^{34}(\bar{x}_{r,m} - \bar{x}_m)^2}. \end{aligned} \tag{77}$$

Thus, the overall SD describes variation between participants' five-item averages, rather than variation across pooled individual ratings. Per-question SDs are calculated separately across participants for each question and method.

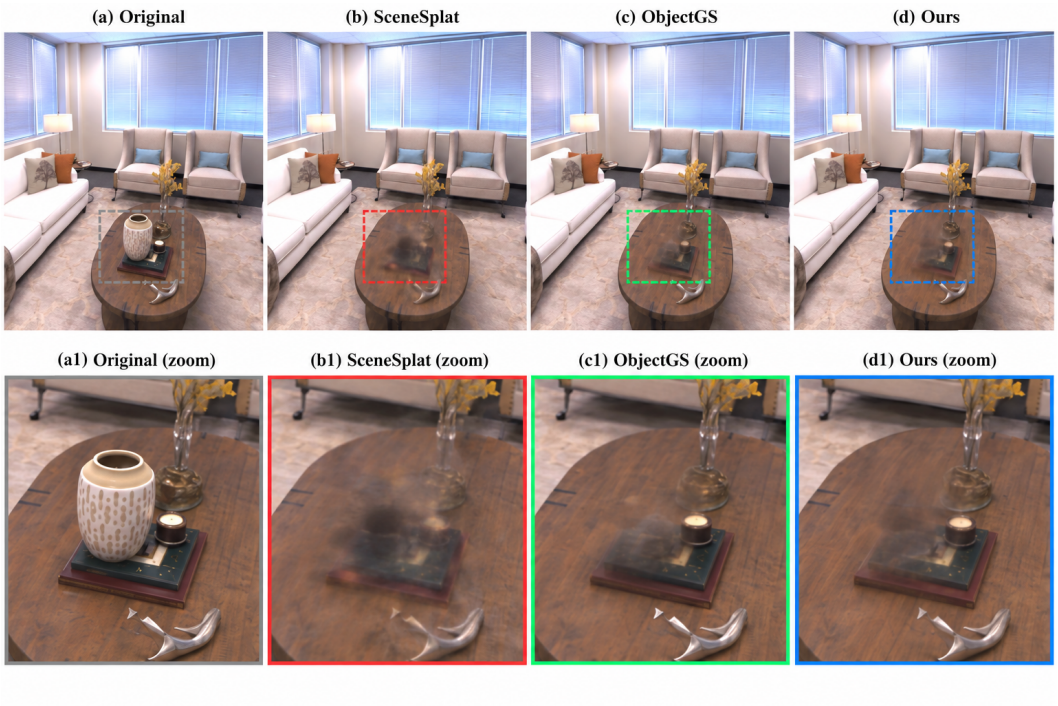



Figure 33: Vase removal comparison, with enlarged views of the tabletop and surrounding objects.

Table 19: Per-question ratings (mean $\pm$ sample SD) across 34 participants. Higher means are better. Best means are bold and second-best distinct means are underlined; tied means share formatting.

| Question | MonoVoc | ObjectGS | SceneSplat |
|---|---|---|---|
| Q1: Scene structure | $\mathbf{4.00} \pm 0.65$ | $\mathbf{4.00} \pm 0.65$ | $\underline{3.85} \pm 0.74$ |
| Q2: Object distinguishability | $\mathbf{4.12} \pm 0.73$ | $\underline{4.03} \pm 0.67$ | $3.88 \pm 0.73$ |
| Q3: Cross-view consistency | $\mathbf{4.00} \pm 0.65$ | $\underline{3.88} \pm 0.73$ | $3.82 \pm 0.76$ |
| Q4: Clarity versus clutter | $\mathbf{4.21} \pm 0.77$ | $\underline{3.88} \pm 0.73$ | $3.71 \pm 0.80$ |
| Q5: Overall usefulness | $\mathbf{4.21} \pm 0.73$ | $\underline{3.88} \pm 0.69$ | $3.74 \pm 0.83$ |

MonoVoc receives the highest mean on four questions and ties with ObjectGS on scene structure preservation (Table 19). Its largest mean advantages over ObjectGS occur on clarity versus clutter and overall usefulness, scoring 4.21 versus 3.88 on both questions. These question-level comparisons are descriptive; the statistical tests below assess the participant-level overall scores.

The overall scores are $4.11 \pm 0.28$ for MonoVoc, $3.94 \pm 0.34$ for ObjectGS, and $3.80 \pm 0.32$ for SceneSplat, where the reported dispersion is sample SD. We additionally calculate approximate 95% confidence intervals for the method means using

$$\bar{x}_m \pm t_{0.975,33} \frac{s_m}{\sqrt{34}}, \qquad t_{0.975,33} \approx 2.0345. \tag{78}$$

These intervals quantify uncertainty in the mean across participants, whereas SD describes variation in their scores. They are not confidence intervals for paired method differences.

### K.3 Paired Statistical Comparisons

Because every participant rated all three methods, we compare overall scores using paired differences $d_r = \bar{x}_{r,\text{MonoVoc}} - \bar{x}_{r,m}$. We conduct exploratory two-sided Wilcoxon signed-rank tests, excluding zero differences and assigning average ranks to tied absolute differences. Exact conditional $p$-values are computed from the sign-permutation distribution of the observed ranks. This accounts for ties without using the tie-free Wilcoxon reference distribution.

Table 20: Overall ratings with approximate Student-$t$ 95% confidence intervals, reported as mean ± CI half-width.

| Method | Mean ± 95% CI half-width |
|---|---|
| MonoVoc | $\mathbf{4.11} \pm 0.10$ |
| ObjectGS | $\underline{3.94} \pm 0.12$ |
| SceneSplat | $3.80 \pm 0.11$ |

Table 21: Paired comparisons of participant-level overall scores. $\Delta$ is the mean MonoVoc-minus-baseline difference. The $+/-/=$ column counts positive, negative, and zero differences. Zero differences are excluded from the signed-rank tests.

| Comparison | $+/-/=$ | $\Delta$ | $W$ | Raw $p$ | Holm $p$ |
|---|---|---|---|---|---|
| MonoVoc–SceneSplat | $22/4/8$ | 0.306 | 31.5 | $6.82 \times 10^{-5}$ | $1.36 \times 10^{-4}$ |
| MonoVoc–ObjectGS | $17/11/6$ | 0.171 | 108.0 | 0.0272 | 0.0272 |

Table 21 reports the smaller signed-rank sum $W$, mean paired differences, and raw and Holm-adjusted $p$-values. Holm correction covers the two comparisons of MonoVoc against ObjectGS and SceneSplat. Both comparisons have adjusted $p < 0.05$ under this analysis.

Signed-rank inference assumes independent participant pairs and symmetry of the paired-difference distribution under the null. As a sensitivity analysis, exact two-sided sign tests, which use only the direction of nonzero differences, give unadjusted $p = 0.000534$ against SceneSplat and $p = 0.345$ against ObjectGS. Thus, evidence for the ObjectGS comparison depends more on the magnitudes of the paired differences and is less robust to test choice. These overall comparisons do not establish significance for individual question-level differences.

### K.4 Overall Preferences

Of the 34 participants, 18 selected MonoVoc, nine selected ObjectGS, six selected SceneSplat, and one expressed no clear preference. The main-body preference table uses all 34 responses: the corresponding shares are 52.9%, 26.5%, 17.6%, and 2.9%. Rounded percentages may not sum to exactly 100%.

For a separate exploratory preference test, we condition on the 33 participants who selected a method. Under an equal-choice null among the three methods,

$$X \sim \mathrm{Binomial}(33, 1/3). \tag{79}$$

MonoVoc receives $18/33 = 54.5\%$ of these selections, giving an exact two-sided binomial $p$-value of 0.0150. This tests MonoVoc's share against the specified equal-choice null; it does not establish a significant pairwise preference difference against each baseline separately. The preference test is reported separately from the two Holm-adjusted rating comparisons.

### K.5 Study Visualizations

The figures summarize question-level ratings, participant-level overall scores, preferences, and response distributions. Question-level error bars show sample SD across participants; overall-score error bars show the approximate 95% confidence intervals defined in Equation 78.

The study characterizes perceived map quality for the self-selected participants and scenes evaluated. It does not establish generalization to other populations, scene collections, or viewing conditions. Evidence for the overall rating difference against ObjectGS depends on the statistical test, as discussed above.

## L Segmentation Quality and a Proposed Sensitivity Diagnostic

MonoVoc uses 2D segmentation maps as semantic observations, so errors in these inputs can propagate into its Gaussian assignments. This appendix explains the failure mechanism and outlines a

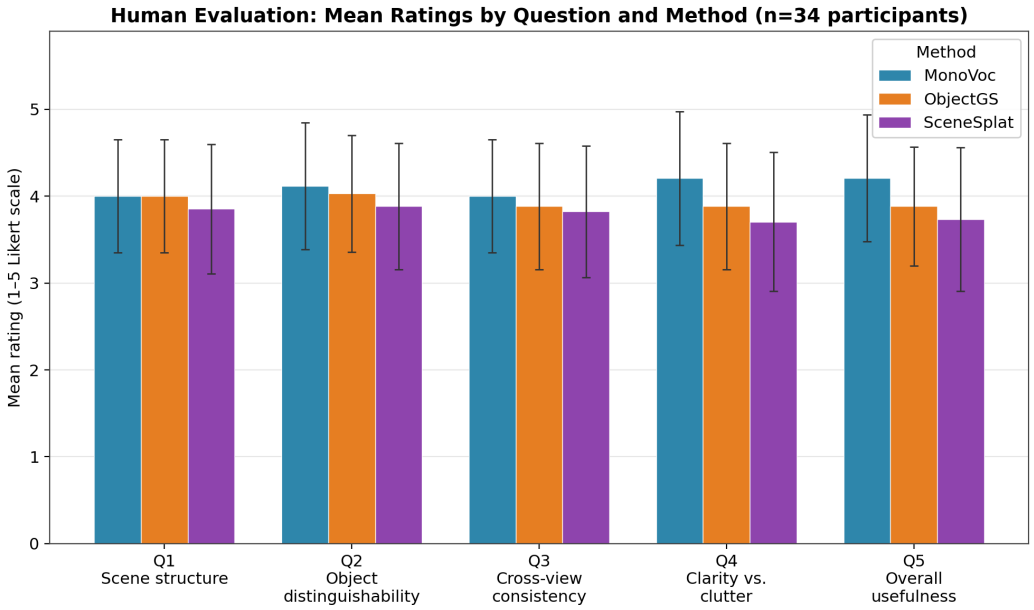
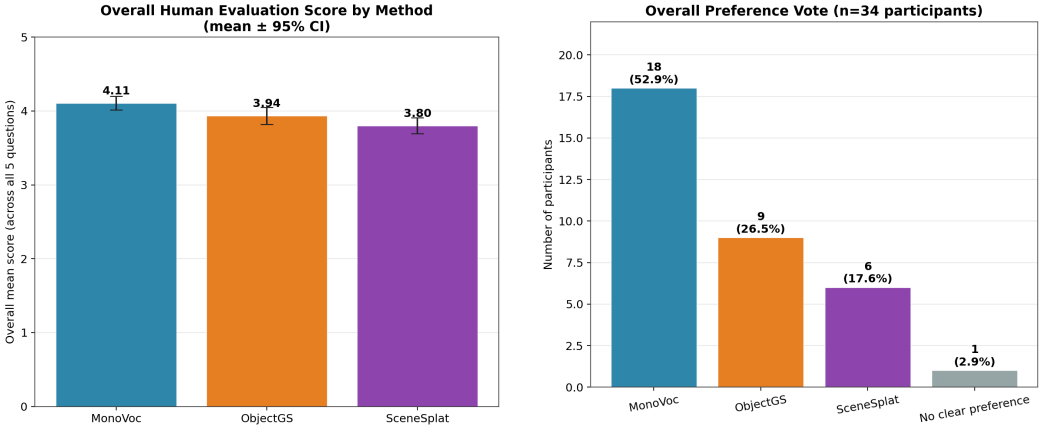



Figure 34: Mean ratings across the five questions. Error bars show sample SD across 34 participants.




(a) Overall means with approximate 95% Student-$t$ confidence intervals.

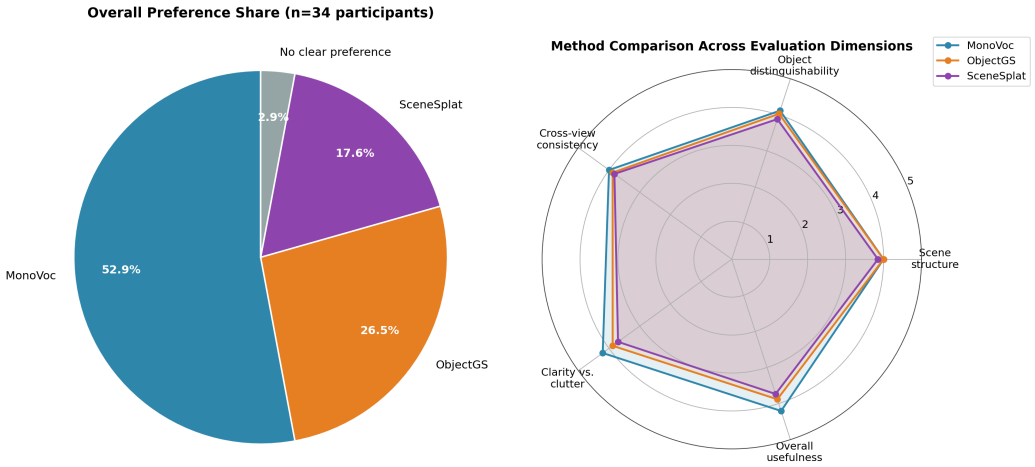
(b) Preference counts and shares of all 34 responses.

Figure 35: Overall ratings and preferences in the human evaluation of the generated 3D Gaussian maps.

diagnostic for measuring sensitivity to segmentation corruption. The diagnostic is not part of the reported quantitative evaluation; no corruption sweep or measured degradation values are reported.

**Illustrative corruption settings.** Potential perturbations include random label noise, boundary-focused label noise, mixed boundary colors, and region artifacts. Random label noise replaces a fraction $\rho$ of pixels with another valid label. Boundary-focused noise restricts replacements to object edges. Mixed boundary colors interpolate between palette entries, while region artifacts introduce holes or isolated mislabeled areas. These settings describe possible diagnostic inputs rather than an evaluated parameter sweep.

**Proposed degradation measure.** A controlled diagnostic would hold Gaussian geometry, camera poses, rendering settings, and quantization parameters fixed, then rerun SCD and confidence-gated




(a) Preference shares among all 34 participants. (b) Mean ratings across the five evaluation dimensions.

Figure 36: Participant preferences and perceived strengths of the three map representations.

quantization with clean and corrupted semantic observations. Both outputs would be evaluated against the same unchanged, clean reference labels on identical views. The resulting degradation could be summarized as

$$\Delta\mathrm{mIoU} = \mathrm{mIoU}_{\mathrm{clean}} - \mathrm{mIoU}_{\mathrm{corrupted}}, \tag{80}$$

where positive values indicate lower agreement with the reference after corruption. This definition specifies a possible evaluation measure; it is not a reported experimental result. Fixing the reconstruction would isolate sensitivity during semantic recovery and assignment, rather than assess the effect of corrupted supervision on reconstruction itself.

**Failure mechanism.** SCD aggregates semantic observations using compositing weights. Incorrect observations can shift recovered colors toward other palette entries and affect subsequent object assignments. Errors near overlapping footprints can interact with color mixing, while errors repeated across frames can produce consistently incorrect evidence. Confidence gating does not guarantee correction of such errors. These are mechanistic considerations, not measured rankings of the proposed corruption settings.

**Practical implications.** Segmentation inputs should preserve discrete labels and maintain consistent object identities across frames. ASPT improves palette separation but does not correct incorrect labels. MonoVoc can accept compatible predicted segmentations, although this capability alone does not establish robustness to their errors.

## M Mathematical Foundations of Semantic Color Deblending

SCD converts composited semantic observations into per-Gaussian color evidence using a normalized weighted accumulation. The derivations below establish why the rendering weights can be used as mixing coefficients, which objective the SCD estimator solves, and under what conditions its output recovers the underlying semantic colors. They also distinguish the implemented estimator from a global inverse and characterize its approximation error. These distinctions are necessary because an exact forward model does not imply that a computationally inexpensive recovery rule is an exact inverse.

### M.1 Notation and Scope

Let $p \in \mathcal{P}$ index a pixel–frame observation and let $i \in \{1, \ldots, N\}$ identify a Gaussian. Its semantic coefficients are $\mathbf{s}_i \in \mathbb{R}^3$, and the known background color is $\mathbf{b} \in \mathbb{R}^3$. Geometry, opacity, camera

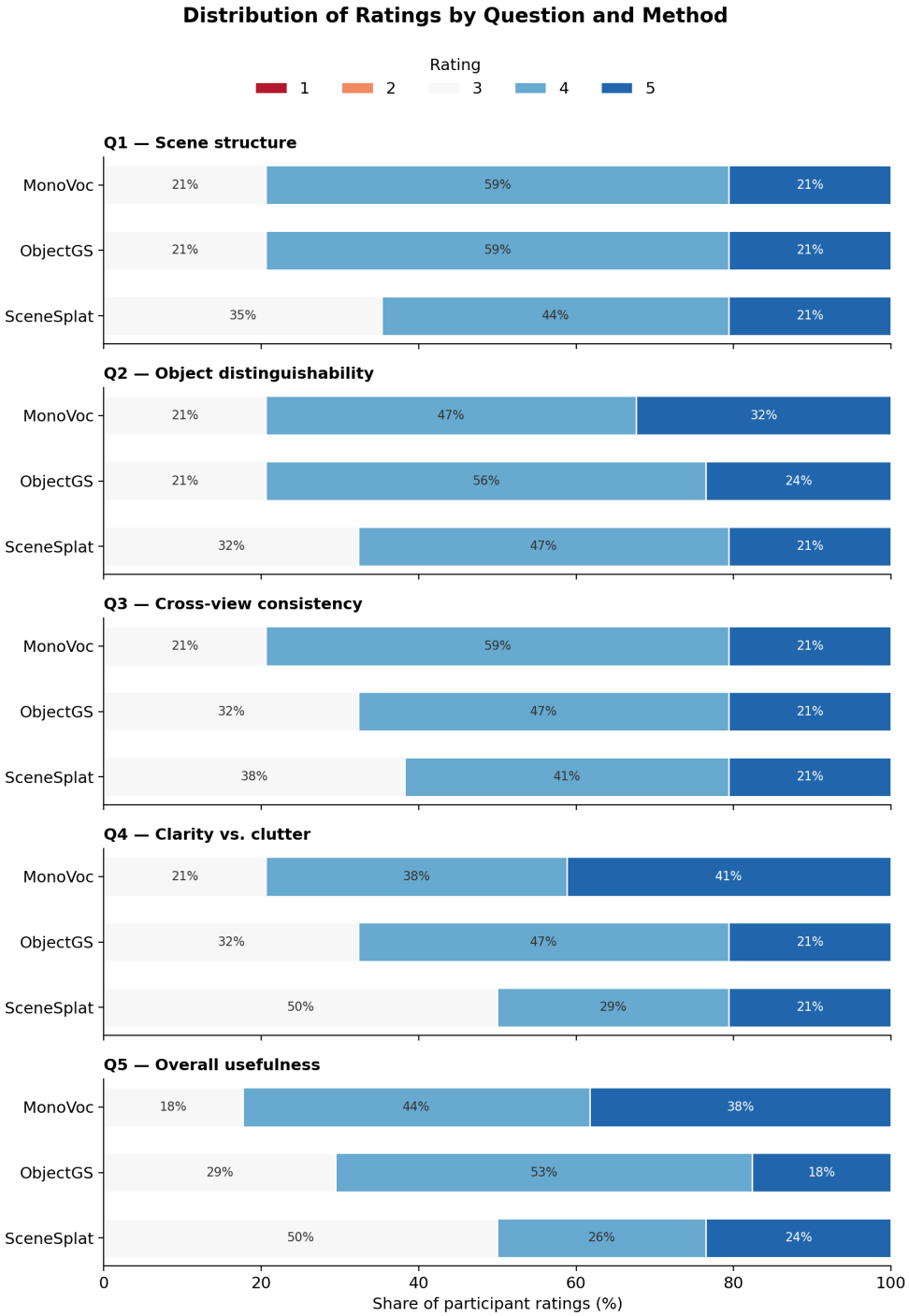



Figure 37: Distribution of participant ratings for each question and method. Stacked segments represent scores from 1 to 5; displayed percentages are rounded.

parameters, and depth ordering are fixed throughout the analysis. We write $j \prec_p i$ when Gaussian $j$ precedes Gaussian $i$ along the ray corresponding to observation $p$, and define

$$T_{i,p} = \prod_{j \prec_p i} (1 - \alpha_{j,p}), \qquad w_{i,p} = \alpha_{i,p} T_{i,p}, \qquad T_{\mathrm{bg},p} = \prod_{j=1}^{N} (1 - \alpha_{j,p}). \tag{81}$$

An empty product equals one. Within proofs concerning a single observation, indices are ordered from front to back.

We allow $\alpha_{i,p} \in [0, 1]$ in the theoretical model so that fully opaque, one-hot mixing is included. Implementations that restrict $\alpha_{i,p} < 1$ approach this case only as a limit. All exact-recovery statements additionally assume observations satisfy the stated forward model; they do not establish that arbitrary segmentation maps admit an exact representation.

For each Gaussian, $\mathcal{W}_i \subseteq \mathcal{P}$ denotes the observations retained by its EWR-defined windows. Windowing restricts the observations accumulated for that Gaussian. The weights in those observations retain their compositing definition, including transmittance from preceding Gaussians. In particular, restricting an accumulation window does not justify discarding foreground occluders when computing transmittance.

### M.2 Dependence of Rendering on Semantic Colors

**Lemma 1 (Semantic-independence of opacity).** The recovery stage treats rendering weights as known quantities. This requires the projected opacity to remain independent of the semantic coefficients being recovered. The following statement makes that assumption explicit.

*Statement.* Let $\boldsymbol{\theta}_i$ collect the fixed non-semantic parameters of Gaussian $i$, and suppose

$$\alpha_{i,p} = g_i(p; \boldsymbol{\theta}_i) \tag{82}$$

does not depend on any $\mathbf{s}_k$. Then

$$\frac{\partial \alpha_{i,p}}{\partial \mathbf{s}_k} = \mathbf{0} \qquad \text{for every } i, k, p. \tag{83}$$

*Proof.* Since $p$ and $\boldsymbol{\theta}_i$ are fixed and $g_i$ has no semantic-color argument,

$$\frac{\partial \alpha_{i,p}}{\partial \mathbf{s}_k} = \frac{\partial}{\partial \mathbf{s}_k} g_i(p; \boldsymbol{\theta}_i) = \mathbf{0}. \tag{84}$$

This result concerns SCD with fixed reconstruction parameters. It does not assert that geometry and semantics are independent during their earlier joint optimization in the reconstruction backbone.

**Lemma 2 (Semantic-independence of transmittance and weights).** Opacity independence extends to visibility and compositing weights. Establishing this extension permits the same weights to be used throughout semantic recovery without differentiating through a changing visibility model.

*Statement.* Under Lemma 1,

$$\frac{\partial T_{i,p}}{\partial \mathbf{s}_k} = \frac{\partial w_{i,p}}{\partial \mathbf{s}_k} = \frac{\partial T_{\mathrm{bg},p}}{\partial \mathbf{s}_k} = \mathbf{0}. \tag{85}$$

*Proof.* Applying the product rule gives

$$\frac{\partial T_{i,p}}{\partial \mathbf{s}_k} = -\sum_{\ell \prec_p i} \left( \prod_{\substack{j \prec_p i \\ j \neq \ell}} (1 - \alpha_{j,p}) \right) \frac{\partial \alpha_{\ell,p}}{\partial \mathbf{s}_k} = \mathbf{0}. \tag{86}$$

The same argument applies to $T_{\mathrm{bg},p}$. Finally,

$$\frac{\partial w_{i,p}}{\partial \mathbf{s}_k} = T_{i,p} \frac{\partial \alpha_{i,p}}{\partial \mathbf{s}_k} + \alpha_{i,p} \frac{\partial T_{i,p}}{\partial \mathbf{s}_k} = \mathbf{0}. \tag{87}$$

Consequently, semantic recovery changes the semantic coefficients while the mixing coefficients remain fixed. This is the basis for the linear-system interpretation below.

**Theorem 3 (Affine rendering and the stacked linear model).** Alpha compositing is nonlinear in several reconstruction parameters, but its dependence on semantic colors is simpler. Identifying that dependence separates the forward rendering model from the approximation used to recover colors.

*Statement.* For fixed rendering weights, the forward model

$$\mathbf{C}_p = \sum_{i=1}^{N} w_{i,p}\mathbf{s}_i + T_{\mathrm{bg},p}\mathbf{b} \tag{88}$$

is affine in the semantic coefficients. Defining

$$\mathbf{y}_p = \mathbf{C}_p - T_{\mathrm{bg},p}\mathbf{b} \tag{89}$$

gives

$$\mathbf{y}_p = \sum_{i=1}^{N} w_{i,p}\mathbf{s}_i. \tag{90}$$

Let $\mathbf{S} \in \mathbb{R}^{N\times 3}$ and $\mathbf{Y} \in \mathbb{R}^{|\mathcal{P}|\times 3}$ have rows $\mathbf{s}_i^\top$ and $\mathbf{y}_p^\top$, respectively. With $\mathbf{W}_{p,i} = w_{i,p}$,

$$\mathbf{Y} = \mathbf{W}\mathbf{S}. \tag{91}$$

*Proof.* By Lemma 2, all coefficients $w_{i,p}$ and the background term are independent of the unknown semantic colors. Thus Equation 88 is a linear combination of those colors plus a constant, and is therefore affine. Subtracting the background term gives Equation 90. Stacking its transpose over observations gives Equation 91.

The stacked system describes the ideal inverse problem. Its linearity alone does not guarantee a unique solution: different Gaussians may have indistinguishable observation weights, or receive no observations. SCD avoids solving this potentially coupled system and instead computes local estimates.

### M.3 Compositing Weights as Mixing Coefficients

**Lemma 4 (Recursive transmittance and its decrement).** The transmittance recursion connects visibility to the contribution of each Gaussian. It also supplies the telescoping identity needed to establish conservation of the total mixing weight.

*Statement.* At a fixed observation, with front-to-back indexing and $T_{1,p} = 1$,

$$T_{i+1,p} = T_{i,p}(1-\alpha_{i,p}), \qquad w_{i,p} = T_{i,p} - T_{i+1,p}. \tag{92}$$

*Proof.* Factoring the last term from the transmittance product gives

$$T_{i+1,p} = \prod_{j=1}^{i}(1-\alpha_{j,p}) = T_{i,p}(1-\alpha_{i,p}). \tag{93}$$

Therefore,

$$T_{i,p} - T_{i+1,p} = T_{i,p}\alpha_{i,p} = w_{i,p}. \tag{94}$$

Thus $w_{i,p}$ is precisely the amount of transmittance consumed by Gaussian $i$ at observation $p$. It accounts for both its projected opacity and the visibility remaining after preceding Gaussians.

**Theorem 5 (Partition of unity and exact convex mixing).** Using compositing weights as semantic evidence requires more than nonnegativity: their relationship to the residual background must also be specified. This result establishes that the forward renderer forms a convex mixture before background subtraction.

*Statement.* For $\alpha_{i,p} \in [0,1]$,

$$0 \le T_{i+1,p} \le T_{i,p} \le 1, \qquad 0 \le w_{i,p} \le 1, \tag{95}$$

and

$$\sum_{i=1}^{N} w_{i,p} + T_{\mathrm{bg},p} = 1. \tag{96}$$

Consequently, Equation 88 is an exact convex mixture of the semantic coefficients and background.

*Proof.* Every factor $1 - \alpha_{i,p}$ lies in $[0, 1]$, so the transmittance products lie in $[0, 1]$ and cannot increase along the ray. Since $w_{i,p} = \alpha_{i,p} T_{i,p}$, the weights are also nonnegative and bounded by one. By Lemma 4,

$$\sum_{i=1}^{N} w_{i,p} = \sum_{i=1}^{N} (T_{i,p} - T_{i+1,p}) = T_{1,p} - T_{N+1,p} = 1 - T_{\mathrm{bg},p}. \tag{97}$$

The coefficients are therefore nonnegative and sum to one when the background is included.

This establishes an exact property of the forward model, not exact recovery by SCD. After background subtraction, the Gaussian weights sum to $1 - T_{\mathrm{bg},p}$, which can be less than one. That residual transparency matters when interpreting the recovered colors.

### M.4 The Windowed SCD Estimator

**Theorem 6 (Strict convexity and the weighted-average solution).** The SCD update has a precise optimization interpretation even though it does not invert the full renderer. Its objective fits one constant semantic color to the observations associated with a Gaussian, weighting each observation by that Gaussian's rendering contribution.

*Statement.* For Gaussian $i$, define

$$W_i = \sum_{p \in \mathcal{W}_i} w_{i,p}, \qquad J_i(\mathbf{s}) = \sum_{p \in \mathcal{W}_i} w_{i,p} \|\mathbf{s} - \mathbf{y}_p\|_2^2. \tag{98}$$

The objective is convex, with Hessian

$$\nabla^2 J_i(\mathbf{s}) = 2 W_i \mathbf{I}_3. \tag{99}$$

If $W_i > 0$, its unique minimizer is

$$\hat{\mathbf{s}}_i = \frac{\sum_{p \in \mathcal{W}_i} w_{i,p} \mathbf{y}_p}{W_i}. \tag{100}$$

*Proof.* Expanding and differentiating the quadratic gives

$$\nabla J_i(\mathbf{s}) = 2 W_i \mathbf{s} - 2 \sum_{p \in \mathcal{W}_i} w_{i,p} \mathbf{y}_p. \tag{101}$$

Differentiating again yields Equation 99. Nonnegative weights imply $W_i \geq 0$, so the Hessian is positive semidefinite. When $W_i > 0$, it is positive definite and the objective is strictly convex. Setting the gradient to zero gives Equation 100.

This proves that windowed accumulation exactly solves the stated per-Gaussian objective. It does not prove that the minimizer equals the generating semantic coefficient under overlap. If $W_i = 0$, all terms vanish and the objective provides no information about the Gaussian's semantic color.

**Corollary 7 (Accumulation and batch invariance).** The closed-form solution permits streaming computation. Observations need not be stored together, which is important when processing many pixels and frames under a limited memory budget.

*Statement.* Define a vector accumulator and a scalar accumulator:

$$\mathbf{a}_i = \sum_{p \in \mathcal{W}_i} w_{i,p} \mathbf{y}_p, \qquad u_i = \sum_{p \in \mathcal{W}_i} w_{i,p}. \tag{102}$$

For $u_i > 0$, $\hat{\mathbf{s}}_i = \mathbf{a}_i / u_i$. Partitioning the retained observations into disjoint batches does not change this estimate in exact arithmetic.

*Proof.* For a disjoint partition $\mathcal{W}_i = \bigcup_\ell \mathcal{B}_{i,\ell}$, finite-sum associativity gives

$$\begin{aligned} \mathbf{a}_i &= \sum_{\ell} \sum_{p \in \mathcal{B}_{i,\ell}} w_{i,p} \mathbf{y}_p, \\ u_i &= \sum_{\ell} \sum_{p \in \mathcal{B}_{i,\ell}} w_{i,p}. \end{aligned} \tag{103}$$

Substitution into Equation 100 proves the result.

Consequently, batch size can control temporary memory without changing the mathematical objective, provided the retained observations and their weights are unchanged. Floating-point summation order can introduce small numerical differences. Changing the EWR windows, however, changes the observation set and can change the estimate.

### M.5 Recovery Conditions and Approximation Error

**Theorem 8 (Exact recovery under one-hot mixing).** An idealized recovery case provides a useful consistency check. The relevant condition is one-hot contribution weight, which is stronger than having only one Gaussian with nonzero opacity at a pixel.

*Statement.* Assume Equation 90 and that, for each observation $p$, there is an index $\iota(p)$ such that

$$w_{\iota(p),p} = 1, \qquad w_{j,p} = 0 \quad \text{for } j \neq \iota(p). \tag{104}$$

Then Equation 100 satisfies $\hat{\mathbf{s}}_i = \mathbf{s}_i$ for every Gaussian with $W_i > 0$.

*Proof.* Under one-hot mixing, $\mathbf{y}_p = \mathbf{s}_{\iota(p)}$. Let

$$\mathcal{W}_i^+ = \{p \in \mathcal{W}_i : \iota(p) = i\}. \tag{105}$$

Only these observations contribute to Gaussian $i$, and each has unit weight. Therefore,

$$\hat{\mathbf{s}}_i = \frac{\sum_{p \in \mathcal{W}_i^+} \mathbf{s}_i}{|\mathcal{W}_i^+|} = \mathbf{s}_i. \tag{106}$$

Non-overlap alone is insufficient. For example, if a single Gaussian contributes with constant weight $a \in (0, 1)$ and all other Gaussian weights are zero, then $\mathbf{y}_p = a\mathbf{s}_i$ and SCD returns $a\mathbf{s}_i$. Exact one-hot recovery therefore represents a fully opaque limiting case when the renderer caps opacity strictly below one.

**Theorem 9 (Overlap and transparency decomposition).** Under general mixing, a Gaussian accumulates observations that also contain other Gaussians' contributions. The following decomposition identifies this contamination and separates it from attenuation associated with residual transparency.

*Statement.* Assume Equation 90. Define

$$H_{ij} = \sum_{p \in \mathcal{W}_i} w_{i,p} w_{j,p}, \qquad h_i = \sum_{p \in \mathcal{W}_i} w_{i,p} T_{\mathrm{bg},p}. \tag{107}$$

For $W_i > 0$,

$$\hat{\mathbf{s}}_i = \frac{H_{ii}}{W_i}\mathbf{s}_i + \sum_{j \neq i} \frac{H_{ij}}{W_i}\mathbf{s}_j. \tag{108}$$

The recovery error $\mathbf{e}_i = \hat{\mathbf{s}}_i - \mathbf{s}_i$ satisfies

$$\mathbf{e}_i = \left(\frac{H_{ii}}{W_i} - 1\right)\mathbf{s}_i + \sum_{j \neq i} \frac{H_{ij}}{W_i}\mathbf{s}_j, \tag{109}$$

or equivalently,

$$\mathbf{e}_i = \sum_{j \neq i} \frac{H_{ij}}{W_i}(\mathbf{s}_j - \mathbf{s}_i) - \frac{h_i}{W_i}\mathbf{s}_i. \tag{110}$$

*Proof.* Substituting the forward model into the estimator gives

$$\begin{aligned} \hat{\mathbf{s}}_i &= \frac{1}{W_i} \sum_{p \in \mathcal{W}_i} w_{i,p} \left( \sum_{j=1}^{N} w_{j,p}\mathbf{s}_j \right) \\ &= \frac{1}{W_i} \sum_{j=1}^{N} H_{ij}\mathbf{s}_j. \end{aligned} \tag{111}$$

Separating the $j = i$ term proves Equation 108; subtracting $\mathbf{s}_i$ gives Equation 109. Furthermore,

$$W_i - H_{ii} = \sum_{p \in \mathcal{W}_i} w_{i,p}(1 - w_{i,p}). \tag{112}$$

By the partition-of-unity identity,

$$1 - w_{i,p} = \sum_{j \neq i} w_{j,p} + T_{\mathrm{bg},p}. \tag{113}$$

Thus

$$W_i - H_{ii} = \sum_{j \neq i} H_{ij} + h_i. \tag{114}$$

Substitution into Equation 109 proves Equation 110.

The error therefore cannot generally be attributed only to off-diagonal overlap. The first term depends on differences between overlapping semantic coefficients, while the second captures attenuation remaining after background subtraction. When the retained observations are fully opaque and all contributing Gaussians have the same semantic coefficient, both terms vanish even if their footprints overlap.

These are deterministic model-error identities, rather than statistical unbiasedness claims. If the observed targets contain an additional residual $\boldsymbol{\varepsilon}_p$, the error also contains $\sum_{p \in \mathcal{W}_i} w_{i,p}\boldsymbol{\varepsilon}_p / W_i$. Consequently, the identities do not guarantee recovery from noisy or systematically incorrect segmentation observations.

**Corollary 10 (Transparency-normalized recovery).** Under the assumptions of Theorem 9, define

$$B_i = W_i - h_i = \sum_{p \in \mathcal{W}_i} w_{i,p}(1 - T_{\mathrm{bg},p}) = \sum_j H_{ij}. \tag{115}$$

For $B_i > 0$, define the corrected estimator

$$\hat{\mathbf{s}}_i^{\mathrm{corr}} = \frac{W_i}{B_i}\hat{\mathbf{s}}_i = \frac{\sum_{p \in \mathcal{W}_i} w_{i,p}\mathbf{y}_p}{\sum_{p \in \mathcal{W}_i} w_{i,p}(1 - T_{\mathrm{bg},p})}, \tag{116}$$

where $\mathbf{y}_p = \sum_j w_{j,p}\mathbf{s}_j$ is the background-subtracted observation in the assumed forward model. Then

$$\hat{\mathbf{s}}_i^{\mathrm{corr}} - \mathbf{s}_i = \sum_{j \neq i} \frac{H_{ij}}{B_i}(\mathbf{s}_j - \mathbf{s}_i). \tag{117}$$

Consequently, if $\mathbf{s}_j = \mathbf{s}_i$ for every $j$ with $H_{ij} > 0$, then $\hat{\mathbf{s}}_i^{\mathrm{corr}} = \mathbf{s}_i$. If $h_i = 0$, the corrected and original estimators coincide.

*Proof.* The partition-of-unity identity gives

$$\sum_j H_{ij} = \sum_{p \in \mathcal{W}_i} w_{i,p} \sum_j w_{j,p} = \sum_{p \in \mathcal{W}_i} w_{i,p}(1 - T_{\mathrm{bg},p}) = B_i. \tag{118}$$

Therefore,

$$\hat{\mathbf{s}}_i^{\mathrm{corr}} = \sum_j \frac{H_{ij}}{B_i}\mathbf{s}_j, \qquad \sum_j \frac{H_{ij}}{B_i} = 1. \tag{119}$$

Subtracting $\mathbf{s}_i$ proves Equation 117. Under local semantic homogeneity, every difference vanishes.

The normalization removes transparency attenuation under the assumed model, but does not remove mixing between different semantic coefficients. If observations additionally contain residuals $\boldsymbol{\varepsilon}_p$, the corrected error includes

$$\frac{1}{B_i} \sum_{p \in \mathcal{W}_i} w_{i,p}\boldsymbol{\varepsilon}_p. \tag{120}$$

Relative to the original estimator, this residual contribution is amplified by $W_i/B_i$. Recovery therefore remains sensitive to observation errors when accumulated opacity is small.

**Corollary 11 (Bounds on the recovery error).** The decomposition also yields bounds that relate recovery error to mixing strength and semantic differences. These bounds explain which properties of the observations can make weighted recovery more accurate without assuming that all overlap disappears.

*Statement.* Under Theorem 9,

$$\|\mathbf{e}_i\|_2 \le \left|\frac{H_{ii}}{W_i} - 1\right| \|\mathbf{s}_i\|_2 + \sum_{j\neq i} \frac{H_{ij}}{W_i}\|\mathbf{s}_j\|_2. \tag{121}$$

An alternative bound is

$$\|\mathbf{e}_i\|_2 \le \sum_{j\neq i} \frac{H_{ij}}{W_i}\|\mathbf{s}_j - \mathbf{s}_i\|_2 + \frac{h_i}{W_i}\|\mathbf{s}_i\|_2. \tag{122}$$

*Proof.* Apply the triangle inequality to Equations 109 and 110, respectively. All $H_{ij}$ and $h_i$ are nonnegative because they are sums of products of nonnegative compositing quantities.

The second bound distinguishes overlap within semantically consistent regions from overlap between different semantic coefficients. It also shows why a general guarantee requires control of residual transparency. These are bounds on continuous color recovery, not guarantees that subsequent palette quantization selects the correct object.

### M.6 Relationship to the Global Inverse

**Lemma 12 (Normal equations of global least squares).** A global inverse fits all Gaussian coefficients jointly to the rendered observations. Comparing its normal equations with SCD makes explicit which interactions are omitted by independent weighted accumulation.

*Statement.* Using the stacked quantities from Theorem 3, any minimizer of

$$\min_{\mathbf{S}\in\mathbb{R}^{N\times 3}} \|\mathbf{W}\mathbf{S} - \mathbf{Y}\|_F^2 \tag{123}$$

satisfies

$$\mathbf{G}\mathbf{S} = \mathbf{B}, \qquad \mathbf{G} = \mathbf{W}^\top\mathbf{W}, \qquad \mathbf{B} = \mathbf{W}^\top\mathbf{Y}. \tag{124}$$

In particular,

$$G_{ij} = \sum_{p\in\mathcal{P}} w_{i,p}w_{j,p}, \qquad \mathbf{B}_{i,:}^\top = \sum_{p\in\mathcal{P}} w_{i,p}\mathbf{y}_p. \tag{125}$$

*Proof.* Differentiating the squared Frobenius norm gives

$$\nabla_{\mathbf{S}}\|\mathbf{W}\mathbf{S} - \mathbf{Y}\|_F^2 = 2\mathbf{W}^\top(\mathbf{W}\mathbf{S} - \mathbf{Y}). \tag{126}$$

Setting the gradient to zero gives Equation 124. The entrywise expressions follow from matrix multiplication. Since the objective is convex, solutions of these normal equations are global minimizers.

A unique unconstrained solution requires $\mathbf{W}$ to have full column rank. Merely observing each Gaussian does not guarantee that condition. In addition, the window-specific coefficients $H_{ij}$ in Theorem 9 need not be symmetric, because $\mathcal{W}_i$ can differ from $\mathcal{W}_j$; they equal the Gram entries $G_{ij}$ when all accumulation sets are the same full observation set.

**Theorem 13 (Distinction from a diagonal approximation).** Dropping off-diagonal terms from the global normal equations does not, by itself, produce the SCD estimator. The denominators differ, and identifying that difference prevents an incorrect interpretation of SCD as the exact diagonal solution.

*Statement.* For this comparison, let every Gaussian use the same observation set $\mathcal{P}$. Replacing $\mathbf{G}$ by $\mathrm{diag}(G_{11}, \ldots, G_{NN})$ gives

$$\widetilde{\mathbf{s}}_i = \frac{\sum_{p\in\mathcal{P}} w_{i,p}\mathbf{y}_p}{\sum_{p\in\mathcal{P}} w_{i,p}^2}, \qquad G_{ii} > 0. \tag{127}$$

Writing $W_i = \sum_{p\in\mathcal{P}} w_{i,p}$, the weighted average instead satisfies

$$\hat{\mathbf{s}}_i = \frac{G_{ii}}{W_i}\widetilde{\mathbf{s}}_i. \tag{128}$$

If every $w_{i,p} \in \{0, 1\}$, then $G_{ii} = W_i$ and the two estimates coincide.

*Proof.* The $i$th global normal equation is

$$G_{ii}\mathbf{s}_i + \sum_{j\neq i} G_{ij}\mathbf{s}_j = \mathbf{B}_{i,:}^{\top}. \tag{129}$$

Removing the off-diagonal terms gives $G_{ii}\widetilde{\mathbf{s}}_i = \mathbf{B}_{i,:}^{\top}$, proving Equation 127. Both estimates have the same numerator, which proves Equation 128. For binary weights, $w_{i,p}^2 = w_{i,p}$ term by term.

More precisely, define

$$\eta_i = \frac{W_i - G_{ii}}{W_i} = \frac{\sum_{p\in\mathcal{P}} w_{i,p}(1 - w_{i,p})}{W_i}. \tag{130}$$

Then

$$\|\widetilde{\mathbf{s}}_i - \hat{\mathbf{s}}_i\|_2 = \eta_i\|\widetilde{\mathbf{s}}_i\|_2. \tag{131}$$

Thus closeness requires a small aggregate discrepancy $\eta_i$; it does not follow merely because individual weights are small. Neither estimator generally solves the full coupled inverse when off-diagonal interactions matter.

### M.7 Multi-Frame Interpretation and Practical Scope

**Theorem 14 (Multi-frame weighted estimation).** SCD aggregates evidence from multiple camera frames using the same objective as the single-index formulation. The multi-frame extension changes the observation set, while preserving the estimator and its streaming implementation.

*Statement.* Let $\mathcal{F}$ index frames and let $\mathcal{P}_{i,f}$ contain the pixels retained by Gaussian $i$'s EWR window in frame $f$. Define

$$\mathcal{W}_i = \{(p, f) : f \in \mathcal{F},\ p \in \mathcal{P}_{i,f}\}. \tag{132}$$

If

$$W_i = \sum_{f\in\mathcal{F}} \sum_{p\in\mathcal{P}_{i,f}} w_{i,p,f} > 0, \tag{133}$$

the unique minimizer of

$$\sum_{f\in\mathcal{F}} \sum_{p\in\mathcal{P}_{i,f}} w_{i,p,f}\|\mathbf{s} - \mathbf{y}_{p,f}\|_2^2 \tag{134}$$

is

$$\hat{\mathbf{s}}_i = \frac{\sum_{f\in\mathcal{F}} \sum_{p\in\mathcal{P}_{i,f}} w_{i,p,f}\mathbf{y}_{p,f}}{\sum_{f\in\mathcal{F}} \sum_{p\in\mathcal{P}_{i,f}} w_{i,p,f}}. \tag{135}$$

*Proof.* Reindex each retained pixel–frame pair as a single observation. The objective is then exactly Equation 98, so Theorem 6 gives Equation 135. Equivalently, frame-wise accumulators

$$\mathbf{a}_{i,f} = \sum_{p\in\mathcal{P}_{i,f}} w_{i,p,f}\mathbf{y}_{p,f}, \qquad u_{i,f} = \sum_{p\in\mathcal{P}_{i,f}} w_{i,p,f} \tag{136}$$

yield

$$\hat{\mathbf{s}}_i = \frac{\sum_f \mathbf{a}_{i,f}}{\sum_f u_{i,f}}. \tag{137}$$

For the ideal forward model, stacking observations over all frames also gives $\mathbf{Y} = \mathbf{WS}$ with $\mathbf{W}_{(p,f),i} = w_{i,p,f}$, as in Theorem 3.

Additional views can provide complementary semantic evidence, but their inclusion alone does not guarantee identifiability or eliminate systematic errors. Repeated incorrect observations can accumulate across frames, and the normalized estimate does not directly measure whether those observations are reliable. This motivates treating recovery and confidence assessment as separate components.

**Implications for MonoVoc.** These results establish that SCD is an exact minimizer of a windowed, per-Gaussian weighted fitting objective and an approximation to semantic inversion under general compositing. They justify its accumulation-based implementation while identifying the limitations caused by overlap, transparency, and observation errors. EWR controls which observations are evaluated, but these proofs do not establish that omitted Gaussian tails have zero effect; the footprint approximation is discussed in Appendix D. Confidence B-Gated Quantization subsequently combines the recovered evidence with palette and spatial information, without a general guarantee of correcting every recovery error.