# CADPALETTE: GEOMETRIC PRIMITIVE SEGMENTATION OF CAD MODELS VIA PALETTE-CODED STRUCTURED 3D LATENTS

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



Figure 1: CADPalette segmentation results across sample shapes. Left: geometric primitive semantic segmentation. Right: instance segmentation. Our method enables these two types of segmentation in a holistic framework.

## ABSTRACT

Geometric primitive segmentation of CAD surfaces assigns each face a geometric type while preserving the boundaries of individual surface components. This task is difficult when discretization varies widely, because evidence learned from one mesh domain can change substantially across native CAD tessellations, regularized remeshes, and scan reconstructions, where altered sampling, resolution, and noise can destabilize boundaries. We introduce CADPalette, an instance-aware geometric primitive segmentation framework that carries primitive types and component membership along one structured 3D generative trajectory. Each surface is converted to Shape O-Voxel and Shape SLAT conditions, while a Palette O-Voxel stores fixed primitive bits and within-shape randomized instance colors in a shared carrier. A fine-tuned flow predicts the corresponding Palette SLAT, and one frozen material SC-VAE decodes both fields for face-level primitive-type and instance readout. Explicit RGB separation, a frozen SC-VAE response-interpolator gate, and latent relation supervision constrain the instance field from target construction through decoding. In the common validation protocol, CADPalette surpasses the baselines by 13.3% points in primitive-type Prim-mIoU and 14.9% points in primitive-type Prim-wIoU, while also surpassing the baselines by 21.5% points in Inst-mIoU, 7.3% points in ARI, and 5.3% points in Pair-F1. Extensive experiments show that CADPalette achieves stronger geometric primitive segmentation performance and better generalization than the existing methods.

## 1 INTRODUCTION

CAD reverse engineering seeks to recover an editable parametric model after the original design intent has been lost (Wang et al., 2024). The input is usually a raw surface mesh from a scan or a tessellated solid. We frame this problem as *geometric primitive segmentation*: each surface face is



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.

assigned a geometric type, such as plane, sphere, cylinder, cone, torus, or other. These primitive categories provide the semantic labels needed for primitive fitting and model reconstruction. An instance partition is a complementary structural output: it groups faces belonging to the same design component, including adjacent components that share the same primitive type. We take geometric primitive segmentation as the main task and report instance segmentation as an additional output of the same surface representation.

The problem is difficult because sampling and scale affect both geometry and neighborhoods. Point methods use sampled metric neighborhoods, serialized models relax them for scalability, and mesh methods retain connectivity at a cost that scales with edges or faces (Wang et al., 2019; Li et al., 2023; Wang, 2023; Wu et al., 2024; Hanocka et al., 2019; Milano et al., 2020). The same surface can therefore yield different evidence after native tessellation, remeshing, or scanning. Existing primitive-segmentation systems commonly use task-specific predictions or post-processing, while instance grouping relies on embeddings, boundaries, region growing, or fitting (Huang et al., 2021; Li et al., 2023; Fang et al., 2025; Li et al., 2019; Sharma et al., 2020; Wang et al., 2025). A useful representation should preserve primitive type and instance membership together while remaining stable across these changes in discretization.

The two outputs describe different properties, yet they can be carried by the same co-registered surface field. Fixed primitive codes preserve the meaning of each geometric category across shapes, whereas randomized RGB colors encode only within-shape instance equivalence. This separation lets one surface field support primitive-type prediction and instance grouping without assigning a global meaning to instance IDs. On the input side, native tessellations, remeshes, scan reconstructions, and other meshes are mapped to a common Shape O-Voxel and compressed into the Shape SLAT space of TRELLIS.2 (Xiang et al., 2026). The flow predictor then operates on sparse condition tokens rather than raw vertices or faces, while mesh geometry remains available for O-Voxelization and face-level readout.

We therefore introduce **CADPalette**, a geometric primitive segmentation framework with an instance-aware structured output. The six-channel carrier of a frozen material Sparse Compression VAE (SC-VAE) becomes a *Palette O-Voxel*: its first three channels store online-randomized instance RGB and its last three store fixed primitive bits. A fine-tuned TRELLIS.2 material flow predicts the corresponding *Palette SLAT* from Shape and Neutral Material SLAT conditions, and one frozen material decoder reconstructs both fields for face-level readout. Explicit RGB separation (E) and a frozen SC-VAE response-interpolator gate (I) screen target colors on VAE-grid contacts, while latent relation supervision (L) regularizes same- and cross-instance neighborhoods in the predicted clean latent. The decoded primitive bits give the primary geometric primitive segmentation, and the decoded RGB field gives an instance partition through deterministic face-graph readout (Fig. 1).

Our contributions are:

- **A unified palette-coded framework for geometric primitive segmentation.** We repurpose a frozen material structured-latent carrier to represent fixed primitive categories and instance membership in one six-channel field. A single fine-tuned flow predicts the Palette SLAT from Shape and Neutral Material conditions, and a frozen SC-VAE decodes face-level primitive labels together with an instance-aware partition under the same inference pipeline.

- **A dual-space randomized palette target.** We combine fixed signed primitive bits with online-randomized instance RGB, making the instance code invariant to permutations of local instance IDs. Explicit RGB separation and a codec-aware frozen SC-VAE response gate filter target colors in complementary spaces, reducing color collisions at VAE-grid contacts while preserving the primitive codebook.

- **Latent relation supervision with deterministic instance readout.** We regularize same- and cross-instance relations in the predicted clean Palette SLAT and recover instances by thresholding decoded RGB on the mesh face-adjacency graph. This connects latent prediction to the final instance partition while keeping geometric primitive segmentation and instance evaluation separately measurable.

## 2 Related Work

**Geometric primitive segmentation.** Geometric primitive segmentation assigns each point or surface face a type, such as a plane or cylinder. Classical approaches fit primitives with hand-designed proposals and iterative optimization, as in RANSAC, Efficient RANSAC, and GlobFit (Fischler & Bolles, 1981; Schnabel et al., 2007; Li et al., 2011). Learned methods such as SPFN, ParSeNet, HPNet, CPFN, BPNet, and probabilistic boundary-guided networks predict primitive, fitting, or boundary cues directly from 3D observations (Li et al., 2019; Sharma et al., 2020; Yan et al., 2021; Lê et al., 2021; Fu et al., 2023; Wang et al., 2023). PrimitiveNet, SED-Net, BGPSeg, and TCIPS also use boundaries or instance cues (Huang et al., 2021; Li et al., 2023; Fang et al., 2025; Wang et al., 2025), but primitive-type prediction and component grouping generally follow separate output or post-processing paths. CADPalette keeps fixed primitive codes and randomized within-shape instance colors in one surface carrier: primitive bits provide the main geometric primitive segmentation, and the color field provides an instance partition.

**Discretization-aware representations and generative priors.** Mesh-native operators use connectivity, including edge, primal-dual, walk, and diffusion features in MeshCNN, PD-MeshNet, MeshWalker, and DiffusionNet (Hanocka et al., 2019; Milano et al., 2020; Lahav & Tal, 2020; Sharp et al., 2022). Point models use sampled or serialized neighborhoods to improve scalability, as in OctFormer, PTv3, Point Transformer, DGCNN, and SED-Net (Wang, 2023; Wu et al., 2024; Zhao et al., 2021; Wang et al., 2019; Li et al., 2023). These choices change the local evidence available after B-rep tessellation, remeshing, or scanning. TRELLIS.2 provides a structured alternative with O-Voxel representations, shape and material VAEs, SLATs, and a sparse flow prior (Xiang et al., 2026; Wang et al., 2026b;a). CADPalette uses this interface for CAD surface parsing and changes the supervised material carrier: fixed primitive bits and randomized instance RGB are packed into six channels, which a fine-tuned material flow predicts in the frozen material-latent space through conditional flow matching (Lipman et al., 2023).

**General 3D part segmentation.** General 3D systems learn object-part masks or feature fields, including SAMesh, GeoSAM2, SAMPart3D, PartField, Find3D, P3-SAM, and PartSAM (Tang et al., 2024; Deng et al., 2026; Yang et al., 2024; Liu et al., 2025; Ma et al., 2025b;a; Zhu et al., 2026). These methods provide context for instance-aware surface grouping, but their part labels describe object structure rather than CAD geometric types. SegViGen and UniPart use generative models for 3D part decomposition; UniPart further encodes part structure with Geom-Seg VecSet and latent diffusion (Li et al., 2026; He et al., 2026). CADPalette shares the use of a 3D generative prior while making geometric primitive categories the primary labels and deriving instance partitions from the same decoded field.

## 3 Method

CADPalette targets geometric primitive segmentation and obtains an instance partition from the same structured surface representation. Fixed binary codes assign each face a primitive type with the same meaning across shapes, while randomized colors record equivalence only within the current shape. Both fields share one Palette SLAT and one frozen material SC-VAE, so primitive labels and instance membership stay co-registered from prediction to face-level readout (Fig. 2).

### 3.1 Palette-Coded Primitive and Instance Carrier

Let a CAD surface contain elements $\{s_i\}_{i=1}^{N}$. Each element has a fixed primitive code $\mathbf{q}_i \in \{0,1\}^3$ and a within-shape instance label $y_i^{\text{ins}} \in \{1,\dots,K\}$. A fixed lookup $\pi : \{0,1\}^3 \to \mathcal{Y}_{\text{prim}}$ maps the eight codewords to the six public primitive categories. The same lookup is applied to reference and predicted codes.

At training micro-step $m$, each instance $k$ receives a newly sampled color $\mathbf{c}_k^{(m)} \in [0,255]^3$. The color has no identity across shapes or training steps. With $\bar{\mathbf{c}} = \mathbf{c}/255$, the normalized target for element $s_i$ is

$$\mathcal{T}_i^{(m)} = \left[2\bar{\mathbf{c}}_{y_i^{\text{ins}}}^{(m)} - 1,\ 2\mathbf{q}_i - 1\right]. \tag{1}$$

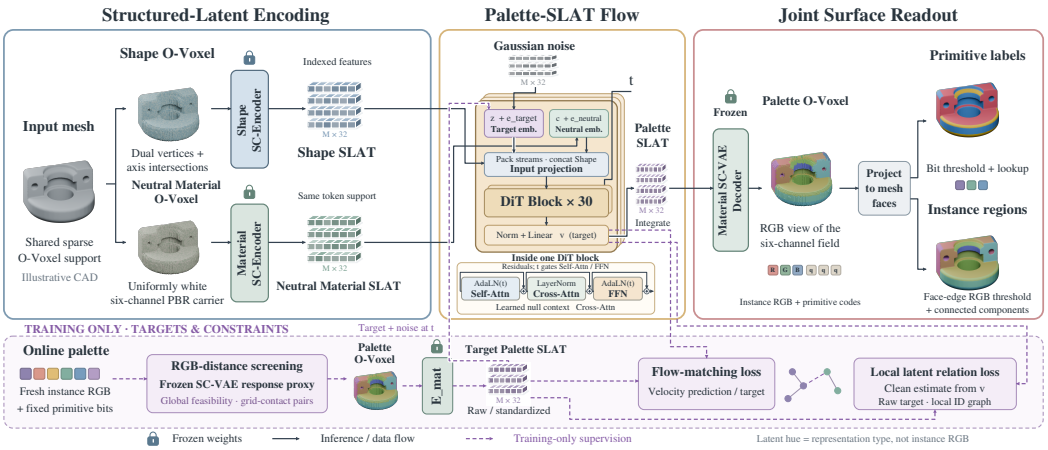



Figure 2: **Overview of CADPalette.** Frozen SC-encoders map aligned Shape and Neutral Material O-Voxels to conditions for a DiT flow that generates one Palette SLAT. The frozen material decoder reconstructs instance-RGB and primitive-code fields for face-level readout. Primitive codes give the main geometric primitive segmentation, while the instance field gives an additional partition. During training, online colors are filtered by explicit RGB separation (E) and a frozen SC-VAE response-interpolator gate (I), while latent relation supervision (L) regularizes the predicted clean latent.

Rasterizing these attributes on the aligned active O-Voxel support gives the six-channel Palette O-Voxel $\mathcal{T}^{(m)}$. Primitive codes retain their fixed meaning across shapes; colors are reassigned online, so the target distribution is invariant to a permutation of local instance IDs.

The frozen material SC-VAE encoder maps $\mathcal{T}^{(m)}$ to a Palette SLAT target, and its decoder later reconstructs the primitive and instance fields. The primitive bits and instance RGB are separated only at readout. The codec keeps the existing O-Voxelization and frozen SC-VAE architecture; only the supervised carrier semantics are changed.

### 3.2 Dual-Space Randomized Instance Palette

The instance field is continuous until the final connected-component readout. Adjacent instances can merge when their target colors are too close, when the frozen codec maps them to similar responses, or when the learned field forms a smooth ramp across a boundary. CADPalette addresses the first two cases during target construction with two independently switchable constraints: *explicit RGB separation* (E) and the *frozen SC-VAE response gate* (I). Both constraints act only on the recorded or deterministically inferred instance-contact graph for the VAE grid.

For a queried color $\mathbf{c}$, a frozen-VAE response interpolator $\widehat{\mathbf{r}}(\mathbf{c})$ predicts the encoder response from a fixed RGB–embedding calibration asset. RGB inputs are scaled by $1/255$; the default interpolator is a global thin-plate radial-basis interpolator with smoothing $10^{-6}$. Its response dimension is inherited from the stored embedding array rather than fixed by the method. We define

$$\widehat{D}_{\mathrm{VAE}}(\mathbf{c}_k, \mathbf{c}_\ell) = \left\| \widehat{\mathbf{r}}(\mathbf{c}_k) - \widehat{\mathbf{r}}(\mathbf{c}_\ell) \right\|_2 . \tag{2}$$

This is a global codec-response proxy; it is not a local boundary-separability measurement.

For each micro-step, the sampler draws fresh continuous-valued RGB colors from $[0, 255]^3$ and assigns them in a deterministic graph order. For every edge $(k, \ell)$ in the VAE-grid contact graph $\mathcal{E}_{\mathrm{grid}}$, E requires $\|\mathbf{c}_k - \mathbf{c}_\ell\|_2 \geq \tau_{\mathrm{RGB}}$ when enabled, and I requires

$$\widehat{D}_{\mathrm{VAE}}(\mathbf{c}_k, \mathbf{c}_\ell) - \delta_{\mathrm{VAE}} \geq \tau_{\mathrm{VAE}} \tag{3}$$

when enabled. The implementation uses $\tau_{\mathrm{RGB}} = 130$ in 8-bit RGB units, $\tau_{\mathrm{VAE}} = 4.0$, and $\delta_{\mathrm{VAE}} = 0.20$. The latter is a one-sided empirical pair-distance margin, not a uniform error bound. No fixed color pool, reusable candidate pool, global uniqueness constraint, or threshold relaxation is used. A proposal budget of 512 attempts per instance is enforced; if no legal assignment is found, sampling fails closed rather than silently changing the target contract.

### 3.3 SHARED STRUCTURED-LATENT PREDICTION

Each input mesh is rasterized onto a common sparse O-Voxel support. The Shape O-Voxel stores local dual-vertex coordinates and three axis-wise intersection indicators; a frozen shape SC-encoder maps it to a Shape SLAT. A uniformly white Neutral Material O-Voxel is mapped by the frozen material SC-encoder to a Neutral Material SLAT used as an auxiliary condition. At training time, the Palette O-Voxel from Eq. (1) is encoded by the same material encoder to form the unknown target Palette SLAT.

At inference, Shape and Neutral Material SLATs are required as inputs. After the pretrained latent statistics are applied, they form the condition $\mathcal{C}$. The flow model predicts sparse Palette-SLAT tokens rather than directly predicting mesh-face types as in previous methods.

The flow is initialized from the pretrained TRELLIS.2 material-SLAT flow and fine-tuned to predict the Palette SLAT target. For $t \sim \mathcal{U}(0,1)$, $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$, and $\sigma_{\min} = 10^{-5}$, define

$$\begin{aligned} \mathbf{z}_t &= (1-t)\mathbf{z}_0 + [\sigma_{\min} + (1-\sigma_{\min})t]\boldsymbol{\epsilon}, \\ \mathbf{u} &= (1-\sigma_{\min})\boldsymbol{\epsilon} - \mathbf{z}_0. \end{aligned} \tag{4}$$

The conditional velocity field is trained with the per-shape normalized loss

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{t,\boldsymbol{\epsilon}} \left[ \frac{1}{B} \sum_{n=1}^{B} \frac{1}{M_n C} \sum_{i=1}^{M_n} \left\| \mathbf{v}_\theta(\mathbf{z}_t^{(n)}, t, \mathcal{C}^{(n)})_i - \mathbf{u}_i^{(n)} \right\|_2^2 \right], \tag{5}$$

where $M_n$ is the active-token count and $C$ is the latent channel count. The within-shape normalization prevents token-rich examples from dominating.

### 3.4 LATENT RELATION SUPERVISION

Flow matching supplies tokenwise supervision but does not explicitly preserve the local instance relations needed by the final RGB partition. Component L therefore operates on the predicted clean latent. From the velocity prediction, estimate

$$\widehat{\mathbf{z}_0} = (1-\sigma_{\min})\mathbf{z}_t - [\sigma_{\min} + (1-\sigma_{\min})t]\mathbf{v}_\theta. \tag{6}$$

Instance IDs are aggregated to latent tokens by majority vote. The confidence of a token is the fraction of its contributing VAE-grid elements assigned to that modal instance ID. Edges are retained only when both endpoint confidences meet the fixed threshold. For an edge $(i, j)$ in this relation graph, both latents use the flow-training coordinate system; pairwise distances are normalized by $\sqrt{C}$:

$$d_{ij} = \frac{\|\mathbf{z}_{0i} - \mathbf{z}_{0j}\|_2}{\sqrt{C}}, \qquad \widehat{d}_{ij} = \frac{\|\widehat{\mathbf{z}}_{0i} - \widehat{\mathbf{z}}_{0j}\|_2}{\sqrt{C}}. \tag{7}$$

Same-instance edges use a Huber regression term. Cross-instance edges use an adaptive hinge margin derived from the detached target distance:

$$\begin{aligned} \ell_{\mathrm{same}}(i,j) &= \mathrm{Huber}(\widehat{d}_{ij} - d_{ij}), \\ \ell_{\mathrm{cross}}(i,j) &= \left[\max(0, m_{ij} - \widehat{d}_{ij})\right]^2, \\ m_{ij} &= \mathrm{clip}(\rho d_{ij}, m_{\min}, m_{\max}). \end{aligned} \tag{8}$$

The implementation uses $\rho = 0.8$, $m_{\min} = 0.2$, $m_{\max} = 2.0$, and confidence threshold $0.85$. Each relation type is averaged within a shape; an absent relation type contributes zero, and shapes without reliable edges are excluded from the relation average. With the same- and cross-instance terms combined as $\mathcal{L}_{\mathrm{rel}}$, the complete training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{FM}} + \lambda_{\mathrm{rel}} \mathcal{L}_{\mathrm{rel}}, \tag{9}$$

with default $\lambda_{\mathrm{rel}} = 0.05$.

L is a latent-space regularizer. It does not backpropagate through the frozen decoder and it does not directly impose a face-level RGB loss. E and I shape the online target colors, whereas L constrains how the flow reproduces local relations in the predicted latent. These interventions are complementary.

### 3.5 Primitive Segmentation and Instance Readout

At inference, the Shape and Neutral Material SLATs condition a flow trajectory from Gaussian noise to one Palette SLAT. The frozen material decoder reconstructs the sparse six-channel Palette O-Voxel, and nearest-center assignment transfers its values to the prepared mesh faces.

For geometric primitive segmentation, the last three channels encode the primitive category. Thresholding these values at zero recovers a binary code, which the fixed mapping $\pi$ converts to a primitive label for each face. Optional spatial smoothing reduces isolated label noise: small connected regions adopt the dominant label along their boundary, and faces with weak support for their current label are reassigned when neighboring faces strongly agree on another category.

For instance segmentation, the first three channels provide a continuous RGB field on the mesh. We connect adjacent faces whose RGB Euclidean distance is at most 12 in the 8-bit color space and treat each connected component as an instance. An optional refinement instead groups faces by region growing, using local and region-level color similarity together with face-normal changes to discourage merging across sharp edges. Small fragments, including regions with blended colors, are absorbed into compatible neighboring regions according to color agreement and shared-boundary support. When primitive predictions are available, substantial connected regions of different types can further split an instance, while isolated label errors do not form new instances. The quantitative evaluations use the direct primitive labels and fixed-threshold instance readout before these optional refinements.

## 4 Experiments

### 4.1 Experimental Protocol

**Implementation details.** We construct a corpus of 274k CAD samples from a deduplicated and curated subset of ABC (Koch et al., 2019), a curated subset of Fusion 360 Gallery models (Lambourne et al., 2021), and additional CAD models collected from the public web. Every retained asset carries fixed three-bit primitive codes and surface-instance annotations. We use OpenCASCADE discretizations, approximately 100k-face remeshes, and controlled boundary/noise perturbations to cover native CAD tessellations, regularized remeshes, and scan-like inputs. For a subset of models, we produced physical prints, rescanned them, and reconstructed the scans as meshes to serve as test samples for the scanned-model evaluation. The complete baseline comparison protocol uses 273k samples for training and a fixed, disjoint validation set of 1k samples. The mechanism studies use a fixed 9k subset of the 273k training samples and the same fixed 1k validation set.

CADPalette is initialized from the publicly released VAE and flow-model pretrained weights of TRELLIS.2; the material SC-VAE encoder and decoder remain frozen. We optimize with AdamW (learning rate $10^{-4}$, weight decay $0.01$, gradient-norm clip $1.0$), a constant learning-rate schedule, and global batch size 64. Flow-matching timesteps are sampled uniformly, and inference uses 12 Euler steps. The mechanism studies use a 10k-step budget and the same optimizer, initialization, and readout as the complete training protocol. All reported comparison and ablation entries are means over two random seeds.

### 4.2 Comparison with the State of the Art

We select eight recent strong baselines covering complementary point- and mesh-processing families. For every point baseline, we sample mesh-face centroids and map predictions back to their source faces; after subsampling, each remaining face takes the label of its nearest sampled centroid. Baselines without a native instance output receive an embedding head trained with the same instance annotations, and mean-shift clustering of the predicted embeddings produces their instance partitions at inference. Table 1 compares methods trained on the common 273k-sample training split under their selected configurations. Every method receives the same fixed 1k validation samples in the same uniformly remeshed triangular-mesh form. Evaluation reports primitive recognition and instance segmentation separately; the complete six-class breakdown and metric definitions are given in Supplementary Material. Figure 3 offers a qualitative comparison: CADPalette preserves small primitive regions and separates adjacent instances more coherently than the representative baselines.

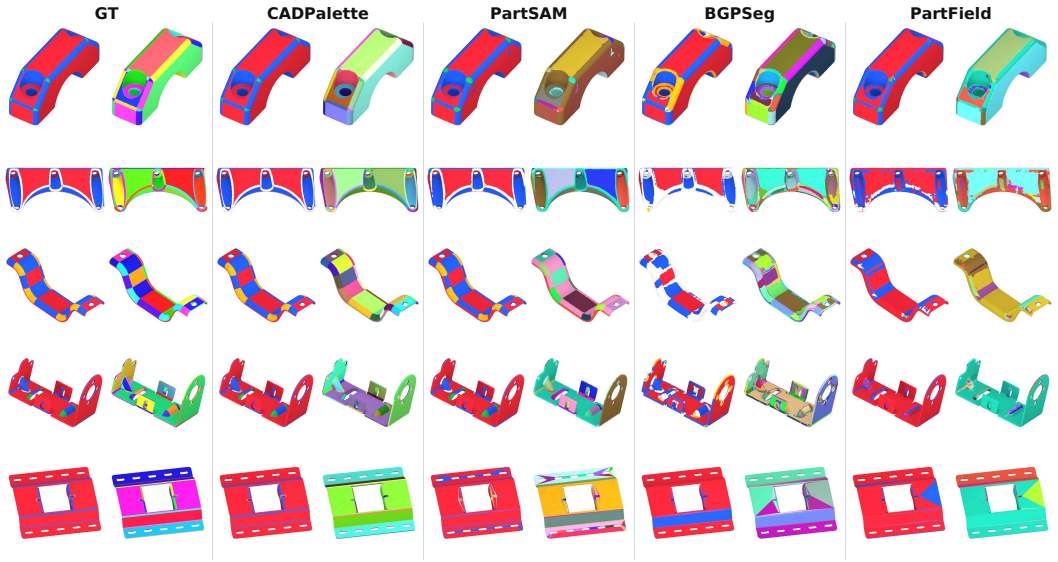
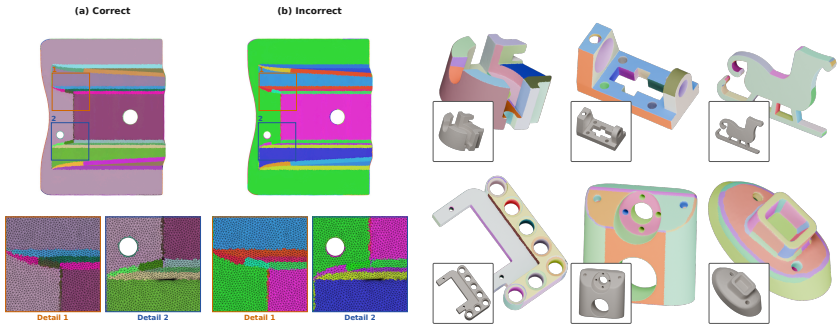



Figure 3: Qualitative comparison of ground-truth annotations and geometric primitive and instance segmentation results on challenging samples. Rows are five representative models; within each method the left image shows geometric primitive segmentation and the right shows instance segmentation. The fifth sample is a native tessellation mesh. Baseline methods include PartSAM (Zhu et al., 2026), BGPSeg (Fang et al., 2025), and PartField (Liu et al., 2025).

Table 1: Quantitative comparison with recent strong baselines for instance segmentation and geometric primitive segmentation. All methods are trained on 273k samples and evaluated on a fixed validation set of 1k samples. The best results are in bold.

| | Geometric primitive segmentation | | Instance segmentation | | |
|---|---|---|---|---|---|
| Method | Prim-mIoU ↑ | Prim-wIoU ↑ | Inst-mIoU ↑ | ARI ↑ | Pair-F1 ↑ |
| *Point-based methods* | | | | | |
| BGPSeg (Fang et al., 2025) | 0.538 | 0.729 | 0.271 | 0.284 | 0.490 |
| SED-Net (Li et al., 2023) | 0.381 | 0.533 | 0.349 | 0.520 | 0.614 |
| TCIPS (Wang et al., 2025) | 0.561 | 0.753 | 0.494 | 0.752 | 0.804 |
| P3-SAM (Ma et al., 2025a) | 0.378 | 0.589 | 0.391 | 0.562 | 0.650 |
| PartSAM (Zhu et al., 2026) | 0.712 | 0.796 | 0.530 | 0.766 | 0.820 |
| *Mesh-based methods* | | | | | |
| PartField (Liu et al., 2025) | 0.664 | 0.743 | 0.536 | 0.843 | 0.877 |
| PatchAlign3D (Hadgi et al., 2026) | 0.491 | 0.692 | 0.294 | 0.374 | 0.451 |
| DiffusionNet (Sharp et al., 2022) | 0.576 | 0.764 | 0.325 | 0.588 | 0.685 |
| **CADPalette** | **0.845** | **0.945** | **0.751** | **0.916** | **0.930** |

Table 1 shows that CADPalette performs strongly on both geometric primitive recognition and instance segmentation, with a substantial lead over the strongest selected baselines. PartSAM leads the selected baselines in primitive recognition, whereas PartField performs best in instance segmentation; neither matches CADPalette on both measures. CADPalette improves both under the same protocol. This result supports representing primitive types and instance membership in a joint latent field, which keeps their spatial correspondence through shared prediction and decoding. Figure 3 shows the same pattern qualitatively: CADPalette preserves small primitive regions and delineates their boundaries, with particularly clear advantages in instance segmentation. Its instance partitions follow the ground-truth regions more faithfully and remain spatially coherent, giving cleaner separation between neighboring components. The quantitative and qualitative comparisons therefore support the use of a joint representation for primitive identification and surface partition recovery.




(a) (b)

Figure 4: Qualitative evidence for RGB-separation behavior and qualitative scan transfer. (a) Matched predictions with and without the E+I+L training components, showing the RGB-separation mechanism. (b) Six scan-reconstructed meshes with corresponding raw scan insets. The models were printed, scanned, and reconstructed as meshes.

Table 2: Factorial analysis of explicit RGB-space separation (E), the frozen SC-VAE global mean-response gate (I), and latent relation supervision (L). The first row keeps the complete configuration but uses one fixed feasible assignment per shape; the remaining eight rows use online resampling. Entries are means over two random seeds. GQ diagnoses color continuity only for flow-based CAD-Palette variants and is not a task-accuracy or cross-method ranking metric.

| Explicit RGB | SC-VAE gate | Latent relation | Assignment | Prim-wIoU ↑ | Inst-mIoU ↑ | ARI ↑ | GQ ↑ |
|---|---|---|---|---|---|---|---|
| ✓ | ✓ | ✓ | fixed | 0.869 | 0.712 | 0.907 | 0.901 |
| × | × | × | online | 0.850 | 0.560 | 0.742 | 0.730 |
| × | × | ✓ | online | 0.856 | 0.605 | 0.805 | 0.780 |
| × | ✓ | × | online | 0.854 | 0.590 | 0.785 | 0.765 |
| × | ✓ | ✓ | online | 0.862 | 0.650 | 0.850 | 0.830 |
| ✓ | × | × | online | 0.858 | 0.620 | 0.820 | 0.790 |
| ✓ | × | ✓ | online | 0.865 | 0.680 | 0.875 | 0.860 |
| ✓ | ✓ | × | online | 0.867 | 0.700 | 0.895 | 0.890 |
| ✓ | ✓ | ✓ | **online** | **0.870** | **0.724** | **0.919** | **0.913** |

Additional qualitative results, including inference results on scanned models, are provided in Supplementary Material.

### 4.3 Palette Separation and Relation Analysis

The instance output is a continuous RGB field followed by threshold-12 connected components on mesh-face adjacency. A prominent failure mode is adjacent-instance merging: two neighboring ground-truth instances can merge when their decoded colors are too close or when a smooth color ramp crosses their boundary (Fig. 4). We therefore evaluate three training components: explicit RGB separation (E), the frozen SC-VAE response-interpolator gate (I), and latent relation supervision (L). E and I filter online target colors on VAE-grid contacts; L regularizes same- and cross-instance relations in the predicted clean latent. The eight online rows form the component comparison, and the additional fixed row isolates online resampling from the complete E+I+L target contract.

Table 2 indicates that E, I, and L primarily improve instance recovery, while primitive recognition remains largely stable. At VAE-grid contacts, E and I increase separation between neighboring instances in the target representation, while L regularizes their local relations in the predicted latent. The combined setting is consistent with fewer color-confusion and merging errors at instance contacts, as illustrated in Figure 4(a). Its higher GQ also provides diagnostic evidence that the decoded color fields are smoother. The fixed-assignment control already retains most of the full E+I+L instance gain, while online resampling adds a modest improvement by varying the color encoding of the same partition.

Table 3: One joint flow versus independently fine-tuned geometric-primitive and instance-segmentation flows. All entries are means over two random seeds.

| Design | Prim-wIoU ↑ | Inst-mIoU ↑ | ARI ↑ |
|---|---|---|---|
| Independent primitive-segmentation flow | 0.858 | – | – |
| Independent instance-segmentation flow | – | 0.688 | 0.872 |
| **CADPalette: one joint flow** | 0.870 | 0.724 | 0.919 |

### 4.4 One Joint Flow versus Two Independent Task Flows

We compare one joint flow with separately fine-tuned geometric-primitive and RGB-only instance flows. The comparison tests whether the two fields can be carried by one structured trajectory; it does not measure a universal parameter-sharing or resource advantage. Under this matched training protocol, the joint flow obtains higher instance-recovery scores while retaining comparable primitive-recognition accuracy. This pattern matches the intended role of the joint carrier: primitive cues describe geometric type, while instance relations distinguish separate surfaces of the same type. With this elaborate six-channel definition, the simultaneous generation of both fields along a single structured trajectory enables effective learning for both geometric primitive and instance segmentation tasks.

### 4.5 Qualitative Scan Transfer

To provide qualitative evidence of transfer beyond the CAD training distribution, we directly run CADPalette inference on scan-reconstructed meshes. Figure 4(b) shows six scan instances together with their original, unsegmented scan renderings. The scan surfaces contain substantial geometric noise, yet the predicted segmentation visibly produces coherent primitive regions and instance colors. This visual comparison supports qualitative instance partitioning under irregular scan-induced tessellation and geometric perturbations. We use the standard inference pipeline throughout, without scan-specific fine-tuning, calibration. Additional qualitative results are provided in Supplementary Section.

## 5 Discussion and Limitations

The main runtime cost currently comes from converting between O-Voxel and mesh representations. GPU memory requirements depend on the complexity of the geometric model. O-Voxel–mesh conversion, VAE reconstruction, and the choice of resolution each introduce approximation errors, which become particularly pronounced for complex models. These errors can further affect both geometric primitive prediction and instance-segmentation results.

## 6 Conclusion

We presented CADPalette for geometric primitive segmentation of CAD surfaces. Its palette-coded structured latent stores fixed primitive bits as the primary face labels and online-randomized instance RGB as an additional component partition. Explicit RGB separation (E) and the frozen SC-VAE response gate (I) select target colors that distinguish neighboring instances, while latent relation supervision (L) encourages the flow to preserve instance relationships. Together, these components reduce adjacent-instance merging and improve instance segmentation. On the labeled CAD validation protocol, the method leads the compared point and mesh baselines on both primitive-type and instance metrics. On unlabeled scan-reconstructed meshes, the same inference pipeline produces visually coherent primitive regions and instance partitions in qualitative inspection. Together, these results support structured latents as a discretization-decoupled representation for CAD surface parsing and scan-based reconstruction.

## AI USE STATEMENT

No generative AI tools were used to write the paper, create its figures, implement the code, or analyze the reported CADPalette results. As an explicitly evaluated training-free baseline in the supplementary material, GPT Image 2.5 was used to generate simulated instance-color images; it was not used to train CADPalette or to generate ground-truth annotations.

## ETHICS STATEMENT

This work studies geometric surface parsing and involves no human subjects, personal data, or sensitive information. The training data is built from public CAD datasets (Koch et al., 2019; Lambourne et al., 2021) and CAD models collected from public web sources, and the scanned evaluation samples were produced by the authors by printing and rescanning CAD models, so no third-party personal data is involved.

## REPRODUCIBILITY STATEMENT

The main paper and the supplementary material together specify the complete protocol needed to reproduce the results reported in this paper. The implementation code of CADPalette will be released as open source in the future.

## REFERENCES

Ken Deng, Yunhan Yang, Jingxiang Sun, Xihui Liu, Yebin Liu, Ding Liang, and Yan-Pei Cao. GeoSAM2: Unleashing the power of SAM2 for 3d part segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 6367–6376, 2026. 3

Zheng Fang, Chuanqing Zhuang, Zhengda Lu, Yiqun Wang, Lupeng Liu, and Jun Xiao. BGPSeg: Boundary-guided primitive instance segmentation of point clouds. *IEEE Trans. Image Process.*, 34:1454–1468, 2025. 2, 3, 7

Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. 3

Rao Fu, Cheng Wen, Qian Li, Xiao Xiao, and Pierre Alliez. BPNet: Bézier primitive segmentation on 3d point clouds. In *Int. Joint Conf. Artificial Intell.*, pp. 754–762, 2023. 3

Souhail Hadgi, Bingchen Gong, Ramana Sundararaman, Emery Pierson, Lei Li, Peter Wonka, and Maks Ovsjanikov. PatchAlign3D: Local feature alignment for dense 3d shape understanding. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 3014–3023, 2026. 7

Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. MeshCNN: A network with an edge. *ACM Trans. Graph.*, 38(4):1–12, 2019. 2, 3

Xufan He, Yushuang Wu, Xiaoyang Guo, Chongjie Ye, Jiaqing Zhou, Tianlei Hu, Xiaoguang Han, and Dong Du. UniPart: Part-level 3d generation with unified 3d geom-seg latents. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 34227–34236, 2026. 3

Jingwei Huang, Yanfeng Zhang, and Mingwei Sun. PrimitiveNet: Primitive instance segmentation with local primitive embedding under adversarial metric. In *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 15323–15333, 2021. 2, 3

Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. ABC: A big CAD model dataset for geometric deep learning. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 9601–9611, 2019. 6, 10

Alon Lahav and Ayellet Tal. MeshWalker: Deep mesh understanding by random walks. *ACM Trans. Graph.*, 39(6):1–13, 2020. 3

Joseph G. Lambourne, Karl D. D. Willis, Pradeep Kumar Jayaraman, Aditya Sanghi, Peter Meltzer, and Hooman Shayani. BRepNet: A topological message passing system for solid models. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 12773–12782, 2021. 6, 10

Eric-Tuan Lê, Minhyuk Sung, Duygu Ceylan, Radomír Mech, Tamy Boubekeur, and Niloy J. Mitra. CPFN: Cascaded primitive fitting networks for high-resolution point clouds. In *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 7438–7446, 2021. 3

Lin Li, Haoran Feng, Zehuan Huang, Haohua Chen, Wenbo Nie, Shaohua Hou, Keqing Fan, Pan Hu, Sheng Wang, Buyu Li, and Lu Sheng. SegviGen: Repurposing 3d generative model for part segmentation. *ACM Trans. Graph.*, 45(4):1–12, 2026. 3

Lingxiao Li, Minhyuk Sung, Anastasia Dubrovina, Li Yi, and Leonidas J. Guibas. Supervised fitting of geometric primitives to 3d point clouds. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 2647–2655, 2019. 2, 3

Yangyan Li, Xiaokun Wu, Yiorgos Chrysanthou, Andrei Sharf, Daniel Cohen-Or, and Niloy J. Mitra. GlobFit: Consistently fitting primitives by discovering global relations. *ACM Trans. Graph.*, 30 (4):52:1–52:12, 2011. 3

Yuanqi Li, Shun Liu, Xinran Yang, Jianwei Guo, Jie Guo, and Yanwen Guo. Surface and edge detection for primitive fitting of point clouds. In *ACM SIGGRAPH 2023 Conference Proceedings*, pp. 1–10, 2023. 2, 3, 7

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *Proc. Int. Conf. Learn. Rep.*, 2023. 3

Minghua Liu, Mikaela Angelina Uy, Donglai Xiang, Hao Su, Sanja Fidler, Nicholas Sharp, and Jun Gao. PartField: Learning 3d feature fields for part segmentation and beyond. In *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 9704–9715, 2025. 3, 7

Changfeng Ma, Yang Li, Xinhao Yan, Jiachen Xu, Yunhan Yang, Chunshi Wang, Zibo Zhao, Yanwen Guo, Zhuo Chen, and Chunchao Guo. P3-SAM: Native 3d part segmentation. *arXiv preprint arXiv:2509.06784*, 2025a. 3, 7

Ziqi Ma, Yisong Yue, and Georgia Gkioxari. Find any part in 3d. In *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 7818–7827, 2025b. 3

Francesco Milano, Antonio Loquercio, Antoni Rosinol, Davide Scaramuzza, and Luca Carlone. Primal-dual mesh convolutional neural networks. In *Proc. Int. Conf. Neural Inf. Process. Syst.*, volume 33, pp. 952–963, 2020. 2, 3

Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient RANSAC for point-cloud shape detection. *Comput. Graph. Forum*, 26(2):214–226, 2007. 3

Gopal Sharma, Difan Liu, Subhransu Maji, Evangelos Kalogerakis, Siddhartha Chaudhuri, and Radomír Měch. ParSeNet: A parametric surface fitting network for 3d point clouds. In *Proc. Eur. Conf. Comput. Vis.*, volume 12352, pp. 261–276, 2020. 2, 3

Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. DiffusionNet: Discretization agnostic learning on surfaces. *ACM Trans. Graph.*, 41(3):1–16, 2022. 3, 7

George Tang, William Zhao, Logan Ford, David Benhaim, and Paul Zhang. Segment any mesh. *arXiv preprint arXiv:2408.13679*, 2024. 3

Hanxiao Wang, Mingyang Zhao, Yiqun Wang, Weize Quan, and Dong-Ming Yan. VQ-CAD: Computer-aided design model generation with vector quantized diffusion. *Computer Aided Geometric Design*, 111:102327, 2024. 1

Hanxiao Wang, Yuan-Chen Guo, Ying-Tian Liu, Zi-Xin Zou, Biao Zhang, Weize Quan, Ding Liang, Yan-Pei Cao, and Dong-Ming Yan. FACE: A face-based autoregressive representation for high-fidelity and efficient mesh generation. *arXiv preprint arXiv:2603.01515*, 2026a. 3

Hanxiao Wang, Ying-Tian Liu, Yuan-Chen Guo, Qi-Yuan Feng, Zi-Xin Zou, Ding Liang, Biao Zhang, and Yan-Pei Cao. Nexus: Native mesh generation with diffusion. *ACM Trans. Graph.*, 45 (4):1–14, 2026b. 3

Peng-Shuai Wang. OctFormer: Octree-based transformers for 3d point clouds. *ACM Trans. Graph.*, 42(4):155:1–155:11, 2023. 2, 3

Shaohu Wang, Fangbo Qin, Yuchuang Tong, Xiuqin Shang, and Zhengtao Zhang. Probabilistic boundary-guided point cloud primitive segmentation network. *IEEE Trans. Instrum. Meas.*, 72: 1–13, 2023. 3

Tao Wang, Weibin Xi, Yong Cheng, Jun Zhang, Ruochen Yin, and Yang Yang. Enhancing primitive segmentation through transformer-based cross-task interaction. *Engineering Applications of Artificial Intelligence*, 158:111307, 2025. 2, 3, 7

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph CNN for learning on point clouds. *ACM Trans. Graph.*, 38(5):146:1–146:12, 2019. 2, 3

Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer V3: Simpler, faster, stronger. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 4840–4851, 2024. 2, 3

Jianfeng Xiang, Xiaoxue Chen, Sicheng Xu, Ruicheng Wang, Zelong Lv, Yu Deng, Hongyuan Zhu, Yue Dong, Hao Zhao, Nicholas Jing Yuan, and Jiaolong Yang. Native and compact structured latents for 3d generation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 14419–14429, 2026. 2, 3

Siming Yan, Zhenpei Yang, Chongyang Ma, Haibin Huang, Etienne Vouga, and Qixing Huang. HPNet: Deep primitive segmentation using hybrid representations. In *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 2733–2742, 2021. 3

Yunhan Yang, Yukun Huang, Yuan-Chen Guo, Liangjun Lu, Xiaoyang Wu, Edmund Y. Lam, Yan-Pei Cao, and Xihui Liu. SAMPart3D: Segment any part in 3d objects. *arXiv preprint arXiv:2411.07184*, 2024. 3

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H. S. Torr, and Vladlen Koltun. Point transformer. In *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 16239–16248, 2021. 3

Zhe Zhu, Le Wan, Rui Xu, Yiheng Zhang, Honghua Chen, Zhiyang Dou, Cheng Lin, Yuan Liu, and Mingqiang Wei. PartSAM: A scalable promptable part segmentation model trained on native 3d data. In *Proc. Int. Conf. Learn. Rep.*, 2026. 3, 7