Name That Part: 3D Part Segmentation and Naming

Soumava Paul* Prakhar Kaushik*‡ Ankit Vaidya Anand Bhattad Alan Yuille
Johns Hopkins University, Baltimore, MD, USA

https://name-that-part.github.io

Figure 1. **ALIGN-Parts** is the first large-scale method to be able to efficiently and semantically segment and name 3D parts of an object, unlike previous methods, which could only perform class-agnostic part segmentation. Our method is also feed-forward, and defines 3D parts according to human-oriented, object-specific affordance descriptions. (left) ALIGN-Parts segments and semantically names 3D parts, unlike PartField [12], which only segments. Our feed-forward method is faster at generating these segments and their names than PartField, as we do not require clustering. (right) Simultaneous part segmentation and naming enable $5-8\times$ more efficient creation of 3D part datasets (bottom) with human-in-the-loop validation.

Abstract

We address semantic 3D part segmentation: decomposing objects into parts with meaningful names. While datasets exist with part annotations, their definitions are inconsistent across datasets, limiting robust training. Previous methods produce unlabeled decompositions or retrieve single parts without complete shape annotations. We propose ALIGN-Parts, which formulates part naming as a direct set alignment task. Our method decomposes shapes into partlets - implicit 3D part representations - matched to part descriptions via bipartite assignment. We combine geometric cues from 3D part fields, appearance from multi-view vision features, and semantic knowledge from language-model-generated affordance descriptions. Text-alignment loss ensures partlets share embedding space with text, enabling open-vocabulary matching. Our efficient and novel, fast one-shot, 3D part segmentation and naming method finds usage in several downstream tasks, including as a scalable annotation engine. As our model supports zero-shot matching to arbitrary descriptions and confidence-calibrated predictions for known categories - with human verification - we create a unified ontology aligning PartNet, 3DCoMPaT++, and Find3D, consisting of 1794 unique 3D parts, and show examples from our newly created Tex-Parts dataset.

1. Introduction

Many vision tasks require 3D parts, not just whole-object labels. Examples include robots grasping handles and creators editing assets. This requires solving two problems at once: geometrically segmenting the parts and semantically naming them. While large datasets of 3D objects exist, only a few provide part annotations, and these annotations are often inconsistent across datasets [14, 15, 17]. An algorithm that can provide accurate and consistent annotations of named parts on any 3D object would enable scalable training data

* Equal Contribution ‡ Project Lead

and support human-in-the-loop annotation pipelines.

Existing methods address only one aspect of this problem. Part segmentation models can identify geometric boundaries but produce unnamed regions [12]. Language-grounded systems can retrieve a single part from a text query but fail to produce a complete set of names for all parts of an object [14]. Classical unsupervised part discovery lacks the semantic grounding to derive consistent definitions. This gap has created a bottleneck: the absence of large-scale, consistently-annotated 3D part data. Inducing consistent parts from unlabeled web assets requires coupling geometric features and semantic knowledge with human verification.

We propose ALIGN-Parts, which formulates 3D part naming as a direct set alignment problem. Rather than deciding per-point which text label to assign, we decompose the shape into a small set of shape-conditioned *partlets*. Each partlet consists of a set of points (a segmentation mask) and a text description (embedding) corresponding to one part. These partlets aggregate information across all points belonging to a part: a single point on a chair seat contains limited information, but the set of all points on the seat specifies the part. We align this set of partlets to a set of candidate part descriptions via bipartite matching in a single forward pass. Each partlet inherits a name from its matched description. A "null" class allows the model to discard unused partlets, enabling the number of parts to adapt per shape while ensuring permutation consistency: each predicted part receives at most one name, and each name is used at most once.

Our key contribution is applying set-level matching to 3D part fields. This enables lower computational complexity: we match a handful of partlets to descriptions instead of all points to all labels. It provides permutation consistency: each predicted part receives at most one name, and each name is used at most once. Because of this, ALIGN-Parts is 100× faster at generating these segments along with names.

To ensure partlets are both geometrically separable and semantically meaningful, we combine three signals. (1) Geometry features from a 3D part-field backbone [12] capture shape structure. (2) Appearance features from multi-view image encoders [16] lifted onto 3D geometry provide texture cues. (3) Semantic knowledge comes from affordance-aware part descriptions that encode form-and-function relationships [5]. For example, a chair seat becomes "the horizontal surface where a person sits," linking its flat, horizontal geometry to its sitting affordance.

For text grounding, we generate these descriptions using a large language model [4] and embed them using MPNet sentence transformers [18]. By representing part descriptions as embeddings in a continuous space, our approach supports scenarios where the model can match partlets to any user-provided set of part descriptions without retraining.

Our approach provides a tool to address the data bottleneck. We construct a unified part ontology using a hybrid LLM-and-human process that normalizes labels and verifies geometric consistency across PartNet, 3DCoMPaT++, and Find3D. We then apply ALIGN-Parts to bootstrap annotations from unlabeled TexVerse assets, creating Tex-Parts: a dataset with 8450 objects spanning 14k part categories. ALIGN-Parts serves as a *scalable annotation engine* that proposes named parts for human verification, converting raw meshes into training data with minimal effort and enabling the construction of large, consistently-annotated 3D datasets. In summary, our contributions are:

- **Direct 3D parts alignment for open-world part naming.** We introduce *partlets* - shape-conditioned part proposals with text embeddings - and match them to candidate descriptions via bipartite assignment, enabling efficient labeling of 3D part segmentation and naming.
- **Geometry-appearance-semantic fusion.** We combine geometric structure, appearance features, and affordance-aware LLM-generated descriptions to produce semantically grounded, visually coherent partlets.
- **Unified ontology and scalable annotation engine.** We harmonize part taxonomies across PartNet, 3DCoMPaT++, and Find3D, and demonstrate a human-in-the-loop pipeline that bootstraps *Tex-Parts*, a verified benchmark of 8450 objects spanning 14k categories derived from Texverse [26]. ALIGN-Parts converts raw meshes into training data with 5-8× less human effort than manual annotation.

2. Related Work

3D Part Segmentation. Traditional methods operate in a purely geometric regime. PartField [12], the current state-of-the-art, learns dense per-point feature fields but produces unlabeled regions. Recent works lift 2D foundation models into 3D: SAM-based approaches [13, 19, 23] adapt Segment-Anything via multi-view projection but require prompts and lack semantic names. PartSTAD [9] adapts GLIP and SAM with 3D-aware objectives, while Diff3F [7] exploits diffusion features, but both operate as multi-stage pipelines without producing complete, non-overlapping named parts.

Language-Grounded 3D Understanding. PartSLIP [11], PartSLIP++ [27], and PartDistill [21] use image-language models for part segmentation but require per-category fine-tuning with predefined vocabularies. Find3D [14] is most related: it trains a point transformer in text embedding space for text-to-part retrieval. However, Find3D operates query-by-query (given "wing, head," it returns masks) rather than producing complete decompositions, and works on individual point features rather than part-level aggregations.

ALIGN-Parts differs fundamentally. We formulate part segmentation and naming as *direct set alignment*: Partlets aggregate point features into part-level representations matched to text via optimal transport, enabling (1) complete, non-

Method	Complete Decomp.	Named Parts	Open Vocab	Permutation Invariant	Part-level Features	Feed Forward
PartField	✓	✗	✗	-	✗	✗
SAMPart3D	✗	✗	✗	✗	✗	✗
Find3D	✗	✓	✓	✗	✗	✗
Ours	✓	✓	✓	✓	✓	✓

Table 1. Comparison with 3D part methods.

overlapping decompositions in one pass, (2) permutation-consistent naming, (3) dynamic part counts, and (4) zero-shot generalization (Table 1). Unlike previous methods, ALIGN-Parts jointly learns segmentation and alignment in a single feed-forward pass. Part count emerges automatically from activated Partlets, eliminating dependence on K prediction or part-name inputs.

Part-Based Generation and Datasets. Generative methods [2, 3, 10, 20, 24] discover part structure through synthesis but lack open-vocabulary naming mechanisms. For datasets, PartNet [15], 3DCoMPaT++ [17], and GAPartNet [8] provide part annotations but use inconsistent taxonomies. ALIGN-Parts constructs a unified ontology across these datasets and creates a unified dataset for cross-dataset evaluation.

3. Method

Overview. We propose ALIGN-Parts, a framework that treats automatic semantic 3D part segmentation as a direct set alignment problem, analogous to DETR [1] in 2D detection. The pipeline consists of three components: (1) a dense feature fusion module combining geometry and appearance via localized bi-directional 3D-aware cross-attention, (2) a Partlets module that learns K adaptive part-level representations by aggregating fused point features, and (3) a semantic grounding module that aligns Partlets to part names via text descriptions of part affordances. At inference, given a 3D shape and candidate part descriptions, our method produces named part segments without specifying part counts, names, or point prompts. Training uses contrastive alignment and differentiable optimal transport matching between predicted Partlets and ground-truth parts, enabling end-to-end learning of both segmentation and semantics.

Problem Formulation. Given a 3D point cloud $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^N$, each point has geometry-first features $\mathbf{f}_i^g \in \mathbb{R}^{d_g}$, appearance-first features $\mathbf{f}_i^a \in \mathbb{R}^{d_a}$, and coordinates $\mathbf{x}_i \in \mathbb{R}^3$. During training, we have ground-truth part masks $\mathbf{M}^{\text{gt}} \in \{0,1\}^{A\times N}$ and corresponding text embeddings $\{\mathbf{t}_a\}_{a=1}^A$ where $\mathbf{t}_a \in \mathbb{R}^{d_t}$ is an MPNet embedding of a language-model-generated part description.

Our model learns $K = 32$ instance-specific Partlet embeddings $\{\mathbf{s}_k \in \mathbb{R}^{d_t}\}_{k=1}^K$ that reside in the same semantic space as text embeddings (we use $d_t = 768$). Each Partlet predicts three outputs: (i) a soft mask $\mathbf{m}_k \in [0,1]^N$ over points, (ii) a partness score $\text{part}_k \in \mathbb{R}$ indicating whether it represents an actual part, and (iii) its embedding $\mathbf{s}_k$ which serves as the prototype for semantic alignment. We select $K = 32$ to accommodate variable part configurations (most shapes have ≤ 28 parts) while allowing future extensibility. Point labels are obtained by matching Partlets to ground-truth parts via optimal transport.

3.1. Architecture

Feature Fusion. The raw geometric features $\mathbf{f}_i^g$ and appearance features $\mathbf{f}_i^a$ capture complementary information: geometry encodes shape characteristics, while appearance provides texture and visual cues. We fuse these modalities through bi-directional cross-attention that we call BiCo Fusion operating on local k=16 nearest neighbor graphs in 3D coordinate space, reducing complexity to $\mathcal{O}(Nk)$.

For each point i with neighbors $\mathcal{N}_i$, geometric features attend to appearance features of neighbors, producing cross-modal features that capture appearance information. Symmetrically, appearance features attend to geometric features of neighbors, capturing geometric information. We incorporate 3D spatial structure through Fourier-encoded relative positional biases. Learned sigmoid gates control how much of this cross-modal information to incorporate into each original feature, based on both the original feature and the attended information. After gated addition and layer normalization, we concatenate both modalities and project through a two-layer MLP to produce fused features $\mathbf{h}_i \in \mathbb{R}^{1024}$ for each point. Complete architectural details are in the supplement.

Points → Partlets: Learning Part-Level Representations. We learn K Partlet embeddings that aggregate point-level information into part-level representations. A Partlet is defined by three components: (i) a soft segmentation mask $\mathbf{m} \in [0,1]^N$ representing membership scores, (ii) a Partlet embedding $\mathbf{s} \in \mathbb{R}^{d_t}$ in the learned semantic space, and (iii) a text embedding $\mathbf{z} \in \mathbb{R}^{d_t}$ representing the part description.

Formally, we learn a parameterized function $f_\theta : \mathbb{R}^{d_h\times N} \to (\mathbb{R}^N \times \mathbb{R}^{d_t})^K$ that maps fused point features $\mathbf{H} = \{\mathbf{h}_i\}_{i=1}^N$ to K Partlets:

$$(\mathbf{m}_k, \mathbf{s}_k) = f_\theta(\mathbf{H}), \quad k = 1, \ldots, K \tag{1}$$

The parameters θ include all weight matrices and biases in the refinement network described below. The motivation for Partlets is: individual point features cannot reliably map to semantic part descriptions (e.g., a single point on a chair seat lacks context to predict "seat"), but aggregating features across all points in a part enables robust semantic grounding.

We initialize K learnable Partlet embeddings $\{\mathbf{s}_k^{(0)} \in \mathbb{R}^{d_t}\}_{k=1}^K$ sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I})$, shared across all shapes but adapted per instance through L refinement layers. At each layer ℓ, each Partlet undergoes three operations:

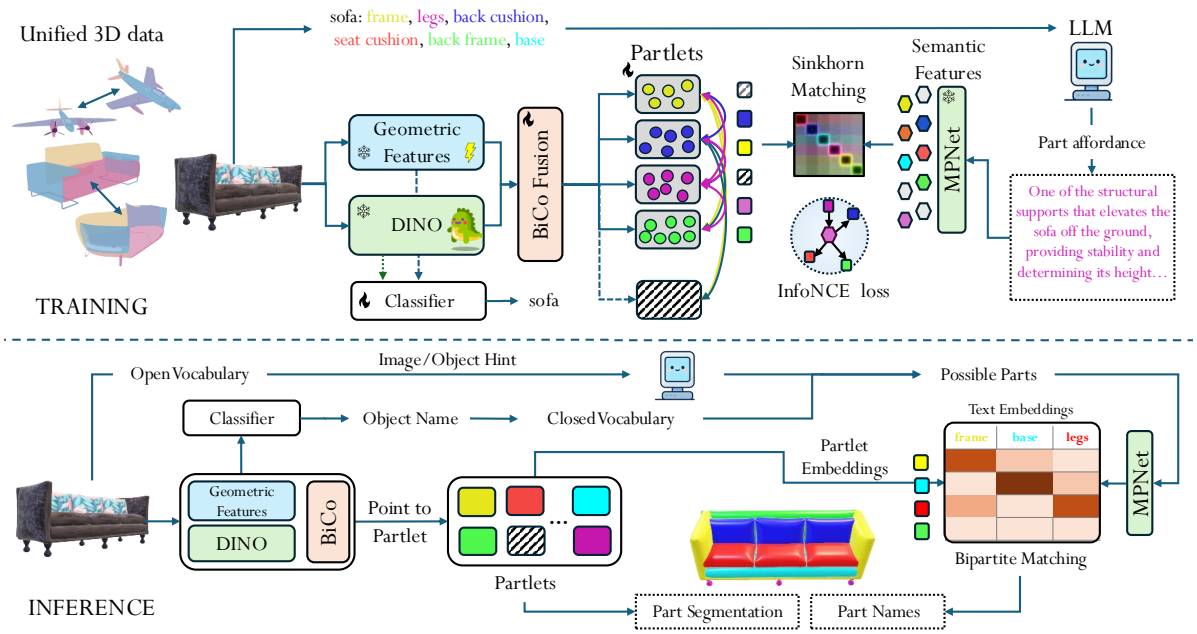

Figure 2. **ALIGN-Parts.** *Top: training.* Given a 3D shape from our semantically unified 3D parts data, geometry features are extracted with PartField and appearance features with DINOv2 from multi-view renderings; the BiCo Fusion module combines these via efficient bi-directional cross-attention on local k=16 nearest-neighbor graphs, reducing complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(Nk)$. A decoder learns K part-level Partlet representations that aggregate these fused features. To handle ambiguous part names (e.g., *legs* for sofa vs. table), an LLM generates affordance-aware descriptions encoded by MPNet; Sinkhorn matching establishes bipartite assignment between Partlet and text embeddings, and an InfoNCE loss aligns them in a shared space. *Bottom: inference.* In closed-vocabulary mode, a trained classifier predicts the object class and retrieves candidate parts; in open-vocabulary mode, an LLM proposes plausible parts. Matched Partlets jointly produce 3D segmentation masks and part names.

Partlet-to-Partlet Interaction: Partlets interact to model part co-occurrence (e.g., chairs have seats and backs):

$$\mathbf{s}_k^{(\ell,1)} = \mathbf{s}_k^{(\ell-1)} + \text{SelfAttn}(\mathbf{s}_k^{(\ell-1)}, \{\mathbf{s}_{k'}\}_{k'=1}^{K}) \tag{2}$$

Point-to-Partlet Aggregation: Partlets gather shape-specific evidence from BiCo-fused point features:

$$\mathbf{s}_k^{(\ell,2)} = \mathbf{s}_k^{(\ell,1)} + \text{CrossAttn}(\mathbf{s}_k^{(\ell,1)}, \{\mathbf{h}_i\}_{i=1}^{N}) \tag{3}$$

Non-linear Transformation: Two-layer MLP with GELU:

$$\mathbf{s}_k^{(\ell)} = \mathbf{s}_k^{(\ell,2)} + \text{MLP}(\mathbf{s}_k^{(\ell,2)}) \tag{4}$$

After L layers, we obtain refined Partlet embeddings $\mathbf{s}_k = \mathbf{s}_k^{(L)}$.

Mask Prediction. Each Partlet predicts point membership via scaled dot-product:

$$m_{ki} = \frac{(\mathbf{W}_q\mathbf{s}_k)^T(\mathbf{W}_k\mathbf{h}_i)}{\sqrt{d_t}} \tag{5}$$

During training, we apply sigmoid activation; at inference, softmax across Partlets yields soft assignments.

Partness Prediction. Each Partlet predicts whether it represents an actual part:

$$\text{part}_k = \mathbf{w}_{\text{part}}^T\mathbf{s}_k + b_{\text{part}} \tag{6}$$

Higher values indicate active Partlets; lower values signify "no-part," enabling dynamic part count adaptation per shape.

Semantic Alignment: Partlet → Part Names. Each Partlet's refined embedding $\mathbf{s}_k$ serves directly as its prototype $\mathbf{z}_k = \mathbf{s}_k \in \mathbb{R}^{d_t}$. After normalization ($\hat{\mathbf{z}}_k = \mathbf{z}_k/\|\mathbf{z}_k\|_2$), we compute cosine similarity with text embeddings $\hat{\mathbf{t}}_a$:

$$\text{sim}(k, a) = \hat{\mathbf{z}}_k \cdot \hat{\mathbf{t}}_a \tag{7}$$

Importantly, Partlet and text embeddings share the same semantic space ($\mathbb{R}^{d_t}$) without intermediate projections—this alignment is driven by the text alignment loss (Section 3.2), enabling open-vocabulary matching at inference without retraining. However, part names are often ambiguous — *handle* could refer to a door handle, a mug handle, or a wheelchair handle, each with distinct geometry and function. To disambiguate, we use affordance-based descriptions (e.g., "the part of a door grasped to open it" for a door handle, "the horizontal surface of a chair where a person sits" for a chair seat) generated by Gemini 2.5 Flash. These are embedded with MPNet (all-mpnet-base-v2), which we found

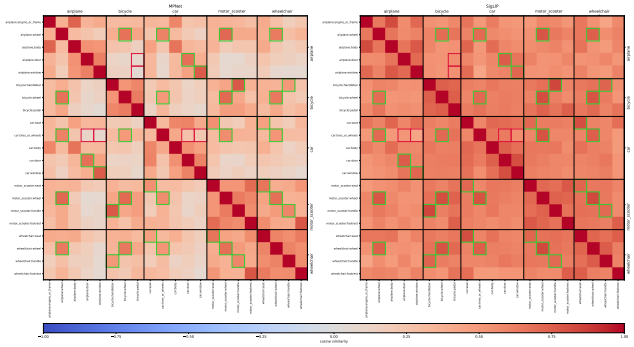

Figure 3. Pairwise cosine similarity heatmaps between text embeddings for MPNet (left) and SigLiP (right). (Zoom in for labels)

more robust than CLIP or SigLIP [25] for capturing long-form semantic affordances (see Fig. 3), especially for similar part names.

3.2. Training: Partlet ↔ Part Names

We establish correspondences between predicted Partlets and ground-truth parts via differentiable optimal transport. The cost matrix $\mathbf{C} \in \mathbb{R}^{K \times A}$ combines mask overlap and semantic similarity:

$$C_{ka} = \mathcal{L}_{\text{mask}}^{(k,a)} + (1 - \text{sim}(\hat{\mathbf{z}}_k, \hat{\mathbf{t}}_a)) \tag{8}$$

where $\mathcal{L}_{\text{mask}}^{(k,a)} = 1 - \text{Dice}(\sigma(\mathbf{m}_k), \mathbf{m}_a^{\text{gt}})$ (equal weighting $\alpha = \beta = 1.0$).

Sinkhorn-Knopp iterations produce a soft assignment matrix $\mathbf{P} \in [0,1]^{K \times A}$. Thresholding yields hard assignments $\pi : \{1, \ldots, K\} \to \{1, \ldots, A\} \cup \{\emptyset\}$, where $\pi(k) = a$ matches Partlet k to part a and $\pi(k) = \emptyset$ indicates no match.

Losses. Let $\mathcal{M} = \{k : \pi(k) \neq \emptyset\}$ denote matched Partlets. Our training objective combines several losses:

Text Alignment Loss. This loss is essential for open-vocabulary grounding. Without it, Partlet embeddings remain geometrically meaningful but semantically ambiguous. We apply InfoNCE contrastive loss to align matched Partlets with their text embeddings:

$$\mathcal{L}_{\text{text}} = \frac{1}{|\mathcal{M}|} \sum_{k \in \mathcal{M}} -\log \frac{\exp(\hat{\mathbf{z}}_k \cdot \hat{\mathbf{t}}_{\pi(k)}/\tau)}{\sum_{a=1}^{A} \exp(\hat{\mathbf{z}}_k \cdot \hat{\mathbf{t}}_a/\tau)} \tag{9}$$

with $\tau = 0.07$. Operating over Partlets rather than individual points makes this optimization tractable and stable.

Mask and Partness Losses. For matched Partlets, binary cross-entropy and Dice loss supervise masks:

$$\begin{aligned}\mathcal{L}_{\text{mask}} = \frac{1}{|\mathcal{M}|} \sum_{k \in \mathcal{M}} [&\text{BCE}(\mathbf{m}_k, \mathbf{m}_{\pi(k)}^{\text{gt}}) \\ &+ (1 - \text{Dice}(\sigma(\mathbf{m}_k), \mathbf{m}_{\pi(k)}^{\text{gt}}))]\end{aligned} \tag{10}$$

Binary classification loss supervises partness, teaching Partlets to predict whether they are active (matched) or inactive ("no-part"):

$$\mathcal{L}_{\text{part}} = \frac{1}{K} \sum_{k=1}^{K} \text{BCE}(\text{part}_k, \mathbf{1}[\pi(k) \neq \emptyset]) \tag{11}$$

Auxiliary Regularizers. Coverage loss prevents over-/under-segmentation by penalizing mask size disparities:

$$\mathcal{L}_{\text{cov}} = \frac{1}{|\mathcal{M}|} \sum_{k \in \mathcal{M}} \left| \frac{\sum_i \sigma(m_{ki}) - \sum_i m_{\pi(k)i}^{\text{gt}}}{N} \right| \tag{12}$$

Overlap loss enforces mutual exclusivity—each point should belong to at most one part:

$$\mathcal{L}_{\text{overlap}} = \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{k=1}^{K} \sigma(m_{ki}) - 1 \right)^2 \tag{13}$$

Global Alignment Loss. Symmetric InfoNCE aligns the global shape representation with class-level text embeddings, providing object-level semantic context:

$$\mathcal{L}_{\text{global}} = \frac{1}{2} \left[\mathcal{L}_{\text{CE}}(\mathbf{S}, \mathbf{I}) + \mathcal{L}_{\text{CE}}(\mathbf{S}^\top, \mathbf{I}) \right] \tag{14}$$

where $\mathbf{S} = \frac{1}{\tau} \hat{\mathbf{Z}}_{\text{global}} \hat{\mathbf{T}}_{\text{class}}^\top$ and $\lambda_{\text{global}} = 0.5$.

Total Loss. The complete training objective as a function of model parameters θ is:

$$\begin{aligned}\mathcal{L}_{\text{total}}(\theta) &= \lambda_{\text{mask}} \mathcal{L}_{\text{mask}}(\theta, \pi) + \lambda_{\text{part}} \mathcal{L}_{\text{part}}(\theta, \pi) \\ &+ \lambda_{\text{text}} \mathcal{L}_{\text{text}}(\theta, \pi) + \lambda_{\text{cov}} \mathcal{L}_{\text{cov}}(\theta, \pi) + \lambda_{\text{ov}} \mathcal{L}_{\text{overlap}}(\theta, \pi) \\ &+ \lambda_{\text{global}} \mathcal{L}_{\text{global}}(\theta)\end{aligned} \tag{15}$$

where θ includes all learnable parameters (BiCo fusion weights, Partlet decoder layers, prediction heads), π is the assignment from Partlets to ground-truth parts, computed via Sinkhorn matching given the current model predictions. All hyperparameters are in the supplement.

3.3. Inference Pipelines and Use Cases

Our model supports three inference modes for different deployment scenarios.

Mode 1: Closed-Vocabulary with Confidence Calibration. We use ALIGN-Parts to scalably annotate large datasets from known categories (e.g., labeling millions of airplane meshes). This is the most practical scenario, enabling efficient curation with minimal oversight.

For training categories $\mathcal{C}$, we predict the object category via global shape-text alignment:

$$c^* = \arg\max_{c \in \mathcal{C}} \text{sim}(\mathbf{z}_{\text{global}}, \mathbf{t}_c) \tag{16}$$

then filter Partlets by partness ($\sigma(\text{part}_k)$>0.5) to obtain active set $\mathcal{K}_{\text{active}}$. We construct cost matrix $\mathbf{C} \in \mathbb{R}^{|\mathcal{K}_{\text{active}}| \times |\mathcal{L}_{c^*}|}$:

$$C_{ka} = 1 - \text{sim}(\hat{\mathbf{z}}_k, \hat{\mathbf{t}}_a) \tag{17}$$

At inference we use the Jonker-Volgenant algorithm [6] for exact optimal assignment—more efficient than Sinkhorn since gradients are not needed.

Because part labels in $\mathcal{L}_{c^*}$ are known from training, we compute Mahalanobis-distance confidence scores:

$$\text{conf}_{\text{maha}}(k) = \exp(-(\mathbf{z}_k - \boldsymbol{\mu}_{a_k^*})^T \boldsymbol{\Sigma}_{a_k^*}^{-1} (\mathbf{z}_k - \boldsymbol{\mu}_{a_k^*})) \tag{18}$$

where $\boldsymbol{\mu}_{a_k^*}$ and $\boldsymbol{\Sigma}_{a_k^*}$ are estimated from training embeddings. Predictions with $\text{conf}_{\text{maha}} > 0.8$ are auto-accepted; lower-confidence predictions are flagged for human verification, dramatically reducing annotation cost.

Mode 2: Open-Vocabulary Grounding. For novel categories, users provide candidate part descriptions $\{\mathbf{t}_a\}_{a=1}^A$ (or generate via LLM from image/hint). After filtering inactive Partlets, each active Partlet matches to the best description:

$$a_k^* = \underset{a \in \{1,\ldots,A\}}{\arg\max}\ \text{sim}(\hat{\mathbf{z}}_k, \hat{\mathbf{t}}_a) \tag{19}$$

Confidence is computed via temperature-calibrated softmax:

$$\text{conf}_{\text{soft}}(k) = \max_a \frac{\exp(\hat{\mathbf{z}}_k \cdot \hat{\mathbf{t}}_a/\tau)}{\sum_{a'=1}^A \exp(\hat{\mathbf{z}}_k \cdot \hat{\mathbf{t}}'_a/\tau)} \tag{20}$$

with $\tau = 0.07$. This is less calibrated than Mahalanobis distance (Mode 1) due to lack of training statistics for novel categories.

Mode 3: Text-Conditioned Part Retrieval. For compatibility with Find3D [14], we retrieve a single part for query $\mathbf{t}_q$:

$$k^* = \underset{k \in \mathcal{K}_{\text{active}}}{\arg\max}\, \text{sim}(\hat{\mathbf{z}}_k, \hat{\mathbf{t}}_q) \tag{21}$$

returning mask $\mathbf{m}_{k^*}$. This mode is primarily for benchmark comparison.

Point Label Assignment. Points are assigned to the highest-scoring Partlet's label:

$$\text{label}_i = a_{k^*}^* \quad \text{where} \quad k^* = \underset{k \in \mathcal{K}_{\text{active}}}{\arg\max}\, \sigma(m_{ki}) \tag{22}$$

Points with $\max_k \sigma(m_{ki}) < 0.5$ remain unlabeled.

3.4. Training Details

Training Setup. ALIGN-Parts is trained on 3 NVIDIA A6000 GPUs for 2 days (batch size 16). **Due to academic compute constraints, we sample 10k points per shape (vs. 100k in PartField)** while maintaining strong performance. Models are normalized to $[-1, 1]^3$ during training.

Datasets. We train on 40,982 shapes from three datasets: 3DCoMPaT++ (8,627), PartNet (32,141), and Find3D (124). All use fine-grained part labels. For evaluation, we hold out 206 shapes: 126 objects (42 categories) from 3DCoMPaT++, 72 objects (24 categories) from PartNet, and 8 novel objects (8 categories) from Find3D.

4. Experiments

4.1. Aligning 3D Part Annotations across Datasets

We harmonize heterogeneous part taxonomies through a two-stage pipeline. First, MPNet embeddings identify similar candidates (cosine similarity > 0.85). Then, an LLM validates semantic equivalence, rejecting spurious matches.

For example, *microwave_oven* (3DCoMPaT) and *microwave* (PartNet) with similarity 0.902 were merged after Gemini confirmed they "describe the same kitchen appliance." Within this class, *door_handle* and *handle* (similarity 0.904) were unified as "external gripping components for opening doors." However, the LLM correctly preserves distinct concepts: despite similarity 0.891, *bed_side_surface_horizontal_bar* and *vertical_bar* remain separate because "their orientation and functions differ significantly." This compressed vocabulary enables training with unified part semantics.

4.2. Comparison on Named-3DPart Segmentation

Baselines. While no prior work addresses *named* 3D part segmentation end-to-end, we compare against two recent baselines: PartField [12] and Find3D [14]. PartField achieves state-of-the-art class-agnostic segmentation but cannot name parts. It clusters a learned feature field (requiring ground-truth part count K). Find3D matches per-point features to SigLIP embeddings of *provided* part queries, needing the list of ground-truth part names as input.

PartField+MPNet (Our Strong Baseline; w/o partlets). We extend PartField with semantic alignment: a linear head maps PartField features to a shared space with MPNet text embeddings, trained via InfoNCE loss. However, it still requires predicting K via an auxiliary classifier, making it brittle to over/under-segmentation from K prediction errors.

Metrics. We report mIoU following [12, 23]: for each ground-truth part, we take the maximum IoU across all predicted parts and average these values. However, mIoU ignores whether the matched prediction has the correct semantic label. To evaluate label accuracy, we introduce **Label-Aware mIoU (LA-mIoU)**: a part receives LA-mIoU = IoU only if the predicted and ground-truth labels match, otherwise 0.0.

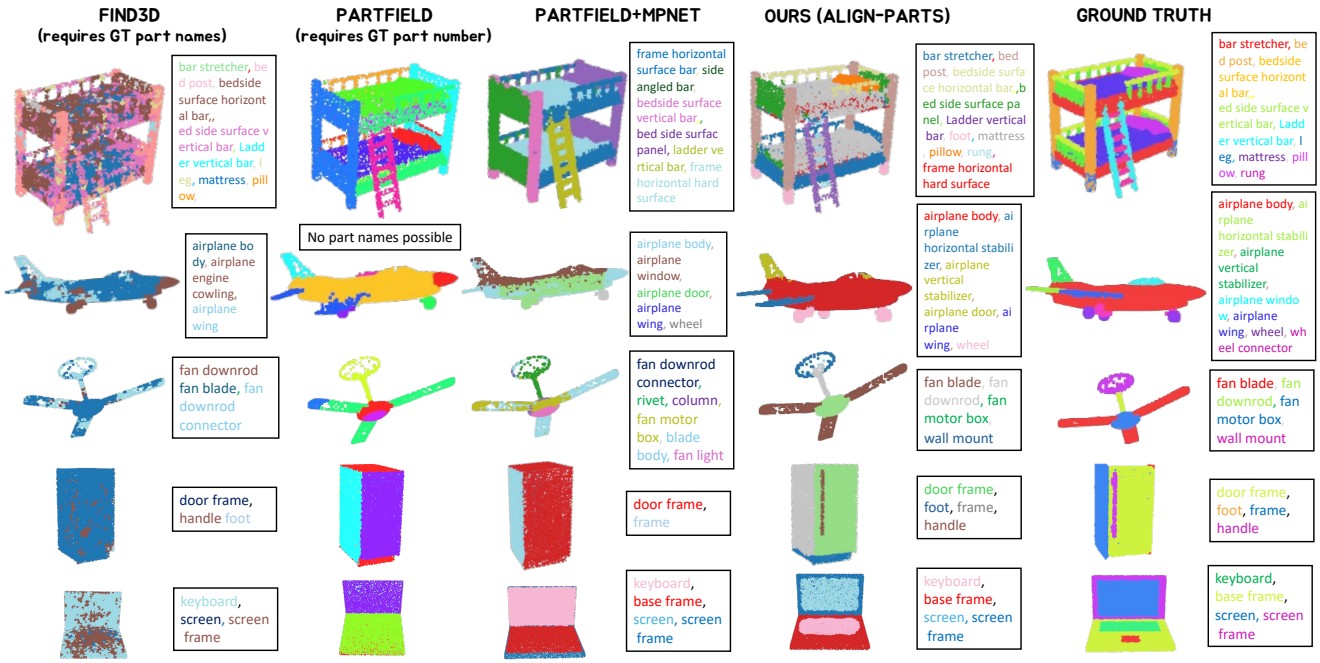

Figure 4. **Qualitative Results.** ALIGN-Parts robustly segments and names 3D parts in a single feed-forward pass (rightmost column). Find3D [14] (first column) fails despite ground-truth part names, unable to segment the laptop in Figure 4 (bottom row). PartField [12] (second column) also fails: it requires ground-truth part counts for clustering, missegments bed bunks (top row), and misses refrigerator handles (second-to-last row). Our strong baseline without Partlets (third column) exhibits similar errors. In contrast, ALIGN-Parts correctly segments tiny parts like handles and groups semantically similar instances (e.g., all ceiling fan blades into single clusters).

Table 2. Evaluation of ALIGN-Parts and related baselines on our test set (never seen during training).

Variant	3DCoMPaT (126)		Find3D (8)		PartNet (72)		Average		Runtime (s)↓
	mIoU↑	LA-mIoU↑	mIoU↑	LA-mIoU↑	mIoU↑	LA-mIoU↑	mIoU↑	LA-mIoU↑	
PartField	0.371	n/a	0.662	n/a	0.521	n/a	0.518	n/a	4.007
Find3D	0.239	0.095	0.379	0.259	0.354	0.232	0.324	0.196	0.249
PartField+MPNet	0.316	0.185	0.590	0.137	0.446	0.276	0.451	0.199	0.026
ALIGN-Parts	0.453	0.284	0.595	0.133	0.753	0.552	0.600	0.323	0.047

5. Results

As shown in Figures 4 and 6 and Table 2, ALIGN-Parts comprehensively outperforms all baselines on both class-agnostic segmentation (mIoU) and named part segmentation (LA-mIoU). On average mIoU, we outperform PartField by 15.8%, and on LA-mIoU, we improve over PartField+MPNet by 62.3% while being 100× faster because we do not require running a K-means clustering algorithm.

Qualitatively, our baselines show several weaknesses. PartField often fragments instances of the same part into multiple segments after clustering, contradicting human labeling conventions in 3DCoMPaT++ and PartNet. This likely stems from its use of SAM to extract unlabeled parts from 2D renderings. In contrast, ALIGN-Parts, trained on human annotations, correctly groups instances: e.g., all four bed posts of a double bed and all three wheels of an airplane are identified as single parts (Figure 6).

Find3D learns per-point semantic vectors without considering shape geometry, resulting in noisy and overlapping segmentations. PartField+MPNet's reliance on predicted cluster counts leads to under-segmentation, as fine parts such as the refrigerator handle and laptop screen frame are missed (Figure 6). Both Find3D and PartField fail to segment even simple structures, such as fan blades, whereas ALIGN-Parts accurately segments and annotates fine-grained parts in complex shapes.

Fine-Part Localization Despite using only 10k sampled points, ALIGN-Parts segments fine parts like the *screw* of scissors (Fig 6) - structures that PartField, trained with 100k points, cannot localize.

Ablations. Table 3 and Figure 5 evaluate ALIGN-Parts's design choices, showing quantitative and qualitative improvements from each component.

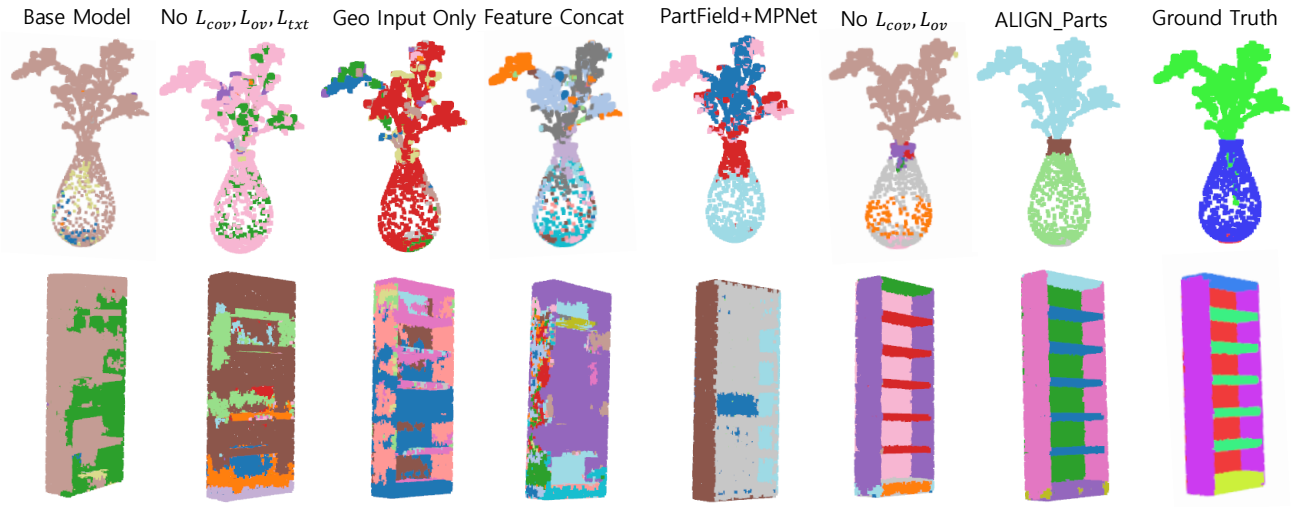

Figure 5. **Ablation (Qualitative results).** Results improve from left to right as components are added sequentially, with ground truth in the final column. The first five columns result in significant misalignments and segmentation leakage. Neither geometric-only features nor naive DINO concatenation improves performance. Major gains arise from Partlets (see sixth and seventh columns) and the coverage loss, which refine fine details and are consistent with quantitative metrics in Table 3.

Table 3. Ablation study on 3DCoMPaT, Find3D, and PartNet. We report mIoU and label-aware mIoU (LA-mIoU).

Variant	3DCoMPaT (126)		Find3D (8)		PartNet (72)		Average	
	mIoU↑	LA-mIoU↑	mIoU↑	LA-mIoU↑	mIoU↑	LA-mIoU↑	mIoU↑	LA-mIoU↑
Base Model	0.228	0.031	0.374	0.046	0.335	0.044	0.312	0.040
No $\mathcal{L}_{\text{cov}}$, $\mathcal{L}_{\text{ov}}$, $\mathcal{L}_{\text{txt}}$	0.233	0.030	0.382	0.018	0.357	0.049	0.324	0.032
No $\mathcal{L}_{\text{cov}}$, $\mathcal{L}_{\text{ov}}$	0.422	0.230	0.499	0.116	0.664	0.445	0.528	0.264
Geo Input Only	0.224	0.022	0.384	0.027	0.332	0.049	0.313	0.033
Feature Concat	0.221	0.030	0.367	0.067	0.317	0.061	0.302	0.053
PartField+MPNet	0.316	0.185	0.590	0.137	0.446	0.276	0.451	0.199
ALIGN-Parts	0.453	0.284	0.595	0.133	0.753	0.552	0.600	0.323

Figure 6. ALIGN-Parts correctly segments the tiny screw despite training with sparser points (10k vs. 100k).

Baseline Comparisons. Using only PartField geometric features, our base model performs slightly better than naive concatenation with DINOv2, indicating simple multi-modal fusion can be counterproductive. Raw geometry without PartField shows no improvement, confirming learned geometric features are essential.

Progressive Component Addition. Adding DINOv2 yields modest gains. The InfoNCE text alignment loss ($\mathcal{L}_{\text{text}}$) dramatically improves LA-mIoU; this variant is the only one besides the full model that correctly segments fine-grained parts like plants in vases. Adding $\mathcal{L}_{\text{cov}}$ and $\mathcal{L}_{\text{overlap}}$ yields the complete ALIGN-Parts with peak performance.

Further Applications. We show results on PartObjaVerse-Tiny in the supplementary. Despite most parts being unseen during training, ALIGN-Parts performs well in this OOD setting, producing both segmentation and annotations.

6. Discussion

ALIGN-Parts reframes *semantic* 3D part segmentation as set alignment, where Partlets jointly learn geometry and semantics via optimal transport without part count supervision. Unlike Find3D's per-point matching or PartField's clustering, our end-to-end approach produces fast, coherent, named parts. Affordance-aware descriptions disambiguate fine-grained parts, and MPNet sentence embeddings outperform CLIP/SigLIP. Despite 10× fewer points, we achieve superior fine-part localization, confirming part-level representations are more data-efficient than dense features. Limitations and future directions are in the supplementary.

7. Acknowledgements

The authors gratefully acknowledge support from the Office of Naval Research (ONR) under Grant N00014-23-1-2641, the National Eye Institute (NEI) under Award R01EY037193. This work was also supported in part by the Whiting School of Engineering.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 3

[2] Minghao Chen, Roman Shapovalov, Iro Laina, Tom Monnier, Jianyuan Wang, David Novotny, and Andrea Vedaldi. Partgen: Part-level 3d generation and reconstruction with multi-view diffusion models. *arXiv preprint arXiv:2412.18608*, 2024. 3

[3] Minghao Chen, Jianyuan Wang, Roman Shapovalov, Tom Monnier, Hyunyoung Jung, Dilin Wang, Rakesh Ranjan, Iro Laina, and Andrea Vedaldi. Autopartgen: Autogressive 3d part generation and discovery. *arXiv preprint arXiv:2507.13346*, 2025. 3

[4] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 2

[5] Jonathan H Connell and Michael Brady. Generating and generalizing models of visual objects. *Artificial Intelligence*, 31(2):159–183, 1987. 2

[6] David F. Crouse. On implementing 2d rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696, 2016. 6

[7] Niladri Shekhar Dutt, Sanjeev Muralikrishnan, and Niloy J Mitra. Diffusion 3d features (diff3f): Decorating untextured shapes with distilled semantic features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4494–4504, 2024. 2

[8] Haoran Geng, Helin Xu, Chengyang Zhao, Chao Xu, Li Yi, Siyuan Huang, and He Wang. Gapartnet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7081–7091, 2023. 3

[9] Hyunjin Kim and Minhyuk Sung. Partstad: 2d-to-3d part segmentation task adaptation. In *ECCV*, 2024. 2

[10] Yuchen Lin, Chenguo Lin, Panwang Pan, Honglei Yan, Yiqiang Feng, Yadong Mu, and Katerina Fragkiadaki. Partcrafter: Structured 3d mesh generation via compositional latent diffusion transformers, 2025. 3

[11] Minghua Liu, Yinhao Zhu, Hong Cai, Shizhong Han, Zhan Ling, Fatih Porikli, and Hao Su. Partslip: Low-shot part segmentation for 3d point clouds via pretrained image-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21736–21746, 2023. 2

[12] 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. *arXiv preprint arXiv:2504.11451*, 2025. 1, 2, 6, 7

[13] 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, 2025. 2

[14] Ziqi Ma, Yisong Yue, and Georgia Gkioxari. Find any part in 3d. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7818–7827, 2025. 1, 2, 6, 7

[15] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 3

[16] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 2

[17] Habib Slim, Xiang Li, Yuchen Li, Mahmoud Ahmed, Mohamed Ayman, Ujjwal Upadhyay, Ahmed Abdelreheem, Arpit Prajapati, Suhail Pothigara, Peter Wonka, and Mohamed Elhoseiny. 3DCoMPaT++: An improved large-scale 3D vision dataset for compositional recognition. *arXiv preprint*, 2023. 1, 3

[18] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33:16857–16867, 2020. 2

[19] George Tang, William Zhao, Logan Ford, David Benhaim, and Paul Zhang. Segment any mesh. *arXiv preprint arXiv:2408.13679*, 2024. 2

[20] Jiaxiang Tang, Ruijie Lu, Zhaoshuo Li, Zekun Hao, Xuan Li, Fangyin Wei, Shuran Song, Gang Zeng, Ming-Yu Liu, and Tsung-Yi Lin. Efficient part-level 3d object generation via dual volume packing. *arXiv preprint arXiv:2506.09980*, 2025. 3

[21] Ardian Umam, Cheng-Kun Yang, Min-Hung Chen, Jen-Hui Chuang, and Yen-Yu Lin. Partdistill: 3d shape part segmentation by vision-language model distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3470–3479, 2024. 2

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[23] 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. 2, 6

[24] Yunhan Yang, Yuan-Chen Guo, Yukun Huang, Zi-Xin Zou, Zhipeng Yu, Yangguang Li, Yan-Pei Cao, and Xihui Liu. Holopart: Generative 3d part amodal segmentation. *arXiv preprint arXiv:2504.07943*, 2025. 3

[25] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023. 5

[26] Yibo Zhang, Li Zhang, Rui Ma, and Nan Cao. Texverse: A universe of 3d objects with high-resolution textures. *arXiv preprint arXiv:2508.10868*, 2025. 2, 7, 8

[27] Yuchen Zhou, Jiayuan Gu, Xuanlin Li, Minghua Liu, Yunhao Fang, and Hao Su. Partslip++: Enhancing low-shot 3d part segmentation via multi-view instance segmentation and maximum likelihood estimation. *arXiv preprint arXiv:2312.03015*, 2023. 2

Name That Part: 3D Part Segmentation and Naming

Supplementary Material

8. Architecture Details

This section provides in-depth details regarding our model's architecture.

8.1. Dense Feature Fusion Module

As described in Section 3.3.1 of the main paper, we fuse geometric ($\mathbf{f}_i^g$) and appearance ($\mathbf{f}_i^a$) features using a bi-directional co-attention module that operates on a k-nearest neighbor (KNN) graph to maintain computational tractability. Here, we provide the detailed equations for this module. Given the KNN graph structure with indices $\mathcal{N}_i$ denoting the k-nearest neighbors of point i, we compute:

Relative Positional Bias. For each neighbor pair (i, j) where $j \in \mathcal{N}_i$, we compute:

$$\mathbf{d}_{ij} = \mathbf{x}_j - \mathbf{x}_i \in \mathbb{R}^3 \tag{23}$$

$$\mathbf{f}_{ij} = \mathbf{d}_{ij} \odot \omega \in \mathbb{R}^{3\times F} \tag{24}$$

$$\mathbf{h}_{ij} = [\sin(\mathbf{f}_{ij}), \cos(\mathbf{f}_{ij})] \in \mathbb{R}^{3\times 2F} \tag{25}$$

$$\mathbf{b}_{ij} = \text{MLP}([\mathbf{d}_{ij}, \text{flatten}(\mathbf{h}_{ij})]) \in \mathbb{R}^H \tag{26}$$

where $\omega = [2^0, 2^1, \ldots, 2^{F-1}]$ with F=6 are logarithmically-spaced frequencies, and the MLP consists of two layers: $\mathbb{R}^{39} \to \mathbb{R}^{64} \to \mathbb{R}^H$ with ReLU activation.

Bi-Directional Cross-Attention. Let $H = 8$ be the number of attention heads and $d_h = d_m/H = 96$ be the head dimension where $d_m = 768$ is the model dimension.

Geometric-to-Appearance Direction:

$$\mathbf{Q}_p^i = \mathbf{W}_q^p \mathbf{f}_i^g \in \mathbb{R}^{H\times d_h} \tag{27}$$

$$\mathbf{K}_a^{ij} = \mathbf{W}_k^a \mathbf{f}_j^a \in \mathbb{R}^{H\times d_h}, \quad \forall j \in \mathcal{N}_i \tag{28}$$

$$\mathbf{V}_a^{ij} = \mathbf{W}_v^a \mathbf{f}_j^a \in \mathbb{R}^{H\times d_h}, \quad \forall j \in \mathcal{N}_i \tag{29}$$

$$\alpha_{pa}^{i,h,j} = \frac{\exp((\mathbf{Q}_p^i[h] \cdot \mathbf{K}_a^{ij}[h])/\sqrt{d_h} + \mathbf{b}_{ij}[h])}{\sum_{j'\in\mathcal{N}_i} \exp((\mathbf{Q}_p^i[h] \cdot \mathbf{K}_a^{ij'}[h])/\sqrt{d_h} + \mathbf{b}_{ij'}[h])} \tag{30}$$

$$\mathbf{z}_p^{i,h} = \sum_{j\in\mathcal{N}_i} \alpha_{pa}^{i,h,j} \mathbf{V}_a^{ij}[h] \tag{31}$$

$$\mathbf{r}_p^i = \mathbf{W}_{pa} \text{concat}_h[\mathbf{z}_p^{i,h}] \in \mathbb{R}^{d_g} \tag{32}$$

Appearance-to-Geometric Direction: Symmetric formulation produces $\mathbf{r}_a^i \in \mathbb{R}^{d_a}$.

Gated Fusion.

$$\mathbf{g}_p^i = \sigma(\mathbf{W}_g^p[\mathbf{f}_i^g; \mathbf{r}_p^i]) \in \mathbb{R}^{d_g} \tag{33}$$

$$\mathbf{g}_a^i = \sigma(\mathbf{W}_g^a[\mathbf{f}_i^a; \mathbf{r}_a^i]) \in \mathbb{R}^{d_a} \tag{34}$$

$$\tilde{\mathbf{f}}_i^g = \text{LayerNorm}(\mathbf{f}_i^g + \mathbf{g}_p^i \odot \mathbf{r}_p^i) \tag{35}$$

$$\tilde{\mathbf{f}}_i^a = \text{LayerNorm}(\mathbf{f}_i^a + \mathbf{g}_a^i \odot \mathbf{r}_a^i) \tag{36}$$

Final Projection.

$$\mathbf{h}_i = \mathbf{W}_2 \text{GELU}(\mathbf{W}_1 \text{LayerNorm}([\tilde{\mathbf{f}}_i^g; \tilde{\mathbf{f}}_i^a])) \tag{37}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d_f\times(d_g+d_a)}$ and $\mathbf{W}_2 \in \mathbb{R}^{d_f\times d_f}$ with $d_g = 448$, $d_a = 768$, and $d_f = 256$.

9. Inference and Confidence Calibration Details

Here, we provide the detailed formulations for the "Mode 1: Closed-Vocabulary Confidence Calibration" pipeline described in the main paper.

9.1. Mahalanobis Parameter Estimation

The Mahalanobis confidence (**Eq. 18**) requires class-conditional statistics (mean and covariance) that are estimated from the training set.

After training, we perform a single forward pass over the entire training dataset. For every query k that is successfully matched to a ground-truth part label ℓ (i.e., $\pi(k) = \ell$), we extract its prototype embedding $\mathbf{z}_k$.

We then compute the empirical mean $\boldsymbol{\mu}_\ell$ for each part label ℓ in our known training vocabulary $\mathcal{C}$:

$$\boldsymbol{\mu}_\ell = \mathbb{E}[\mathbf{z}_k | \pi(k) = \ell] \tag{38}$$

For robustness, we compute a single, shared covariance matrix $\boldsymbol{\Sigma}$ by pooling the embeddings from all part classes:

$$\boldsymbol{\Sigma} = \text{Cov}(\{\mathbf{z}_k\}_{\forall k, \ell \text{ s.t. } \pi(k)=\ell}) \tag{39}$$

We apply regularization (e.g., adding a small value $\epsilon\mathbf{I}$ to the diagonal) before computing the inverse $\boldsymbol{\Sigma}^{-1}$ to ensure numerical stability. These pre-computed $\boldsymbol{\mu}_\ell$ and $\boldsymbol{\Sigma}^{-1}$ are stored and used at inference time for Mode 2.

9.2. Fused Confidence Formulation

As mentioned in the main paper, the final confidence score $\text{conf}(k)$ for a matched query k is a fusion of the softmax confidence ($\text{conf}_{\text{soft}}$) and the Mahalanobis confidence ($\text{conf}_{\text{maha}}$). We combine them as follows:

$$\text{conf}(k) = \alpha \cdot \text{conf}_{\text{soft}}(k) + (1-\alpha) \cdot \sigma(\beta \cdot (\text{conf}_{\text{maha}}(k) - 0.5)) \tag{40}$$

where $\sigma(\cdot)$ is the sigmoid function.

- $\text{conf}_{\text{soft}}(k)$ is the temperature-calibrated softmax score (**Eq. 20**).
- $\text{conf}_{\text{maha}}(k)$ is the Mahalanobis confidence (**Eq. 18**).
- α and β are hyperparameters that balance the two scores. We set $\alpha = 0.5$ and $\beta = 1.0$ based on calibration on a held-out validation set.

Annotations with $\text{conf}(k) < \tau_{\text{conf}}$ (where $\tau_{\text{conf}} = 0.5$) are flagged as low-confidence and routed to a human annotator for manual review.

Architecture. The model has 34M parameters total: 5.7M for feature Fusion, 26.8M for Partlets, and 1.5M for the global classifier. Feature interactions (Partlet-to-points and Partlet-to-Partlet) use 3 transformer [22] blocks with multi-head cross-attention, LayerNorm, residual connections, and feedforward layers. The BiCo fusion employs sparse 16-NN attention with 3D relative positional bias computed via a learned MLP over Fourier-encoded (F=6 frequencies) displacement vectors, providing geometric context while maintaining $\mathcal{O}(Nk)$ complexity. We also note that for calculating the runtime we do not include data preprocessing time as they vary depending on parallelization and system capabilities.

We set the number of Partlets to 32, as this value provides a reasonable estimate for the typical number of semantic parts found in most objects in our unified dataset. This choice is further validated by analyzing the statistics of part counts across the full dataset, which confirm that 32 accommodates the majority of objects without excessive oversegmentation or loss of fine granularity.

Optimization. Loss weights: $\lambda_{\text{mask}} = 1.0$, $\lambda_{\text{part}} = 0.5$, $\lambda_{\text{text}} = 1.0$, $\lambda_{\text{cov}} = 0.5$, $\lambda_{\text{overlap}} = 0.1$, $\lambda_{\text{global}} = 1.0$. We use AdamW with an initial learning rate of 3e-4 and cosine annealing to a minimum of 5e-6.

9.3. Choice of Text Encoder

We adopt MPNet [18] as our text encoder for part descriptions rather than SigLIP due to its superior structure-preserving properties for sentence-level embeddings. SigLIP, optimized for short image captions (e.g., "A photo of a dog"), exhibits pathologically high cosine similarities across semantically distinct part descriptions generated by Gemini, undermining the discriminative structure necessary for partlet-based learning. For instance, MPNet correctly assigns high similarities (>0.8) to functionally equivalent parts across classes - such as wheels (airplane, car, bicycle, wheelchair), doors (airplane, car), and handles (scooter, bicycle, wheelchair), while maintaining low similarities (<0.3) between parts with different affordances, such as tires vs. doors/windows or pedals vs. airplane components. In contrast, SigLIP assigns uniformly high similarities to both sets, collapsing the semantic space and preventing our partlets from learning meaningful text-conditioned part alignment during training. We could have used SigLIP on just the part labels instead of the affordance descriptions, and SigLIP would be an appropriate encoder in that setting. However, with such a design choice, we would not have been able to disambiguate between the same part labels across different object categories (e.g., wheels of a *wheelchair* vs wheels of an *airplane*).

Why affordance descriptions? A key motivation for incorporating affordance information into part annotations is rooted in the cognitive science understanding that humans interpret and define object parts not just by geometry, but by their function, context, and description. Short or generic part names (e.g., “leg”, “handle”) are often ambiguous across different objects, lacking any semantic detail regarding the role or meaning of a part within a specific context. For example, “legs” fulfill distinct structural functions and take on different forms for chairs, tables, or sofas, a distinction that arises from their object-specific affordances. Prior work shows that such affordance-based cues and descriptive information guide human part recognition and reduce label ambiguity, supporting more robust reasoning and communication. Thus, by situating part annotations within functional and contextual descriptions, our approach enables higher-quality, less ambiguous labeling, consistent with cognitive models of human object understanding.

9.4. Vocabulary Compression using MPNet and Gemini-2.5-Flash

Our vocabulary compression system employs a two-stage pipeline that combines MPNet embeddings for candidate generation and Gemini LLM verification to identify and merge duplicate classes and parts across a 3D object taxonomy. For confirmed matches, the system successfully identifies semantically equivalent entities with high MPNet similarity scores that Gemini validates as identical: for example, "laptop_computer" and "laptop" (similarity: 0.944) are merged because Gemini recognizes that "both candidate names refer to the exact same physical device... consistently define it as a portable personal computer designed for mobile use," while within the "microwave_oven" class, "door_glass" and "glass" (similarity: 0.865) are unified because Gemini concludes "both descriptions refer to the transparent panel integrated into the door... that allows viewing food and contains radiation. The secondary part name ’glass’ is a concise reference to the "door_glass". Similarly, "bed_footboard" and "footboard" (similarity: 0.953) are merged as Gemini states, "bed_footboard" is a more explicit naming of ’footboard’, and their descriptions are semantically identical, describing a panel at the foot of the bed opposite the headboard." For rejected pairs, the system correctly distinguishes semantically distinct parts despite high embed-

ding similarity: "car_front_bumper" and "car_rear_bumper" (similarity: 0.879) are kept separate because Gemini determines "while both parts are bumpers with the same protective function, their specified locations (front vs. rear) make them distinct semantic parts for a 3D car object," and within the "chair" class, "back_frame_horizontal_rod" and "back_frame_vertical_rod" (similarity: 0.943) remain separate because Gemini explains "the parts are distinct based on their orientation within the back frame: one is explicitly described as a 'horizontal rod' providing reinforcement for the backrest, while the other is a 'vertical rod' providing structural support." The compressed vocabulary output maintains canonical names (choosing more verbose/descriptive variants), aggregates part counts across merged entities, and produces a mapping log that records every alias resolution for downstream lookup when legacy names are encountered during inference.

10. Metrics for Semantic 3D Part Segmentation

We evaluate our method using two complementary metrics that progressively incorporate semantic label correctness. (1) Class-agnostic mIoU: Following prior work, for each ground-truth part, we compute the maximum IoU across all predicted segments and average these values, ignoring semantic labels entirely - this captures pure geometric segmentation quality. (2) Label-Aware mIoU (LA-mIoU): For each ground-truth part, we identify the predicted segment with the highest geometric overlap (as in class-agnostic mIoU), then assign credit only if its semantic label exactly matches the ground truth; otherwise, the part contributes 0.0 - this measures joint geometry-semantic accuracy with strict label matching. The gap between class-agnostic mIoU and LA-mIoU reveals semantic prediction errors.

11. Inference-time Ablations

We evaluate two additional inference modes of our ALIGN-Parts model, extending beyond the primary dynamic part activation approach to better understand the contributions of part cardinality and label information in our segmentation pipeline. The first alternative mode, which we term the *clustering+part number* setting, completely forgoes the use of any part vocabularies or text labels during inference. Instead, it relies solely on the fused geometric and appearance features output by the model, upon which we run k-means clustering to produce purely class-agnostic instance clusters. This setup rigorously probes the ability of the learned feature embeddings, untethered to semantic labels, to support coherent part decompositions across diverse objects, essentially isolating the impact of visual and geometric cues alone. The second mode, called *+Part number*, examines whether providing the model with the exact ground-truth part count for each input shape improves segmentation quality compared to the default setting, where the model dynamically infers the number of parts to activate. After producing all candidate partlet masks and calculating their partness scores, this mode ranks the partlets by a saliency score, which is a composite measure combining the confidence that a partlet corresponds to an actual part (i.e., partness) and the average mask coverage over the point cloud (mean mask probability mass over points). From this ranking, the top M partlets are retained, where M is the true number of parts for the target shape, and every point in the shape is assigned the best matching mask among these selected partlets to yield a hard K-way partition. These inference ablation modes and their quantitative outcomes are detailed in Tab 4, demonstrating that ALIGN-Parts is able to robustly estimate accurate part cardinality and segmentation even without explicit part label or count guidance, and that the fused multimodal features alone provide meaningful cues towards coherent part delineation.

12. Experiments and Analysis

Given the challenges inherent in semantic 3D part segmentation, we find that no current published work is directly comparable to our method. To enable rigorous evaluation, we introduce our own strong baseline detailed in Sec. 12.1. While we do include comparisons against class-agnostic 3D part segmentation methods in this manuscript, it is important to note that these do not constitute an entirely fair benchmark for our approach. Most prior methods have been trained using proprietary, closed-source Objaverse-scale datasets, with specific data details and part annotations rarely disclosed publicly.

In contrast, our experiments are conducted on fully open, publicly available datasets, and our methodology itself improves upon these resources, making our results more easily reproducible and comparable for future researchers. Furthermore, a key emphasis of our approach is efficiency: we process only 10,000 input points per shape, in stark contrast to the 100,000 points typically used by class-agnostic segmentation baselines. This restriction stems from the academic compute limitations we faced, while prior works often benefit from corporate-scale GPU resources.

Despite these constraints, our method achieves competitive or superior performance relative to existing baselines. It is reasonable to expect that - if provided with similar data volume and computational resources - ALIGN-Parts would further extend its advantage on standard metrics and benchmarks. Our design choices thus not only democratize research in 3D part segmentation but also highlight the promise of reproducibility, accessibility, and efficiency for large-scale semantic understanding in open 3D datasets.

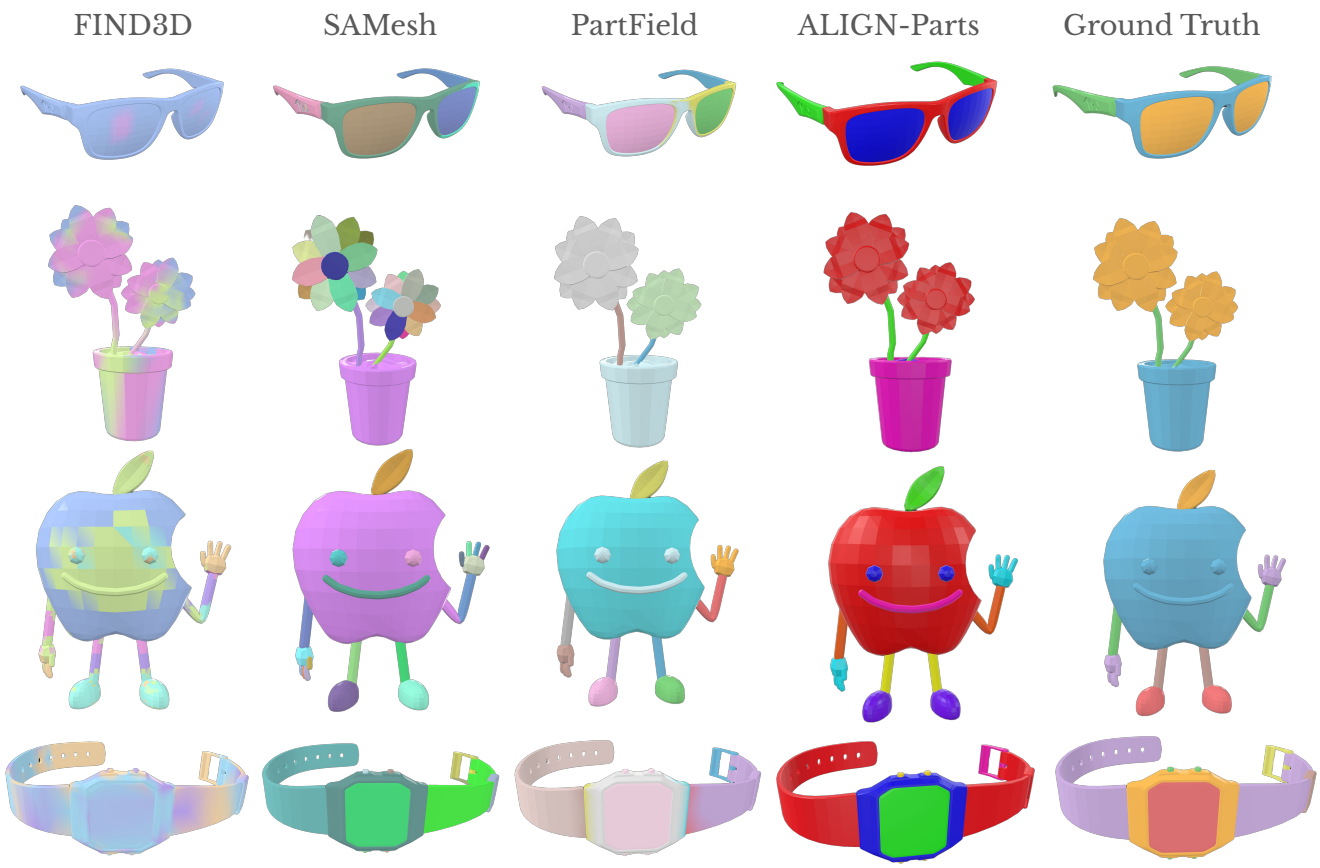

Figure 7. **Qualitative Results on PartObjaverse-Tiny.** Qualitative comparisons for class-agnostic part segmentation on unseen objects from PartObjaverse-Tiny. For our model (ALIGN-Parts), this scenario is a true out-of-distribution (OOD) challenge: nearly all part categories are novel relative to training. Notably, our core objective is *semantic 3D part segmentation*, and evaluation on class-agnostic segmentation is provided mainly for completeness and comparability to previous works. As prior methods do not release their training data or part annotations, we cannot train our model for a fairer setup. Despite being trained on less data and with only $\frac{1}{10}$th the input points, our method generalizes robustly in OOD settings.

Table 4. Evaluation of different inference modes of ALIGN-Parts on our test set, using mean IoU (mIoU) and label-aware mIoU (LA-mIoU). Providing additional ground-truth part count information only slightly improves the model's performance, showing that ALIGN-Parts often estimates accurate part cardinality based on just the input geometric and appearance features of a 3D shape.

Variant	3DCoMPaT (126)		Find3D (8)		PartNet (72)		Average	
	mIoU↑	LA-mIoU↑	mIoU↑	LA-mIoU↑	mIoU↑	LA-mIoU↑	mIoU↑	LA-mIoU↑
Clustering + Part Number	0.370	n/a	0.528	n/a	0.537	n/a	0.478	n/a
+Part Number	0.452	0.268	0.625	0.138	0.757	0.559	0.611	0.322
No Part Number	0.453	0.268	0.595	0.133	0.753	0.546	0.600	0.316

12.1. PartField+MPNet baseline

Given that our task of semantic part segmentation (in contrast to the relatively easier and more prominent class agnostic part segmentation), we create our own baseline - PartField + MPNet, which assigns labels to parts obtained by KMeans clustering on per-point features. We experimented with two variants of this model, in terms of input features: PartField and PartField + DINOV2, and found that the latter usually yields much better performance. So, without loss of generality, our baseline PartField + MPNet refers to the model where we have per-part PartField + DINOv2 features fused through cross-attention. Specifically, we employ cross-attention fusion to combine per-part geometric (448-D) and appearance (768-D) features, projecting them through 512-D hidden layers into a shared 256-D latent space. The architecture consists of a dense feature fuser (2.8M parameters) with 4

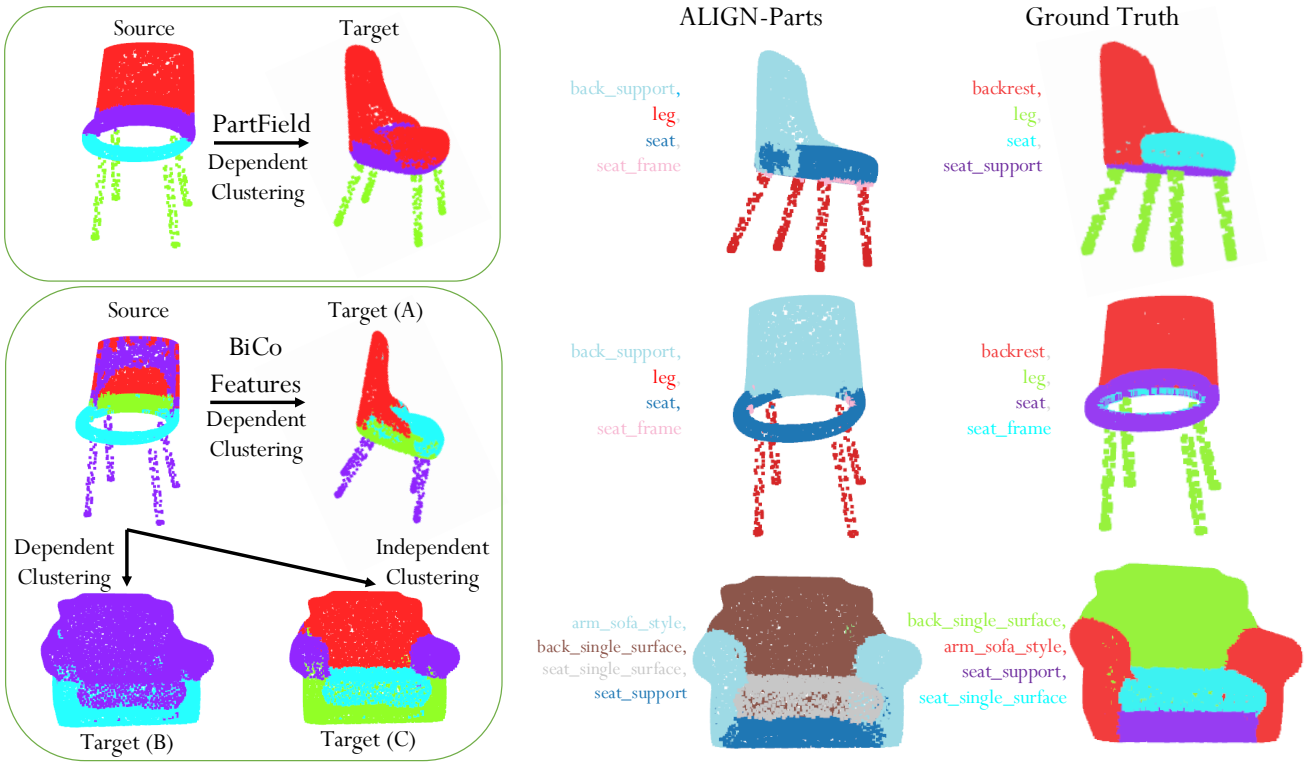

Figure 8. **3D Shape Co-Segmentation Analysis.** *Left: clustering-based co-segmentation.* Prior methods such as PartField perform *dependent clustering* by first segmenting a source shape via feature clustering and then using the resulting cluster means to initialize K-means on a target shape, implicitly enforcing part correspondence; this can break when the target has a different part count or geometry, causing errors such as the red backrest region bleeding into the seat on the target chair. Using the same dependent co-segmentation strategy with our fused BiCo features (**BiCo Feature Dependent Clustering**) yields improved transfers on moderately similar targets (Target A), but performance degrades on more challenging targets with greater variation in part structure (Target B). As an alternative, we apply **independent clustering** to Target C, where the target is segmented with source initialization and clusters are matched post hoc by comparing source and target cluster centers, which proves more reliable for difficult co-segmentation cases. *Middle and right: feedforward ALIGN-Parts.* In contrast to all clustering-based variants, the proposed feedforward ALIGN-Parts model (middle) directly predicts part segmentation and names, achieving robust results across shapes with differing part counts and topologies, and eliminating any dependence on source shapes or explicit co-segmentation.

attention heads operating at 512-D, followed by dedicated MLP projectors for local part features (0.39M), semantic text embeddings (0.52M), and global shape descriptors (0.75M), totaling approximately 5.1M parameters. Training optimizes three objectives: symmetric InfoNCE loss for local part-text alignment, a global-level contrastive loss between shape and class embeddings, and a cross-entropy clustering loss that predicts part counts with equal weighting (λ=1.0) across all terms. The model is trained for 100 epochs using AdamW with learning rate 3e-4, weight decay 1e-5, and cosine annealing schedule ($\eta_{\min}$=5e-6) with batch size 64. The part count prediction head (0.63M parameters) uses a two-layer MLP with GELU activation to classify the number of semantic parts from fused global features. All projectors and attention mechanisms utilize dropout regularization (p = 0.1) to prevent overfitting during training. During inference, PartField + MPNet first predicts object category by comparing the projected global feature against all class embeddings, then performs soft k-means clustering (k from the part count head) on fused point-level features with Hungarian matching to assign semantic labels by computing cosine similarity between projected cluster centroids and MPNet embeddings of candidate part names.

12.2. Results on PartObjaverse-Tiny

The evaluation of our model (ALIGN-Parts) on PartObjaverse-Tiny represents a rigorous out-of-distribution (OOD) benchmark for semantic and class-agnostic part segmentation. Unlike prior works that are often trained on millions of Objaverse data points with extensive part label coverage - frequently leveraging closed-source annotations and proprietary splits - ALIGN-Parts is developed with

a substantially smaller, open dataset and has not been exposed to the vast majority of categories or parts in this test scenario. In practice, nearly all part categories in PartObjaverse-Tiny are novel for our model, making this an especially challenging transfer and open-vocabulary generalization task.

It is essential to note that our core objective is semantic 3D part segmentation, and our evaluation on class-agnostic part segmentation in this benchmark is performed chiefly for completeness and comparability with existing methods. Due to the lack of public access to the training data and part annotations used by previous approaches, our model cannot be trained under exactly the same conditions for a perfectly fair comparison. Moreover, most baseline methods operate on $\sim$100,000 input points per shape and benefit from significant computational resources, while ALIGN-Parts utilizes only $10{,}000$ points per shape due to academic hardware constraints - yet still achieves competitive performance.

Despite these pronounced limitations in data volume and compute, our model generalizes robustly in the OOD setting provided by PartObjaverse-Tiny. As an additional step to mitigate the domain gap, we fine-tune ALIGN-Parts on roughly 100 objects sampled from Objaverse. We present qualitative visualizations in Figure 7, showing strong segmentation quality and consistency across completely novel categories, highlighting the adaptability and efficiency of our pipeline for open-world 3D part segmentation.

12.3. Part-Retrieval Comparison with Find3D

Beyond semantic segmentation, ALIGN-Parts also supports text-driven part retrieval—the task of localizing and retrieving point cloud regions corresponding to natural language part queries. This capability, introduced by Find3D, enables flexible, open-vocabulary part discovery directly from unstructured text descriptions. Our approach performs retrieval by constraining the candidate label vocabulary to only those parts known to be present in the target object class, rather than the full semantic vocabulary. Additionally, we set the number of active partlets to match the ground-truth part count for the object, which serves as an oracle constraint. While this restriction reduces search space and assignment ambiguity - allowing the model to match predicted part slots to a small, object-specific set of valid labels rather than choosing from dozens of candidates - it also enables fairer, more interpretable comparisons.

This constrained retrieval setup typically yields higher segmentation accuracy by minimizing false positive label assignments and focusing the model's attention on semantically coherent parts. The key advantage of our approach lies in the compositional three-level hierarchy: point cloud $\rightarrow$ partlet $\rightarrow$ part label. This formulation naturally encourages the discovery of connected point groups with consistent semantic meaning, whereas alternatives may suffer from fragmentation or over-segmentation.

We present qualitative comparisons with Find3D on two representative 3D objects from the *airplane* and *motorbike* object classes in the Objaverse-General benchmark (part of our closed-vocabulary evaluation set), released by the Find3D authors. As shown in Figure 9, ALIGN-Parts consistently retrieves more spatially coherent and semantically meaningful part groups, demonstrating the effectiveness of our partlet-based design for part localization and retrieval tasks.

12.4. 3D Shape Co-Segmentation and Part Label Transfer

Figure 8 shows results and analysis of 3D Shape Co-Segmentation using ALIGN-Parts (and the BiCo features) as compared to PartField. A classical approach to 3D part segmentation operates in a co-segmentation setting, where multiple shapes from the same category are jointly analyzed to establish consistent part correspondence. Prior methods, including PartField, employ what we call *dependent clustering*: they first segment a source shape using feature clustering, then initialize k-means clustering on a target shape using the source cluster centroids, implicitly enforcing part correspondence. While this strategy can succeed on geometrically similar shapes, it proves fragile when target shapes exhibit different part counts or topologies. For example, in Figure 8, dependent clustering on a moderately similar target chair (Target A) produces reasonable results, but fails dramatically on targets with substantial part variation (Target B), causing geometric boundaries to blur (e.g., the backrest merging incorrectly with the seat).

An alternative, *independent clustering* approach segments each target shape autonomously and then matches clusters post hoc by comparing source and target cluster centers. As shown in Target C, this mode is more robust to topological differences, though it forgoes any direct geometric correspondence to the source.

In contrast to both clustering-based paradigms, our proposed ALIGN-Parts adopts a fully *feedforward*, discriminative approach that predicts part segmentation masks and semantic labels jointly, without requiring source shape initialization or explicit co-segmentation. This design eliminates brittleness to part count variation and geometric mismatches, enabling robust generalization across shapes with diverse part structures and semantics. As demonstrated in Figure 8 (middle and right panels), ALIGN-Parts consistently produces accurate, semantically grounded part segmentations regardless of target shape complexity.

13. TexParts Dataset

A central aim of our approach is to enable the construction of a high-quality 3D part annotation dataset with minimal manual intervention, ensuring both unified and comprehen-

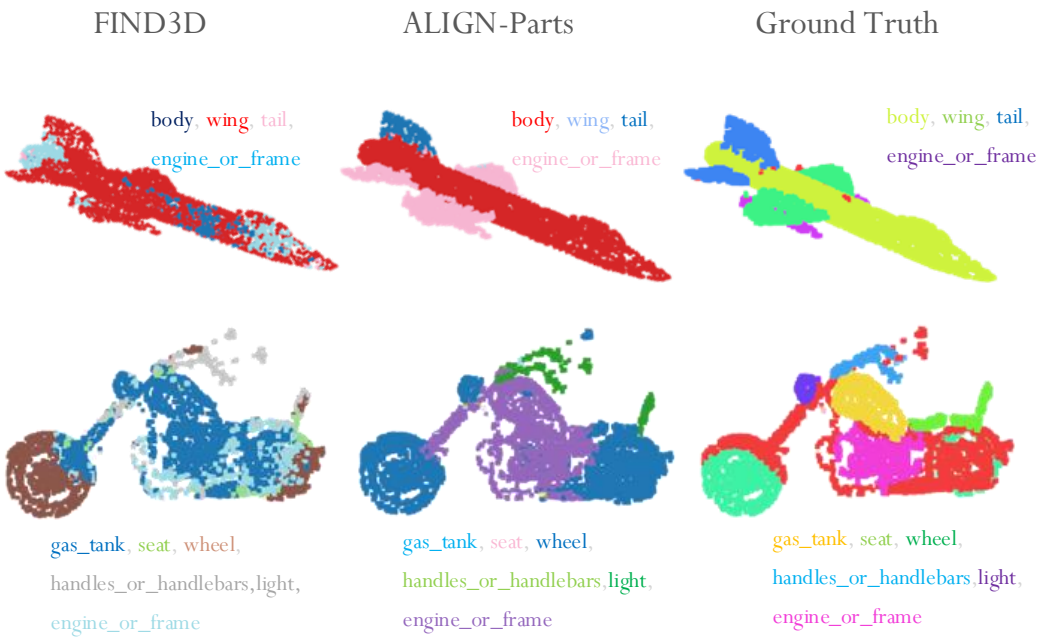

Figure 9. **Part Retrieval Comparison with Find3D.** We demonstrate text-driven part retrieval on two representative objects (airplane and motorbike) from Objaverse-General. Given natural language part queries (e.g., "body", "wing", "gas tank", "wheel"), ALIGN-Parts identifies and retrieves spatially coherent point groups corresponding to each part. Compared to Find3D (left), our method produces more semantically and spatially consistent part retrievals by leveraging the hierarchical point → partlet → part label decomposition. This design encourages the discovery of well-connected, semantically meaningful regions rather than fragmented point clusters. Ground truth part segmentations (right) show the target labels. ALIGN-Parts achieves results that closely align with ground truth, validating the effectiveness of our partlet-based formulation for open-vocabulary part localization and retrieval.

sive part labeling at scale. For this purpose, we select the TexVerse dataset as our unannotated 3D source corpus, leveraging its exceptional quality, high-resolution textures, and extensive diversity of 3D assets [26]. TexVerse consists of over 850,000 unique 3D models with physically based rendering (PBR) materials and rich metadata, making it an ideal foundation for large-scale part segmentation.

Our pipeline begins with the automated filtering of TexVerse models: using Gemini-Flash LLM, we combine thumbnail images and other metadata to preselect high-quality objects and exclude inadequate or malformed models. Next, we apply our ALIGN-Parts model and save, for each shape, its predicted part masks, part names, and both semantic and segmentation confidence scores. To prioritize downstream annotation effort, we sort objects by their average confidence score (in descending order) so that annotators see the most reliable candidates first. Selected objects are then routed to human annotators for validation and correction.

During annotation, the annotators operate with several aids: a part name prompting tool for searching or extending the active part vocabulary, and (optionally) the ability to reference unlabeled geometric part masks produced by PartField. Our annotation process is explicitly bilevel - phase one focuses on validating and making minor edits to ALIGN-Parts predictions, while phase two addresses new or missing parts requiring more substantive manual annotation. By the time of submitting this work, the first phase has covered approximately 8,000 objects, comprising around 14,000 unique part categories. Examples from the dataset are shown in Figure 10.

A key observation from our annotation workflow is the dramatic reduction in manual effort enabled by our methodology: annotating 3D objects from scratch typically takes anywhere from 15 to 25 minutes per shape, while our model-assisted pipeline reduces annotation time to just 3 to 5 minutes on average - a time saving of approximately 5–8× without sacrificing annotation quality.

Importantly, and in clear contrast to recent approaches that keep their Objaverse-derived part annotations closed-source, we commit to releasing TexParts as a public resource upon publication, with the aim of advancing large-scale open research in semantic 3D part understanding.

14. Limitations and Future Work

Key limitations are: noisy real-world scans challenge our manifold assumptions, Mahalanobis confidence degrades under distribution shift, and open-vocabulary generalization is

Figure 10. **TexParts Dataset.** We demonstrate human-in-the-loop annotation of Texverse [26] using ALIGN-Parts, enabling scalable dense 3D part segmentation.

limited to categories similar to the training data. Future work should extend this to articulated objects and integrate part-level alignments into foundation 3D models for manipulation and generation.

More broadly, the primary limitations of our work stem from the relatively restricted set of objects and parts on which ALIGN-Parts has been trained, compared to the vast (though finite) variety of parts that occur in the real world. This gap is largely due to the scarcity of large-scale 3D datasets with dense part annotations and a unified, operational definition of what constitutes a "part". In effect, this creates a chicken-and-egg problem: ALIGN-Parts was designed to enable robust 3D part annotation at scale, yet the robustness and coverage of the model itself are constrained by the limited annotated data available for training.

Future work will focus on mitigating this dependency by exploring self-supervised or weakly supervised formulations and by incorporating stronger 3D priors, for example from generative models or skeletal/medial representations. Another important direction is to reduce the current reliance on frozen PartField features by enabling full end-to-end training of the geometric feature extractor, which was not pursued here primarily due to computational constraints rather than methodological ones. Despite these limitations, our framework is immediately usable by organizations and research labs with abundant compute and proprietary 3D assets, who can scale ALIGN-Parts to richer, closed-source datasets and drive progress towards truly large-scale 3D scene understanding at the part level.