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

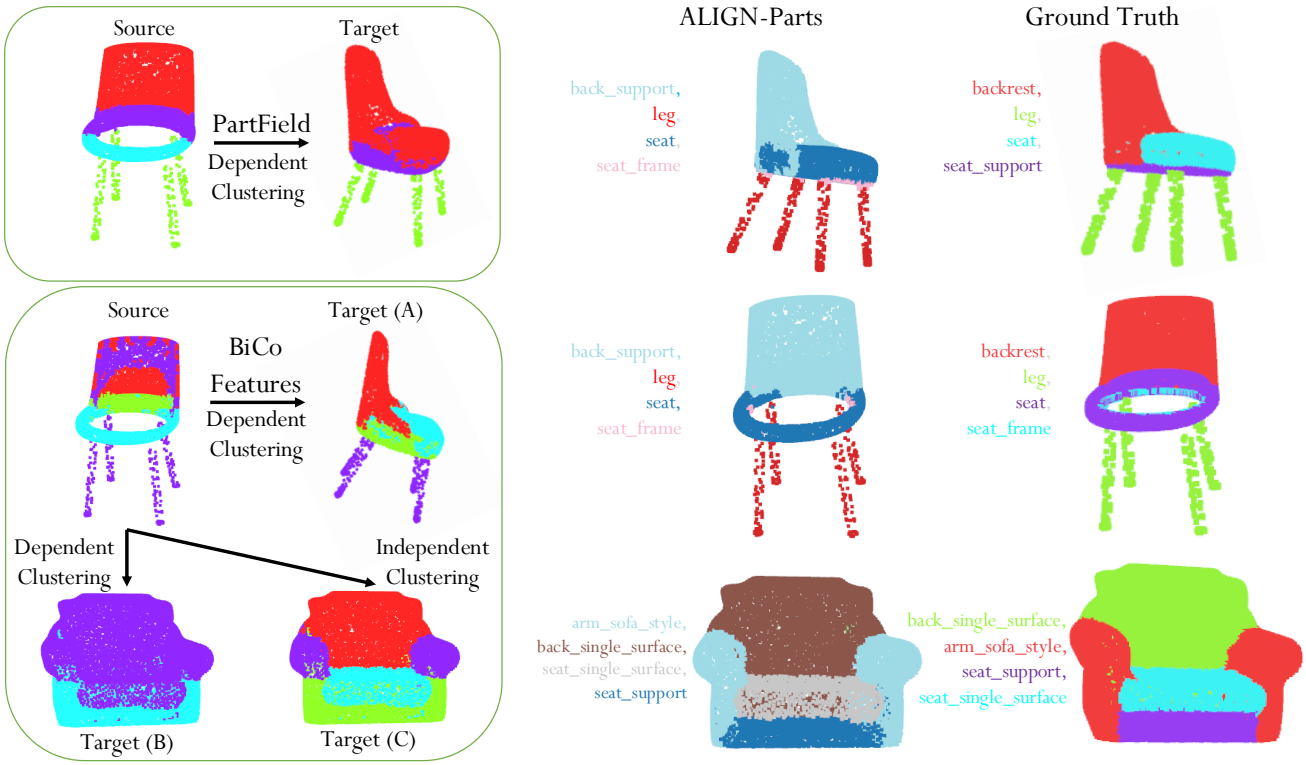

Figure 8. **3D Shape Co-Segmentation Analysis.** *Left: clustering-based co-segmentation.* Prior methods such as PartField perform *dependent clustering* by first segmenting a source shape via feature clustering and then using the resulting cluster means to initialize K-means on a target shape, implicitly 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 (η_{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-

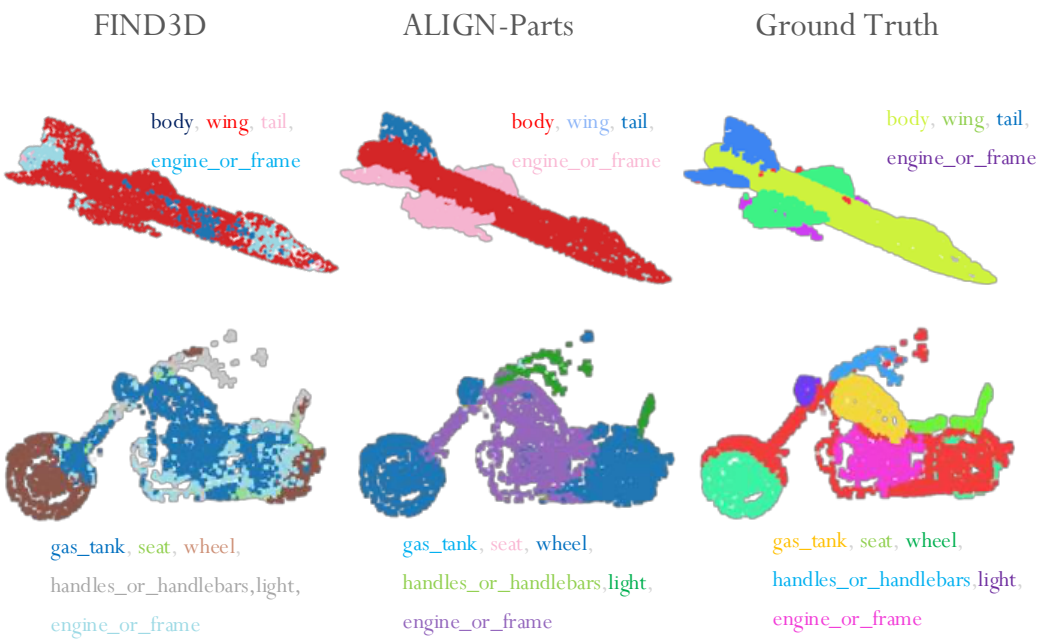

Figure 9. **Part Retrieval Comparison with Find3D.** We demonstrate text-driven part retrieval on two representative objects (airplane and motorbike) from Objaverse-General. Given natural language part queries (e.g., "body", "wing", "gas tank", "wheel"), ALIGN-Parts identifies and retrieves spatially 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.