Attribute-Guided and Hybrid Approaches for Interpretable 3D Object Retrieval

Juwon Lee
Content Research Division
ETRI
Daejeon, Rep. of Korea
zacurr@etri.re.kr

Suwoong Lee
Content Research Division
ETRI
Daejeon, Rep. of Korea
suwoong@etri.re.kr

Seungjae Lee
Content Research Division
ETRI
Daejeon, Rep. of Korea
seungjlee@etri.re.kr

***Abstract*—We propose a 3D content retrieval framework that enables interpretable, efficient, and user-controllable search. Existing methods often rely on latent embeddings, which lack semantic transparency. In contrast, our approach extracts explicit semantic attributes, such as shape, material, and style, from 3D object descriptions. We develop two retrieval prototypes: (1) an attribute-only system using interpretable keywords and (2) a hybrid system combining feature similarity and attribute-based filtering. Using a testbed of 3D objects from Objaverse dataset and ULIP-Objaverse-Triplets resource, we demonstrate improved retrieval control and alignment with user intent.**

***Index Terms*—3D object retrieval, attribute extraction, retrieval by attribute**

I. Introduction

The success of CLIP [1] has advanced vision–language understanding by enabling flexible multimodal tasks and inspiring models like DALL·E, Stable Diffusion, and large multimodal models (LMMs). Motivated by this, large-scale 3D datasets such as Objaverse [2] and Objaverse-XL [3] have emerged to support similar advancements in the 3D domain.

Building on these datasets, models like ULIP [4] and OpenShape [5] learn joint embeddings across 3D objects, images, and text, enabling multimodal tasks like 3D retrieval and classification. However, most training captions are templated from sparse metadata (e.g., *a picture of [object]* or *a point cloud model of [object]*), limiting their semantic richness and weakening text alignment in downstream tasks.

To address this, ULIP-2 [6] uses a fully automated pipeline to render 3D models into multi-view images and generate richer captions using vision–language models like BLIP-2 [7]. This improves the quality of textual supervision and broadens usable training data.

In 3D content retrieval, textual descriptions are embedded into a joint feature space alongside visual and 3D representations. However, rather than directly leveraging the textual information, the retrieval process relies solely on similarity within the embedding space. This indirect use of text can result in suboptimal results, often retrieving content that is semantically misaligned with the user's intent due to weak alignment between the text and other modalities. When relevant results are not returned, users are often forced to reformulate their queries—whether in the form of text, images, or 3D models—multiple times, leading to repeated feature extraction and similarity computation. This process significantly increases overall retrieval time and computational cost.

In this paper, we propose a new retrieval framework that pre-extracts semantic attributes—such as shape, material, and components—from 3D object descriptions and uses them directly during retrieval. This approach improves precision, transparency, and efficiency by enabling attribute-guided filtering and interpretable interaction.

II. Dataset Analysis and Preprocessing

We use Objaverse [2] as the source of 3D models for our experiments. Although it includes over 800K assets with associated metadata, the quality and consistency of this metadata vary widely, limiting its direct usability for semantic retrieval.

A. Metadata Limitations

We focus on four text-based metadata fields relevant to attribute extraction: name, categories, tags, and description. Although intended to convey semantic information, many entries are missing or contain non-informative content, reducing their utility for downstream tasks.

- Name: While some names are meaningful (e.g., 'wooden chair'), many are placeholders or arbitrary strings (e.g., 'test1', '1234').
- Categories: Based on Sketchfab's 18-class taxonomy, but over 52% of objects lack this data, and many are assigned multiple overlapping categories.
- Tags: The vocabulary is large (167K+ unique tags) but noisy; many are tool-related (e.g., 'blender', '3d') and over a third of objects have no tags.
- Description: Often missing or filled with irrelevant technical notes (e.g., 'Prob Threshold: 0.5') or meaningless sequences (e.g., '12562').

Due to their sparsity, noise, and inconsistency, these fields are ill-suited for direct use in retrieval tasks—underscoring the need for automated and scalable attribute extraction.

B. Testbed Construction for Attribute Extraction

To support attribute-level 3D retrieval, we construct a structured testbed by organizing the objects according to Sketchfab's 18-category taxonomy. This enables category-aware retrieval and facilitates the extraction of representative attributes

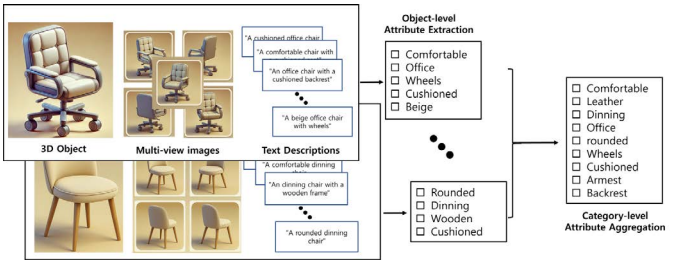


Fig. 1. Pipeline for object-level attribute extraction and category-level aggregation

for each semantic class. Compared to object-level attributes, category-level aggregation produces more robust profiles by reducing noise and capturing generalizable patterns. These profiles serve as interpretable targets for semantic retrieval.

To provide the necessary visual and textual resources for attribute extraction, we incorporate the ULIP-Objaverse Triplets dataset [6] released by ULIP-2, which aligns each 3D object with multiple rendered views and corresponding natural language captions. This multimodal resource complements the native metadata and serves as the primary source for extracting object attributes.

Each object in the triplets dataset is represented by:

- a 3D point cloud (available at multiple sampling densities: 2K, 8K, 10K),
- 12 rendered images from uniformly spaced viewpoints (30° intervals), totaling 9.58M images,
- and 10 generated descriptions per image using BLIP-2-opt6.7B, resulting in over 95.8M caption candidates.

We assess the suitability of the generated descriptions for attribute extraction. Due to uniform spherical rendering, key features are occasionally occluded, resulting in inaccurate or generic captions. Moreover, a domain gap between real images (used to train BLIP-2) and synthetic Objaverse renderings further degrades quality. To address this, we follow [6] and apply CLIP-ViT-Large [1] to select the most relevant caption per view based on image–text similarity.

While imperfect, the filtered descriptions offer significantly richer semantic content than the original metadata, capturing important object characteristics like shape, material, and structure. Our final testbed comprises 3D objects in 18 categories, each with 12 rendered images and 12 high-confidence captions (one per view), supporting our attribute extraction and retrieval experiments.

III. Proposed System

Our system enhances 3D content retrieval by pre-extracting interpretable attributes from natural language descriptions of 3D objects. Unlike conventional embedding-only methods, it enables the retrieval to be guided by explicit attribute keywords. The framework consists of three stages: (1) object-level attribute extraction, (2) category-level attribute aggregation, and (3) attribute-based retrieval, with the first two stages illustrated in Fig. 1.

Fig. 2. Category-level attributes for the *fashion-style* category. (a) Raw frequency-based aggregation of object-level general attributes. (b) Refined general attributes and (c) refined adjective attributes, visualized as word clouds after removing non-informative and generic terms.

A. Object-Level Attribute Extraction

We extract attributes from the 12 natural language descriptions per object, which are generated from multi-view renderings. After evaluating several NLP libraries (RAKE, YAKE, Textacy, and spaCy), we adopt spaCy [8] for its robust part-of-speech (POS) tagging, enabling fine-grained control over the types of attributes retrieved.

We identify two complementary types of attributes:

- General attributes (e.g., ‘wheels’, ‘wood’, ‘backrest’) - parts or materials of objects, primarily extracted by noun phrases.
- Adjective attributes (e.g., ‘cushioned’, ‘rounded’, ‘comfortable’) - perceptual or stylistic descriptors.

Each description is syntactically parsed to extract relevant patterns (for example, adjective-noun pairs), resulting in a candidate set of attributes per object. To ensure consistency, all attributes are lowercased, deduplicated, and normalized in all views.

For general attributes, we apply lemmatization and remove articles (e.g., ‘a’, ‘an’, ‘the’) to unify variants like ‘a shoe’, ‘the shoe’ and ‘shoes’ into the canonical form ‘shoe’. This normalization mitigates duplication due to morphological variation. In contrast, lemmatization is avoided for adjective attributes, where semantic drift can occur, for example, ‘left’ becoming ‘leave’ or ‘colored’ reduced to ‘color’. Thus, adjective attributes are preserved in their original surface form to maintain semantic fidelity.

The resulting structured and normalized attribute set forms a robust foundation for category-level aggregation and attribute-based retrieval in subsequent stages.

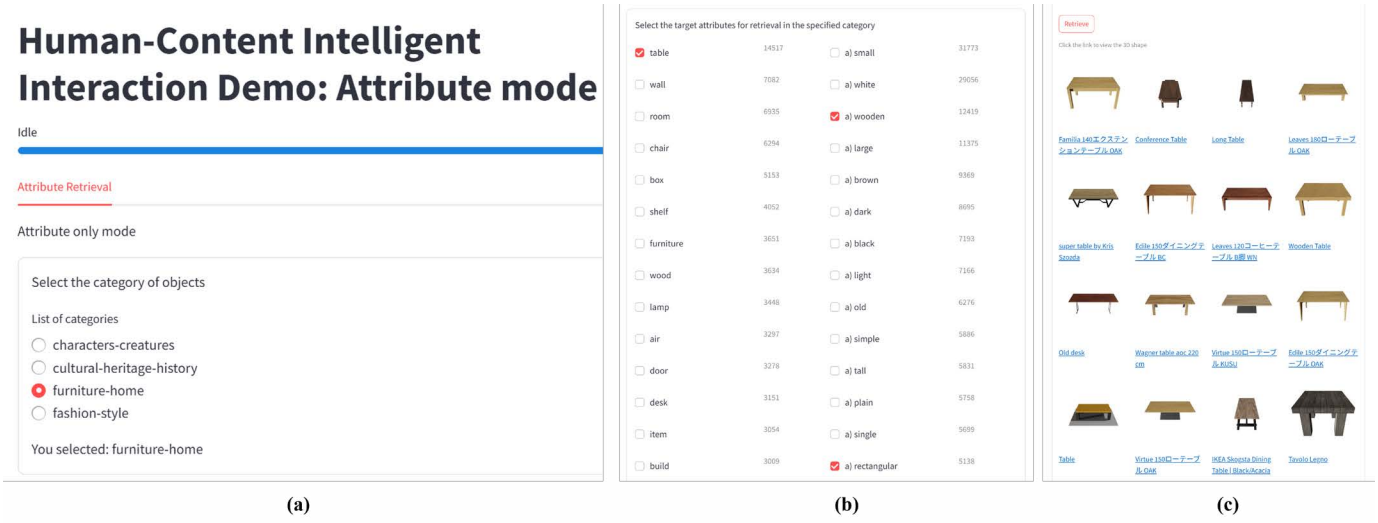


Fig. 3. Attribute-based retrieval interface and example results. (a) The user selects a category (e.g., *furniture-home*). (b) Based on the selected category, a curated list of general and adjective attributes is displayed; the user selects 'table' (object type), 'wooden' (material), and 'rectangular' (shape). (c) The system retrieves and ranks 3D objects that match the selected attributes.

B. Category-Level Attribute Aggregation

To build interpretable and representative profiles for each category, we aggregate object-level attributes across the 18 Sketchfab taxonomy classes. For each category, we collect attributes from its constituent objects and compute their frequencies, as shown in Fig. 2(a).

This reveals both informative and uninformative patterns. While attributes like 'shoe', 'ring', and 'woman' meaningfully represent the *fashion-style* category, we also observe frequent generic or dataset-specific terms such as '3d model', 'object', or 'digital', which provide little discriminative value. Additionally, rendering biases introduce visual context terms like 'gray background' or 'white object', unrelated to object semantics.

To address this, we apply filtering to remove generic or non-informative terms. The resulting cleaned attribute profiles better reflect category-specific characteristics and improve semantic clarity. Fig. 2(b,c) shows refined general and adjective attributes for the *fashion-style* category, visualized as word clouds.

These category-level profiles are more robust than raw object-level attributes, offering a generalizable and interpretable foundation for downstream attribute-based retrieval.

C. Attribute-Based Retrieval

This stage enables interpretable and user-guided 3D content retrieval using semantic attributes. Unlike conventional methods that rely solely on latent embeddings, our approach uses explicit attributes that were extracted and aggregated in the previous stages. We implement two retrieval prototypes: (1) an attribute-only interface, and (2) a hybrid system combining embedding similarity with attribute constraints.

Both share a common matching logic. For each object, we reference the object-level attributes extracted in Section III-A to determine relevance. A match is declared if the object contains attributes that satisfy the user-selected query terms. Based on empirical observations, we find that relying on adjective-only terms (e.g., 'round') often leads to ambiguous or irrelevant results. To improve precision, we define matching in two configurations:

- General-only match: where matching is based solely on general (noun) attributes (e.g., 'table').
- Adjective + general match: where matching requires co-occurrence of adjective–noun pairs (e.g., 'round table').

1) Matching Score Metrics: Given a query attribute set Q and object attributes A_o, we define:

a) Attribute Match Count (AMC): The number of attributes in A_o that contain or match each selected query attribute as a substring. This count is computed separately for general and adjective–general combinations.

b) Attribute Match Ratio (AMR): It is defined as:

$$AMR = N_{A_o,m}/N_{A_o}, \tag{1}$$

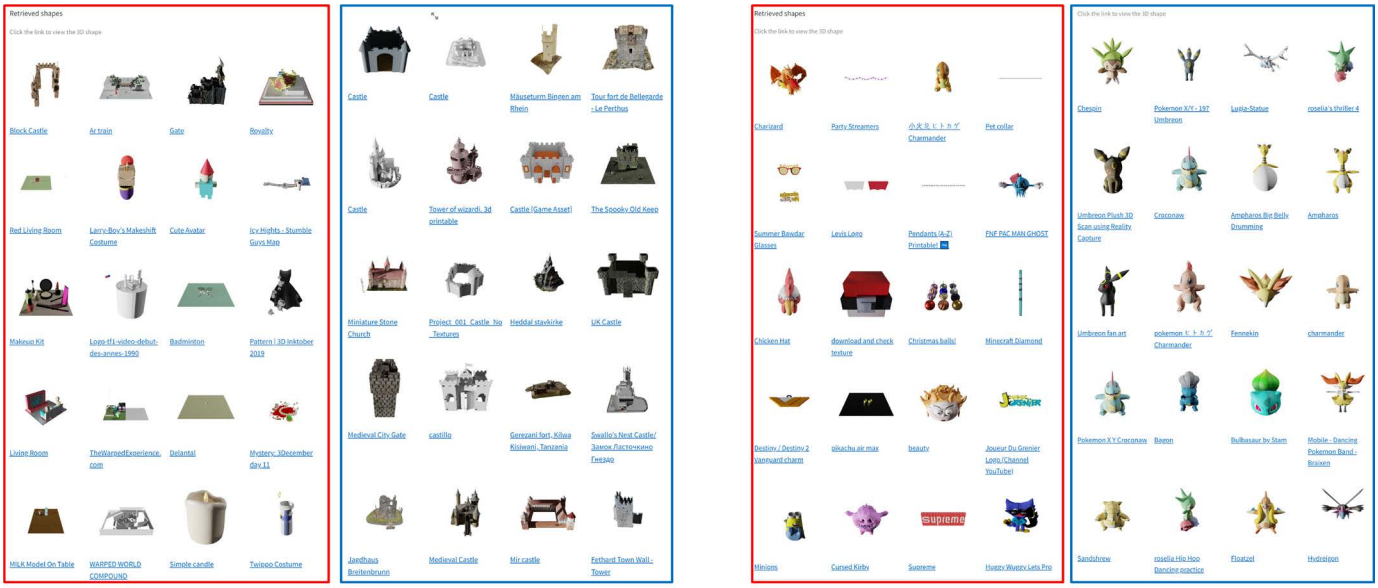
where N_{A_o} is the total number of attributes in A_o and $N_{A_o,m}$ is the number of matched attributes in A_o. This measures the proportion of the object's attributes that are relevant to the query.

c) Attribute Satisfaction Rate (ASR): This metric measures how completely the object satisfies the user's intent. It is defined as:

$$ASR = N_{Q,m}/N_Q, \tag{2}$$

where N_Q is the total number of Q and $N_{Q,m}$ is the number of matched attributes in Q. It reflects the proportion of user-selected attributes that are present in the object's attribute set. The value ranges from 0 to 1, corresponding to no $(0/N_Q)$ to full (N_Q/N_Q) satisfaction for a query with N_Q selected attributes.

Each metric is calculated for both general and adjective + general attribute types, resulting in six scores per object.



(a) **(b)**

Fig. 4. Comparison of retrieval results between feature embedding–based retrieval (left, red border) and attribute-only retrieval (right, blue border). (a) Results for the query 'castle'. (b) Results for the query 'pokemon'.

Ranking is determined via a priority cascade: first by ASR (adjective + general), then ASR (general), followed by AMC and AMR as tie-breakers.

This multi-metric strategy prioritizes semantically rich and specific matches (e.g., 'round wooden table') over generic or partial matches, improving both relevance and interoperability.

2) Attribute-Only Retrieval Prototype: This prototype enables category-aware, attribute-based retrieval without embeddings. Users select a category (e.g., *furniture-home*) and then choose attributes (e.g., 'wooden', 'rectangular', 'table') from a curated list derived from category-level profiles (Section III-B). The system compares each object's attributes (Section III-A) to the query and ranks results using the AMC/AMR/ASR metrics. As shown in Fig. 3, this allows intuitive, interpretable search by attribute combinations.

3) Feature–Attribute Hybrid Prototype: This hybrid system integrates visual–language embeddings with attribute filtering. Users provide a multimodal query (text, image, or 3D), which is encoded via a pretrained model (e.g., OpenShape [5] or ULIP [4]). Top-N results are retrieved by embedding similarity. Unlike the static lists in the attribute-only system, the hybrid interface dynamically extracts dominant attributes from the top-N results and surfaces them as filters. Users refine results by selecting preferred attributes, which triggers re-ranking using the same attribute match metrics (AMC, AMR, ASR). Final scores are computed by combining embedding similarity and attribute alignment.

This hybrid pipeline offers several advantages:

- It enables users to refine results without re-running expensive embedding computations or repeating feature similarity searches.
- It provides semantic interpretability through visible and controllable attribute filters.
- It supports exploratory browsing by surfacing meaningful attribute dimensions—such as material, shape, or style—that are often entangled or latent in embedding space but are clearly exposed through language-aligned attributes.

Together, these prototypes demonstrate flexibility across retrieval workflows—whether driven by interpretable attribute selection or guided by multimodal queries.

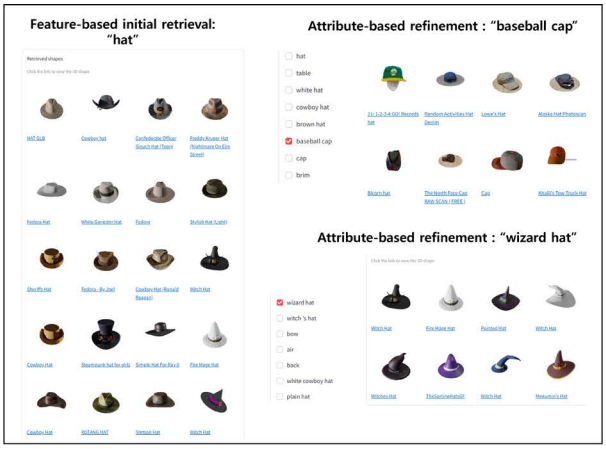
IV. Experimental Results

We qualitatively evaluate our retrieval framework using the structured testbed from Objaverse 1.0 (Section II-B), testing both the attribute-only and hybrid prototypes on selected categories. Due to the absence of ground-truth relevance labels, we assess results via visual comparison, focusing on semantic coherence and alignment with user intent.

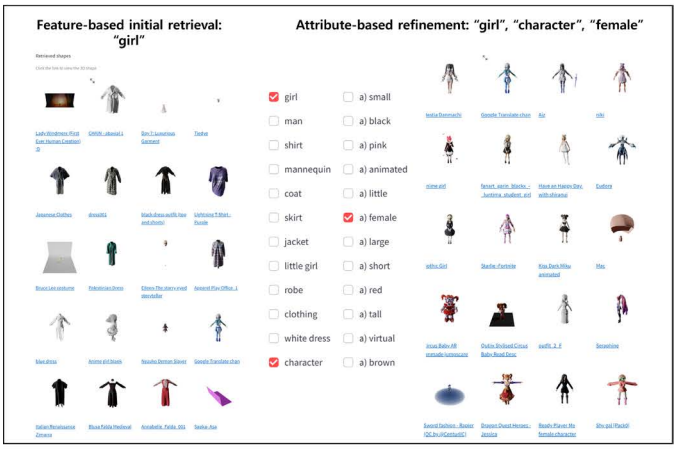
A. Attribute-Only Retrieval: Comparison with Feature-Based Methods

We begin by evaluating the attribute-only prototype, where users issue queries by selecting semantic attributes from curated category-level sets—without providing any visual or textual embeddings. We compare the results against OpenShape's embedding-based pipeline [5], which encodes natural language queries (e.g., 'castle') into feature vectors and retrieves similar 3D objects using cosine similarity in a shared feature space.

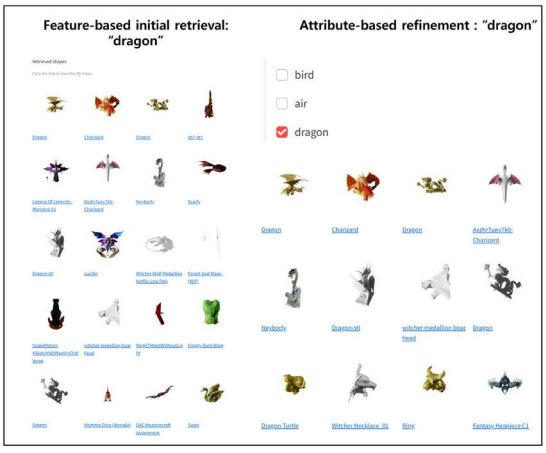
As shown in Fig. 4, the feature-based method retrieves only a few relevant items, whereas the attribute-based method yields results that are more semantically consistent and visually



(a)



(b)



(c)

Fig. 5. Showcases of hybrid retrieval in three scenarios: (a) Attribute-Guided Exploration of Subcategories (b) Intent-Specific Refinement (c) Noise Reduction via Attribute-Based Reranking

coherent across the top-ranked items. For instance, in the 'castle' query, attribute-based retrieval predominantly returns medieval-style buildings, while the feature-based method includes several off-topic or weakly related items (e.g., 'candles', 'living rooms') among the Top-20 results. Overall, feature-based methods are effective at capturing broad similarity, but tend to include semantically off-topic results in lower ranks. In contrast, attribute-based retrieval leverages explicit, interpretable criteria to yield more focused and user-aligned results. This distinction is particularly valuable in exploratory or fine-grained search scenarios.

B. Feature–Attribute Hybrid Retrieval: Use Cases

We evaluate the hybrid prototype through three representative use cases:

1) Attribute-Guided Exploration of Subcategories: In this scenario, the user queries 'hat' using a reference image or text input and receives a diverse set of top-N results based on embedding similarity. The system analyzes these results and surfaces frequent attributes—such as baseball, wizard, or cowboy—as suggested filters. By selecting one or more of these attributes, the user can guide the search toward specific subtypes of interest (e.g., 'baseball cap'), as illustrated in Fig. 5(a).

2) Intent-Specific Refinement: The user issues a query for 'girl' within the *characters–creatures* category. As shown in Fig. 5(b), the initial embedding-based retrieval returns mostly clothing items—such as dresses, shirts, and costumes—due to ambiguity in both the query term and the embedding space. However, given the context of the *characters–creatures* category, the user is likely seeking girl characters rather than garments. To refine the results, the user selects relevant semantic attributes—'girl', 'character', and 'female'—from a dynamically suggested list derived from the top-ranked candidates. The system then re-ranks the results using the hybrid attribute–matching logic, yielding more appropriate 3D models such as anime-style avatars, humanoid characters, and stylized girl figures. This example illustrates how the hybrid system enables intent disambiguation and targeted refinement, allowing users to steer results toward their intended targets without repeating computationally expensive embedding-based searches.

3) Noise Reduction via Attribute-Based Reranking: In this case, we demonstrate how the hybrid system filters out noisy or loosely related results returned by the initial embedding-based retrieval. As shown in Fig. 5(c), the user begins with a feature-based query for 'dragon', which yields a variety of items—including other animals or fantasy accessories—due to the broad semantics captured by the embedding space. To improve relevance, the user activates the 'dragon' attribute in the refinement interface. This filters and re-ranks the Top-K results based on semantic alignment with the selected attribute. The updated ranking promotes dragon-like 3D models with stronger visual and conceptual alignment (e.g., flying dragon characters, statues), while demoting unrelated items. This example illustrates how the hybrid system supports lightweight

semantic filtering without re-embedding or recomputing similarity scores, enabling interpretable and user-directed refinement of noisy retrieval results.

These experiments highlight the complementary strengths of both prototypes: the attribute-only system enables interpretable, category-aware retrieval without embeddings, while the hybrid system allows rich multimodal queries followed by efficient, transparent refinement. Together, they demonstrate the flexibility and effectiveness of our framework across diverse 3D content retrieval workflows.

V. Conclusion

We present a retrieval framework that enhances 3D object search by leveraging interpretable, attribute-based reasoning. Rather than relying solely on latent features, we extract human-aligned attributes—such as shape, material, and style—from textual descriptions of 3D objects.

Our system supports two retrieval modes: an attribute-only prototype enabling category-aware search without embeddings, and a hybrid prototype that combines multimodal queries with dynamic attribute refinement. Both operate on a shared testbed derived from Objaverse using filtered ULIP-2 captions for attribute extraction.

Qualitative experiments show that attribute-guided search improves semantic consistency, precision, and user alignment compared to feature-only methods. The hybrid model further enables intent clarification and noise reduction without the cost of repeated embedding computation.

Future directions include scaling to larger datasets, integrating user feedback loops, and exploring fine-grained attribute learning with supervision or interactive labeling. Ultimately, our goal is to make 3D content retrieval more transparent, interpretable, and semantically controllable.

Acknowledgment

This work was supported by internal fund/grant of ETRI. (Exploration and proof-of-concept of missing technologies in human-computer intelligent interaction (HCoII): Developing scene-to-metaverse analysis and transformation technologies, 24RC1500, Contribution Rate:50%) and 2025 Cultural Heritage Smart Preservation & Utilization R&D Program of Korea Heritage Service, National Research Institute of Cultural Heritage (Project No.: RS-2024-00396158, Contribution Rate:50%)

References

[1] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PmLR, 2021.

[2] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi, "Objaverse: A universe of annotated 3d objects," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13142–13153, 2023.

[3] M. Deitke, R. Liu, M. Wallingford, H. Ngo, O. Michel, A. Kusupati, A. Fan, C. Laforte, V. Voleti, S. Y. Gadre, *et al.*, "Objaverse-xl: A universe of 10m+ 3d objects," *Advances in Neural Information Processing Systems*, vol. 36, pp. 35799–35813, 2023.

[4] L. Xue, M. Gao, C. Xing, R. Martín-Martín, J. Wu, C. Xiong, R. Xu, J. C. Niebles, and S. Savarese, "Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1179–1189, 2023.

[5] M. Liu, R. Shi, K. Kuang, Y. Zhu, X. Li, S. Han, H. Cai, F. Porikli, and H. Su, "Openshape: Scaling up 3d shape representation towards open-world understanding," *Advances in neural information processing systems*, vol. 36, pp. 44860–44879, 2023.

[6] L. Xue, N. Yu, S. Zhang, A. Panagopoulou, J. Li, R. Martín-Martín, J. Wu, C. Xiong, R. Xu, J. C. Niebles, *et al.*, "Ulip-2: Towards scalable multimodal pre-training for 3d understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27091–27101, 2024.

[7] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *International conference on machine learning*, pp. 19730–19742, PMLR, 2023.

[8] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd, "spaCy: Industrial-strength Natural Language Processing in Python," 2020.