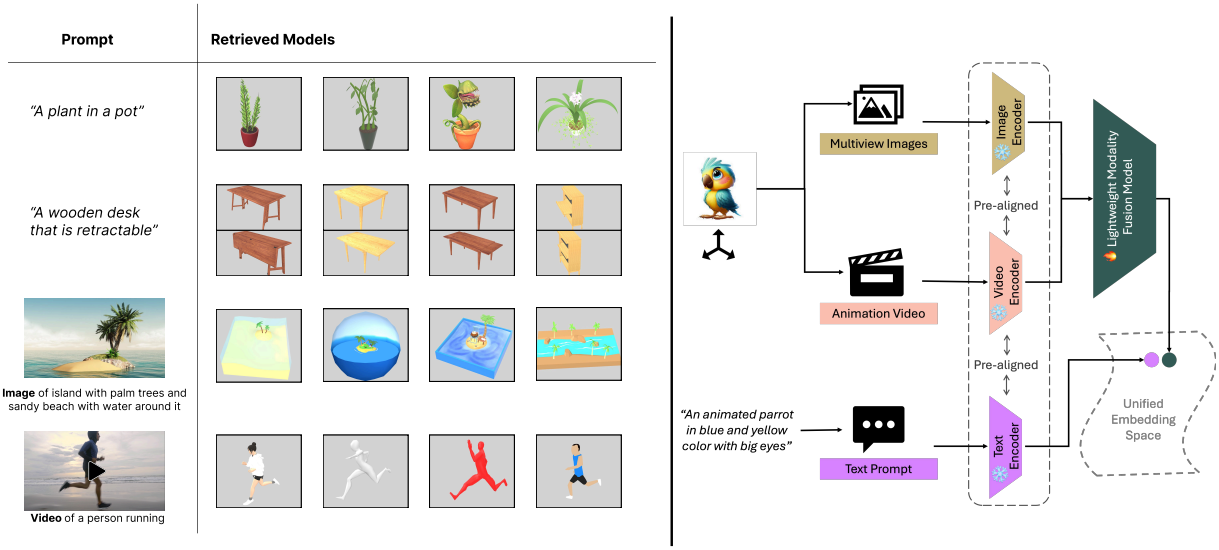
Learning Visual and Motion Aware 3D Model Representations for Semantic Retrieval

Sruti Srinidhi and Anthony Rowe



(a) Examples of 3D models retrieved using text, image, and video prompts

(b) High-level methodology for learning unified 3D model representations

Figure 1: **3D model retrieval examples and methodology. (a) Examples of 3D models retrieved for different input prompts.** Prompts can be textual (e.g., descriptions of visual or dynamic features such as a "retractable table"), or based on sample images and videos that capture visual appearance or motion. **(b) High-level methodology for learning 3D model representations.** Each 3D model is represented using multiview images to capture appearance and videos to capture dynamics. A trainable multimodal fusion model combines the outputs of pretrained encoders into a unified embedding space, enabling cross-modal retrieval.

Abstract—Virtual Reality (VR) content creators today face lengthy trial-and-error cycles when searching for 3D models, which relies heavily on manually annotated metadata. However, this approach struggles to capture the rich visual and dynamic details inherent to 3D models. Prior work has attempted to address this by rendering models as sets of images, but has largely remained restricted to category-level search and static geometry. In VR environments where asset behavior is often as important as appearance, we present a visual and motion aware 3D retrieval method that represents each model through multiview renders and short animations, extending search beyond static geometry to include dynamic behaviors. Pretrained image–text and video–text encoders extract features that are combined into a unified embedding by training a lightweight multimodal fusion model. This enables 3D model retrieval from text, images, and videos without requiring abundant and detailed ground truth data, supporting more nuanced and semantically meaningful search. On text queries referencing both appearance and motion, our method achieves a 23-percentage-point improvement in top-5 retrieval accuracy on the Objaverse dataset, while also attaining competitive accuracy on the ModelNet40 dataset and maintaining robustness as the model library size scales. A user study further shows that participants strongly preferred our retrievals compared to the one used by Sketchfab, one of today's most widely used 3D repositories. To support future work, we release 26k multi-view renders, animation clips, and 500 manually annotated visual and motion aware captions for the Objaverse dataset. The data and implementation is available at https://github.com/srutisrinidhi/3DModelSearch.

Index Terms—Virtual Reality, 3D Asset Retrieval, Multimodal Machine Learning, Motion-Aware Embeddings, Scene Authoring

1 Introduction

The rapid growth of virtual reality (VR) and immersive applications has created a strong demand for effective search and retrieval of 3D models. Current solutions like online repositories (e.g. Sketchfab [32]) or asset libraries in 3D authoring tools (like Unity [35] and Unreal [5]) largely rely on metadata-based search, where results are organized by manually assigned tags or broad category labels. While effective for coarse filtering, these approaches fall short in supporting natural language queries and in capturing the semantics of how 3D models look and move. As a result, creators often face long trial-and-error cycles when sourcing assets for interactive experiences. In addition, with the growing use of Large Language Models (LLMs) in authoring 3D scenes, being able to semantically retrieve 3D models is essential.

Recent advances in Vision Language Models (VLMs) have transformed retrieval in 2D domains, enabling robust alignment between text, images, and video. However, their potential for 3D content re-

• *Sruti Srinidhi is with Electrical and Computer Engineering, Carnegie Mellon University. E-mail: ssrinidh@andrew.cmu.edu*
• *Anthony Rowe is with Electrical and Computer Engineering, Carnegie Mellon University and Bosch Research. E-mail: agr@andrew.cmu.edu.*

Manuscript received xx xxx. 202x; accepted xx xxx. 202x. Date of Publication xx xxx. 202x; date of current version xx xxx. 202x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.202x.xxxxxxx

mains underexplored. Prior work primarily focuses on static shape descriptors or learned embeddings of geometry, with limited consideration of motion. This neglects a critical aspect of 3D models in VR, where many models are designed to be animated, and their dynamic behavior is central to their meaning (e.g., a "running dog" vs. a "sleeping dog").

Unlike images and videos on the Internet, there is a huge scarcity of richly annotated 3D models, particularly those with animations or motion information. Public repositories typically provide only minimal textual descriptions or user-assigned tags, which rarely capture fine-grained visual or motion semantics such as "a person wearing a black baseball cap walking", or "a three ducks of different sizes swimming across a pond". This lack of detailed captions makes it difficult to train dedicated 3D–text encoders at scale, in stark contrast to the abundance of data available for image–text or video–text learning. The problem is further amplified in VR contexts, where asset behavior is often as important as asset appearance. To overcome this, we adopt a proxy strategy: rather than learning directly from scarce 3D annotations, we generate 2D renderings and animated sequences of each model, enabling the direct reuse of pretrained image and video retrieval models.

In our work, we present a visual and motion aware 3D model representation learning and retrieval strategy that treats assets as multimodal entities composed of both visual features and dynamics. Each model is represented through a combination of static multiview images and short rendered animation clips, which are aligned with natural language queries using state of the art pretrained image–text and video–text retrieval models. Rather than introducing an entirely new algorithmic optimization or architecture, our primary novelty lies in introducing motion as an essential, previously underutilized dimension for 3D model representation, and establishing an effective proxy strategy to capture these dynamic semantics. Surprisingly, we find that a lightweight fusion strategy, such as simple score averaging across modalities, consistently outperforms more complex joint embedding learning methods. These results highlight motion as an essential retrieval dimension while also showing that simple integration methods with lower resource requirements can be highly effective.

We validate our approach through both benchmarks and user studies. Quantitatively, we compare our retrieval accuracy against prior state-of-the-art methods and ablated variants of our model. Our method achieves a 23-percentage-point improvement in top-5 retrieval accuracy (87% vs. 63.8% for the next best prior work) on the Objaverse dataset [4] for prompts involving both visual and motion cues, and competitive performance (85%) on the ModelNet40 dataset [38], which has only static models (geometry and texture). The gains are most pronounced on larger datasets, highlighting that our approach learns more discriminative features and remains robust as the corpus scales. Qualitatively, we conduct a user study comparing our system to Sketchfab, an industry standard for 3D model search. Participants consistently preferred our results, rating our retrievals higher overall (3.88 vs. 2.88 out of 5) and scoring our animations substantially better (3.96 vs. 1.72). Notably, our system also significantly outperformed Sketchfab in visual appearance alignment (3.80 vs. 2.74), demonstrating that adding motion does not come at the cost of visual precision. These findings demonstrate the practical benefits of visual and motion-aware search in real-world VR workflows. Finally, to support future research, we release a dataset of 26k rendered images and video clips for Objaverse, with 500 manually annotated motion-aware captions, providing a compact yet valuable benchmark for the community.

Our key contributions are:

- A **3D representation learning strategy that leverages multi-view images and short animation clips**, enabling the reuse of powerful pretrained image and video language models.
- Identification of **motion as an essential retrieval dimension**, showing that dynamic behavior is critical for aligning language with 3D model semantics.
- A **benchmark and dataset for text-to-3D retrieval**, including 26k multi-view images and animation clips, as well as 500 detailed motion-aware captions for Objaverse. [1]
- Quantitative benchmarks and user studies showing the effectiveness of multimodal fusion and the value of semantic retrieval over Sketchfab.

2 Related Work

2.1 3D Representation Learning

Learning effective 3D representations has been a long-standing goal, with approaches spanning geometry-based, view-based, and multimodal paradigms. Early 3D encoders learn directly from geometry, capturing invariances to permutation, viewpoint, and rigid transforms. Representative models include PointNet/PointNet++ [27, 28], DGCNN [37], and multi-view CNNs such as MVCNN [33] and MVTN [9]. Subsequent work explored self-supervised and transformer-based pretraining [25, 39, 43], improving robustness but remaining limited to category-level semantics and under-representing fine-grained or open-vocabulary attributes, primarily due to the lack of richly annotated training data.

The rise of large-scale pretraining has shifted focus toward generalizable semantic encoders aligned across modalities. Image–text models like CLIP [30] and ALIGN [14], video–text encoders such as VideoCLIP [40] and InternVideo [36], and multimodal spaces like LanguageBind [49] and ImageBind [8] enable zero-shot comparison across images, videos, and text. Although training analogous encoders directly on 3D models is not yet feasible due to limited large-scale 3D–text data, these image- and video-text models provide transferable priors that we leverage as building blocks.

To transfer these priors to 3D, view-based methods render objects and aggregate 2D features. PointCLIP [46], CLIP2Point [12], and CLIP-Goes-3D [10] pool CLIP features across multiviews, while captioning pipelines such as CAP3D [21] leverage renderings to produce natural-language supervision. More recently, language-anchored pretraining approaches, like ULIP [41] and OpenShape [19], jointly align 3D, image, and text encoders to support open-vocabulary retrieval and zero-shot tasks [17, 18, 47].

Despite these advances, most methods operate on static geometry or image sets, limiting their ability to distinguish models with similar appearance but different dynamics. Moreover, reliance on synthetic captions may introduce noise through hallucinated attributes. Our work targets this gap by combining pre-aligned image and video encoders with pooled multiview renderings and short turntable clips to learn motion-aware 3D embeddings without requiring explicit 3D captions.

2.2 3D Search and Retrieval

3D search systems span a spectrum of approaches, from geometric similarity and structured metadata to context-aware and language-driven retrieval. Early methods focused on shape descriptors and sketch-based search [7, 16, 22, 24], later extended by scene context and relational reasoning [6]. Metadata-driven systems leveraged tags, parts, and annotations, which are mostly manually annotated, for structured queries [1, 23, 48].

Recent work has shifted toward embedding-based retrieval using multimodal encoders aligned across text, image, and 3D modalities [10, 19, 41]. These models support open-vocabulary and zero-shot retrieval, and have been integrated into authoring tools for VR scene generation and layout composition [13, 42, 45]. In immersive settings, such systems allow users to retrieve, place, and manipulate 3D models using natural language or high-level intent, bypassing traditional menu-based interfaces. For example, VRCopilot and Holodeck support compositional scene control through voice or text prompts, enabling iterative scene editing and spatial reasoning in real time. Other frameworks like SceneSuggest and ATISS [26, 31] incorporate learned spatial priors to recommend plausible object placements, while diffusion-based models [20, 29] generate realistic room layouts conditioned on functional context. These tools reflect a broader trend toward multimodal, context-aware VR authoring workflows that unify retrieval, layout,

[1] Data and code can be found at https://github.com/srutisrinidhi/3DModelSearch

and interaction into a single semantic interface. Together, these techniques reflect the growing convergence of 3D understanding, semantic alignment, and interactive content creation.

3 Our Approach

Our goal is to enable text-driven retrieval of 3D models by learning multimodal embeddings that capture both *appearance* and *motion*. Unlike images or videos, 3D models rarely come with high-quality natural language descriptions, making it difficult to train a dedicated 3D-text encoder. To address this, we adopt a proxy strategy: we render 3D models as **multiview images** and **short animation clips**, and leverage powerful pretrained encoders that jointly aligns text, image, and video in a shared representation space. In our implementation we use the LanguageBind [49] pretrained encoders, but the method is encoder-agnostic: any model that provides a unified image–video–text space can be swapped in without changing the pipeline. This design supports text, image, and video based retrieval of 3D models, enabling more expressive search (Fig. 1b). This section details how we learn these multimodal 3D representations from the individual modalities (Fig. 2).

3.1 Multiview Image Representation

For each 3D model, we generate $M = 6$ canonical renderings from fixed orientations $(\pm x, \pm y, \pm z)$. We chose these views because they give uniform coverage of all principal sides with minimal redundancy, whereas fewer views miss at least one side. Each view I_m is encoded using the LanguageBind image encoder f_I:

$$z_m = f_I(I_m), \quad z_m \in \mathbb{R}^d, \quad m = 1, \ldots, 6 \tag{1}$$

While each z_m captures a valid perspective of the model, they contain overlapping information and may emphasize different features. To form a single representation z_{img}, we train lightweight fusion models that learn to combine the six views. We experiment with:

- **MLP fusion (order-invariant):** a lightweight fusion model that scores each view with a *shared* per-view MLP, softmax-normalizes the scores, and forms a symmetric weighted sum before a final projection. This parameter-efficient design is robust on smaller datasets.
- **Attention fusion (content-adaptive):** a self-attention layer over the six view embeddings that models inter-view interactions and assigns content-aware weights, emphasizing views with unique or discriminative cues; the extra flexibility typically helps on larger datasets at the cost of more compute.

Training objectives. Because high-quality text annotations for 3D models are scarce, we train the fusion model without paired captions using two self-consistency losses.

Cosine Similarity (intra-model). We encourage the fused representation to stay close to its constituent views:

$$\mathscr{L}_{\cos} = 1 - \max_{m \in \{1,\ldots,6\}} \cos\big(z_{\text{img}}, z_m\big) \tag{2}$$

Batch-Wise Contrastive Loss (inter-model). Let the batch size be B. For each model $i \in \{1, \ldots, B\}$ we sample one positive view index m_i^+ uniformly from $\{1, \ldots, 6\}$ and take the corresponding normalized embedding $v_i = \frac{z_{i,m_i^+}}{\|z_{i,m_i^+}\|}$. We also L2-normalize the fused embedding $o_i = \frac{z_{\text{img},i}}{\|z_{\text{img},i}\|}$. We form the logits matrix

$$\ell_{ij} = \frac{o_i^\top v_j}{\tau}, \quad i, j \in \{1, \ldots, B\} \tag{3}$$

and apply cross-entropy with identity labels (diagonal as positives, off-diagonals as negatives):

$$\mathscr{L}_{\text{ctr}} = -\frac{1}{B} \sum_{i=1}^{B} \log \frac{\exp(\ell_{ii})}{\sum_{j=1}^{B} \exp(\ell_{ij})} \tag{4}$$

where τ is a temperature hyperparameter.

For each fused embedding, the randomly chosen view of the *same* model is treated as the positive, while the corresponding views from all other models in the batch serve as negatives.

Total objective., where $\lambda_{\cos}$ and λ_{ctr} are the weights representing how much each loss contributes to the total loss.

$$\mathscr{L}_{\text{img}} = \lambda_{\cos} \mathscr{L}_{\cos} + \lambda_{\text{ctr}} \mathscr{L}_{\text{ctr}} \tag{5}$$

3.2 Video Representation

To capture dynamics, we render a short animation clip V for each 3D model and encode it with the frozen LanguageBind video encoder f_V:

$$z_{\text{vid}} = f_V(V), \quad z_{\text{vid}} \in \mathbb{R}^d \tag{6}$$

This representation captures temporal semantics (e.g., "walking," "flying," "collapsing") that static images cannot convey. We chose a 6 seconds video clip given our dataset, as the animations were shorter than that time frame, although this length can be tuned given the dataset and the encoder's context window.

3.3 Multimodal Fusion

Finally, we construct a unified multimodal embedding z_{3D} that integrates both z_{img} and z_{vid}. Although we experimented with trainable fusion layers (results in Table 1), similar to the multiview image fusion model, we observed that a simple average provided consistently higher accuracy:

$$z_{\text{3D}} = \frac{1}{2}(z_{\text{img}} + z_{\text{vid}}) \tag{7}$$

This result suggests that pretrained encoders are already well aligned across modalities, and that computationally lightweight fusion is both effective and robust. Moreover, because z_{3D} lies in the same embedding space as the pretrained text encoder, retrieval can be performed directly.

3.4 Multimodal Retrieval

At query time, a natural language prompt T is encoded with the LanguageBind text encoder f_T to obtain $z_{\text{text}} = f_T(T)$. Each model's multimodal embedding z_{3D} is then compared with z_{text} in the shared space using cosine similarity:

$$\text{sim}(z_{\text{3D}}, z_{\text{text}}) = \frac{z_{\text{3D}} \cdot z_{\text{text}}}{\|z_{\text{3D}}\| \, \|z_{\text{text}}\|} \tag{8}$$

Since the LanguageBind framework pre-aligns text, image, and video encoders into a common space, our system naturally supports cross-modal retrieval from multiple query modalities, including text, images, or videos. Because these encoders share a unified representation space, the similarity metrics and mathematical distances are entirely symmetric across modalities. Consequently, the quantitative performance trends for text-based queries (detailed in Section 5.1) serve as a direct proxy for the system's underlying image- and video-based query capabilities.

We restrict our quantitative benchmark evaluation to text queries because standard 3D evaluation datasets explicitly provide natural language captions as their ground-truth retrieval targets. Instead, we validate our cross-modal capabilities for image and video queries qualitatively through our user study (Section 5.4).

For scalability, all z_{3D} embeddings are stored in a vector database, specifically ChromaDB [2], which allows an efficient approximate search for nearest neighbor using cosine distance, and hence can be used to retrieve similar 3D model embeddings. This allows interactive, low-latency retrieval across thousands of models, making the system practical for VR content creation applications.

4 Dataset Generation

To train and evaluate our method, we construct a multimodal dataset based on Objaverse [4], supplemented with ModelNet40 [38] for additional baselines.

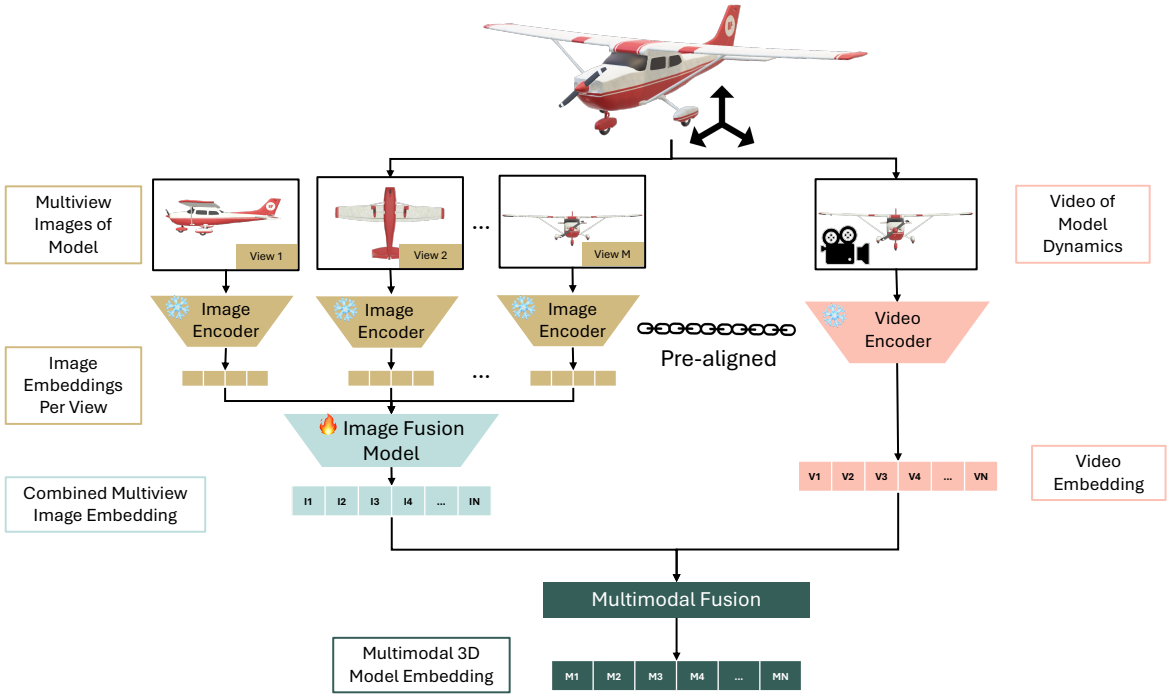


Figure 2: **Multimodal 3D model representation learning for text-to-3D retrieval.** Each 3D model is rendered from M canonical views. A frozen, pretrained image encoder produces per-view embeddings, which a lightweight, trainable image-fusion model aggregates into a single multiview image embedding. In parallel, a frozen video encoder computes a video embedding from a short dynamics clip. The final 3D model representation is obtained by a non-trainable multimodal fusion step implemented as the mean of the multiview image and video embeddings. The resulting 3D embedding is aligned in the same embedding space as the text encoder, enabling text-to-3D retrieval. The snowflake and fire symbols denote frozen and trainable parameters, respectively.

Objaverse We generate 6 canonical multiview renderings for each 3D model and record 6-second video clips to capture motion and animation dynamics. This results in a dataset of over 26,000 models. While prior work has released static renderings, our contribution is the first large-scale release of rendered *videos* for 3D models. We make all data and rendering scripts (for Blender [3]) publicly available to facilitate reproducibility and future research.

Captions We used the automatically generated Cap3D captions [21] as a starting point. However, these captions often contain errors as they are generated using large language models and lack descriptions of dynamic behaviors. They also have high variability in terms of length and descriptive detail. As such, we do not use them as our ground truth retrieval. Instead, we manually curate a benchmark set of 500 models with detailed captions describing both appearance and motion. This provides a small but valuable benchmark for motion-aware 3D retrieval.

ModelNet40 For comparison, we also generate multiview images and videos for ModelNet40. Because these models are static, the resulting videos reduce to extended renderings of fixed geometry without meaningful motion. Although they lack dynamics, this provides a useful baseline for evaluating retrieval performance when only appearance cues are available.

Overall, our dataset contributions are: (i) the first large-scale release of rendered videos for Objaverse models, (ii) a curated benchmark of 500 motion-aware captions, and (iii) reproducible Blender pipelines for extending the dataset.

5 Evaluation

5.1 Text-to-3D Retrieval Accuracy

To evaluate how well our 3D representation captures both visual and motion semantics, we measure text-based retrieval accuracy on two benchmarks. On the Objaverse dataset, which consists of publicly available models from Sketchfab [32], we use our manually curated set of 500 detailed captions that describe both appearance and dynamics. On ModelNet40, we use the standard category-level captions, which are more representative of prior baseline setups. We compare against three representative baselines: ULIP [41], OpenShape [19], and CLIP-Goes-3D [10]. We also compare against metadata-based search, GPT-4o–generated descriptions, and single modality image-only and video-only based retrieval. In addition, we ablate different multiview image fusion strategies, including simple averaging, max pooling (closest image match), and trainable fusion modules (MLP and attention), as described in Section 3. We also experiment with a trainable multimodal fusion model, following the same training strategy as the multiview image fusion, and compare it to a simple mean of the multiview image and video embeddings (our solution).

We evaluate retrieval on a model library of 500 models and report Top-1, Top-5, and Top-10 accuracy for both datasets. We also compute the Mean Reciprocal Rank (MRR), defined as $\text{MRR} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\text{rank}_i}$, where rank_i is the position of the first correct retrieval for query i and N is the number of retrieval queries made, which captures how highly the correct result is ranked on average across all queries. Having an MRR of 1 indicates that the top-1 retrieval is the expected model. For Objaverse, each caption is instance-specific and corresponds to a single target model. In contrast, ModelNet40 uses category-level captions, so multiple models may match a query, and success is defined as retrieving any model of the correct category within the top-k results. The results can be seen in Table 1.

Table 1 also shows that our multimodal approach outperforms all baselines on Objaverse, achieving 87.3% Top-5 accuracy and an MRR of 0.806. Among baselines, GPT-4o–based text descriptions are the strongest on Objaverse, while OpenShape achieves the best results on ModelNet40, which primarily reflects category-level appearance cues. Our ablations highlight several trends. Multi-view image features consistently improve over single-image retrieval, and attention-based

Table 1: Zero-shot text to 3D retrieval on Objaverse and ModelNet40. Top-k results are reported as percentages. MRR is shown as mean.

Method	Objaverse				ModelNet40			
	Top1	Top5	Top10	MRR	Top1	Top5	Top10	MRR
Baselines								
ULIP	40.5	59.0	66.0	0.483	37.5	55.0	60.0	0.453
OpenShape	36.8	63.8	72.3	0.481	**80.0**	**95.0**	**95.0**	**0.868**
Clip-Goes-3D	1.8	9.5	14.3	0.052	12.5	25.0	30.0	0.174
Metadata	25.8	41.5	47.8	0.321	—	—	—	—
Text Description (GPT-4o)	55.0	74.5	78.8	0.634	60.0	85.0	87.5	0.694
Ours (ablations)								
Single Image	40.0	58.8	63.3	0.481	22.5	35.0	40.0	0.283
Average Across All View Images	65.3	76.8	79.8	0.704	60.0	80.0	82.5	0.700
Maximum Similarity Across All View Images	35.3	60.5	70.5	0.461	65.0	82.5	85.0	0.740
Video Only	62.5	77.8	80.8	0.692	55.0	65.0	70.0	0.585
Multiview Image Fusion using MLP	68.5	82.3	84.8	0.743	70.0	77.5	82.5	0.745
Multiview Image Fusion using Attention	70.5	83.3	85.8	0.760	70.0	80.0	85.0	0.740
Ours: Multimodal Retrieval (Trained Fusion Model)	60.5	83.0	85.7	0.699	—	—	—	—
Ours: Multimodal Retrieval (Mean of Multiview Image + Video)	**75.5**	**87.3**	**89.5**	**0.806**	70.0	85.0	85.0	0.721

fusion yields the strongest image-only results. Video-only retrieval demonstrates the value of motion, and combining images with videos further boosts performance. Notably, simple averaging of image and video embeddings surpasses more complex trained fusion, suggesting that pretrained encoders are already well aligned across modalities. Together, these results confirm that motion-aware multimodal embeddings provide the most robust retrieval performance.

On ModelNet40, our approach achieves competitive performance, though OpenShape remains the strongest baseline. This gap is expected, as ModelNet40's 3D models are static meshes without textures or motion, limiting the benefit of our multimodal design. In this setting, the video modality effectively is another rendering of static geometry, providing little additional signal beyond the multi-view images. Consequently, our multimodal fusion results are very close to those of the image-only models, consistent with the absence of dynamic cues in the dataset.

5.2 Granularity Stress Test

To evaluate the system's sensitivity to subtle visual and motion cues, we conduct a fine-grained retrieval analysis within a controlled library. This corpus consists of six variants of a single base 3D model performing three distinct actions: punching, jumping jacks, and standing in a T-pose. These models also include a secondary visual variable: cap color (red vs. blue). This setup serves as a stress test for the system's ability to distinguish between models where the geometry and motion are highly similar, but specific semantic details differ.

As shown in the relative attribute alignment heatmap in Figure 3, the method demonstrates high discriminative power for fused motion and appearance queries. For a complex prompt like "A man in a red cap punching," the system successfully isolates the correct target with a relative similarity score of 1.00. However, the matrix reveals an interesting hierarchy in attribute strength: visual attributes, such as cap color, often act as stronger semantic anchors than the action itself. For instance, when the system is prompted for a "red cap punching" model, the second-highest matches are typically other models wearing red caps performing different actions, rather than models performing the correct action in a different color. We hypothesize that this is due to the nature of the underlying multimodal embedding. Because the embedding is composed of both image and video encoders, a static visual attribute like "red cap" provides a consistent signal across both modalities. In contrast, the motion signal for "punching" is primarily captured by the video encoder alone. This results in visual appearance features occasionally overshadowing motion cues during the retrieval process. On the right of Figure 3, we provide two renderings of the models to illustrate how these different attributes, such as cap color, as well as stills from the animations, appear in the evaluation.

5.3 Retrieval Robustness to Model Library Size

While our model performs well on smaller libraries (e.g., 500 models), practical model libraries can be much larger. We therefore study how retrieval accuracy scales with library size. To quantify robustness, we measure the proportion of retrieval accuracy retained as the library grows, where higher values indicate greater resilience to scaling. Specifically, let *Top-5@N* denote the probability that the groundtruth model appears among the top five results retrieved from an *N*-item library. We report retained accuracy as

$$\text{retained}(N) = \frac{\text{Top-5@N}}{\text{Top-5@500}} \times 100,$$

where 100% indicates no degradation relative to the 500-model setting. We evaluate the retained retrieval accuracy at library sizes of 500, 1k, 10k, and 26k models.

From Figure 4, we observe that retrieval performance decreases for all models as the library size grows, but the rate of degradation differs significantly. Baseline methods such as OpenShape and ULIP drop sharply, while CLIP-Goes-3D struggles even at moderate scales. In contrast, our model retains 57% of its top-5 accuracy even at 26k candidates, demonstrating substantially greater robustness to scaling. This indicates that our multimodal embeddings capture more discriminative and transferable features, enabling reliable retrieval performance even in large-scale model libraries.

5.4 Qualitative Retrieval Accuracy User Study

To complement our quantitative evaluation, we conduct a user study[2] comparing our retrieval system with Sketchfab. We recruit 14 participants, eight with prior experience using and searching for 3D models and six with no previous exposure to 3D modeling.

Each participant is initially asked to use the same four fixed prompts (two text, one image, one video) when searching for 3D models on both the retrieval systems (ours and sketchfab). The participant is then

[2]The user study procedures were reviewed and received an exemption determination by the Institutional Review Board at Carnegie Mellon University (Protocol #STUDY2025_00000430) .

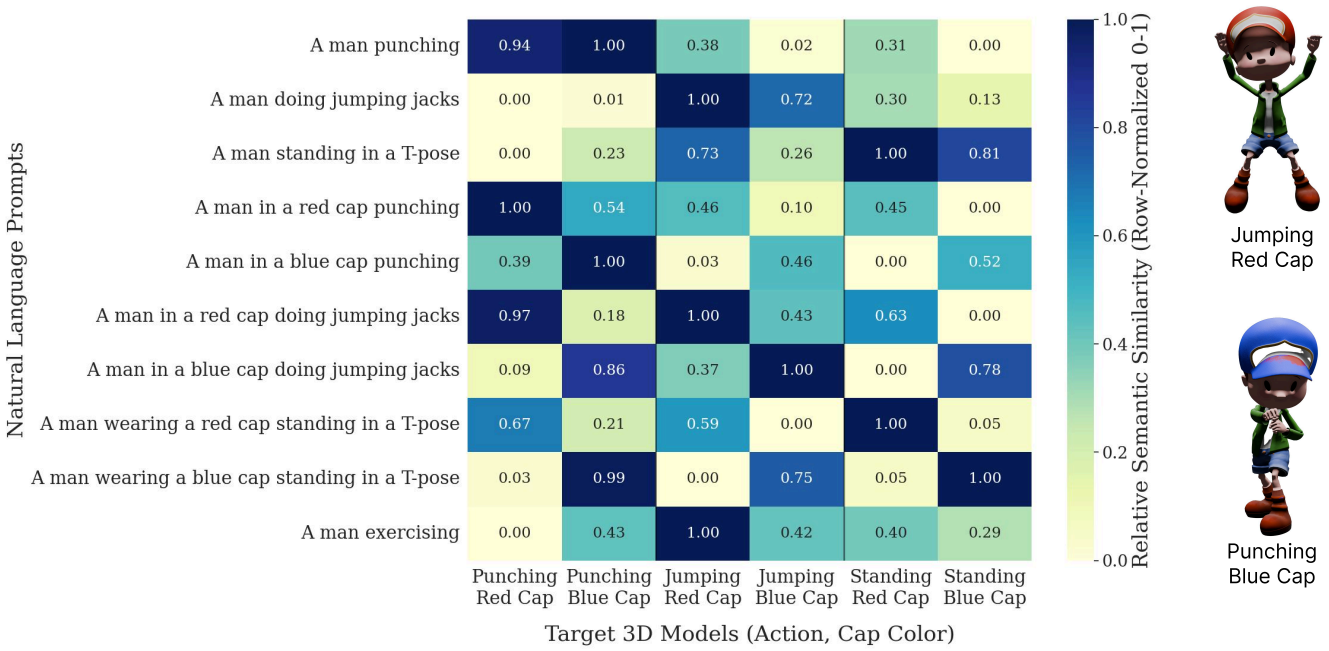


Figure 3: **Granularity Stress Test Analysis.** (Left) Heatmap displaying row-normalized relative semantic similarity scores for the 6-model library for each given text prompt. (Right) Renderings of the "Red Cap" and "Blue Cap" variants with different motion cues. These renderings illustrate the visual differences in the corpus and representative stills from the animation sequences used for retrieval.

asked to come with 4 more prompts on their own. For each prompt on each system, they inspect the top-5 retrievals and answer three main questions on a 5-point Likert scale (higher is better): (1) overall relevance to the prompt, (2) relevance of visual appearance, and (3) relevance of motion and animation. The motion score is ignored in our data analysis when the prompt does not specify any kind of animation.

The average scores for all prompts are shown in Fig. 5. Our retrieval system outperforms Sketchfab on every criterion: overall relevance (3.88 vs. 2.88), visual match (3.80 vs. 2.74), and motion/animation match (3.96 vs. 1.72). The largest improvement is in motion scores, indicating better alignment with dynamic semantics, though we also observe general improvement in retrieval quality, particularly for detailed prompts. When asked to select their preferred platform for each prompt, participants chose our retrieval system 85.7% of the time for the four fixed prompts that we provided. However, for the prompts the participants created themselves, our system was preferred only 67.8% of the time. We attribute this discrepancy primarily to the limited scope of our model library (26k Objaverse models) compared to Sketchfab's substantially larger, open-ended database. Consequently, specific models that users searched for (e.g., "library with a worm resting on one of the books" or "a hacker typing at a keyboard") were often unavailable in our collection, while Sketchfab's broader catalog increased the likelihood of finding relevant matches. Additionally, since Sketchfab does not natively support image- or video-based queries, participants had to reformulate such prompts as text descriptions. Given these limitations, we report descriptive statistics without claims of statistical significance. We specifically chose Sketchfab as a baseline for this user study because it represents the widely accepted industry standard for metadata-tag-based search and contains a 3D model distribution that directly reflects the Objaverse dataset used in our model library.

5.5 Performance Overhead

On a Linux desktop with an RTX 4090, the **per-model time to add a new 3D model (capture → embedding) is ≈ 31.4** s in total: image capture (6 views) 1.84 s, video capture (6 s clip) 19.30 s, image embedding (6 images) 8.38 s (median), and video embedding 1.90 s (median). Multimodal fusion (multiview fusion over six images, averaged with video) and adding to the database takes 1.8 ms per model, which is quite small compared to the previous steps.

For retrieval on a 26k-item database, the database lookup itself is fast: top-10 retrieval latency is 1.5 ms (p50) and 52 ms (p95). **End-to-end retrieval (text embedding + retrieval) is 2.20** s (p50) and 2.25 s (p95), dominated by getting the text embedding at ≈ 2.19 s.

The **most memory-consuming phase is image capture** (GPU ∼ 8.38 GB VRAM; CPU ∼ 7.52 GB), followed by video capture (GPU ∼ 7.52 GB; CPU ∼ 8.59 GB). The embedding generation steps use less GPU memory (∼ 3.54–4.08 GB) but show higher CPU peaks (∼ 9.13 GB).

Although processing and indexing new models require the bulk of computation and time, these steps occur only once at ingestion. Once a model has been captured, embedded, and added to the database, subsequent searches incur negligible overhead. This design makes the system practical and scalable as the one-time ingestion cost is amortized over all future queries, while retrieval remains lightweight and responsive even at large corpus sizes. It is important to note that all these figures are rough estimates intended to convey operational cost. The absolute values vary with software and hardware infrastructure, scene geometry and textures, and corpus scale.

6 Discussion

Our results highlight the promise and current limitations of multimodal 3D retrieval. By jointly leveraging multiview images and rendered motion clips, our approach outperforms existing baselines, especially for prompts that reference dynamic behaviors. The ability to align text with visual appearance and motion cues makes retrieval more expressive and closer to how creators naturally think about models when building immersive environments.

6.1 Semantic Precision and Failure Modes

While our method achieves a significant 23-percentage-point improvement in top-5 accuracy over the next best prior work, a qualitative analysis reveals specific scenarios that challenge current multimodal embedding spaces:

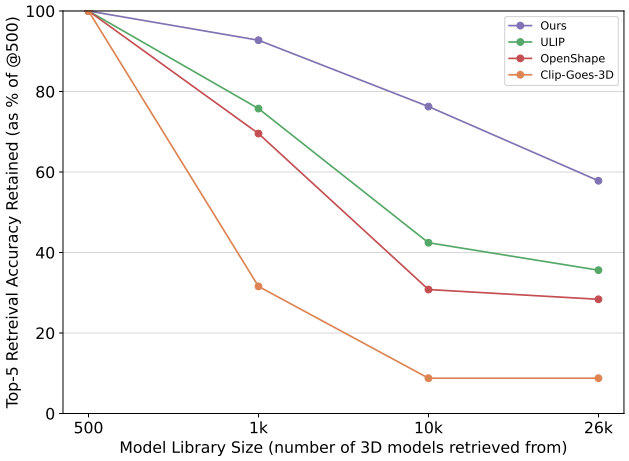


Figure 4: **Top-5 retrieval accuracy retained as the model library scales.** Each line shows the top-5 accuracy retained for our model and baselines as model library scales from 500 to 1k, 10k and 26k.

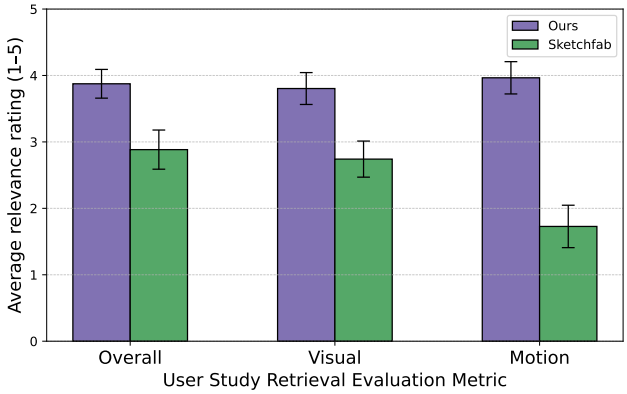


Figure 5: User study (n=14) comparing our retrieval system and Sketchfab. Bars show mean Likert ratings (1–5; higher is better) for Overall, Visual, and Motion relevance of the top-5 retrieved models across all prompts and error bars denote SEM. Our retrieval system scores higher than Sketchfab on all three metrics, with the largest margin on Motion.

1. **Detailed and Over-Specified Queries:** We observe that as queries become increasingly detailed (e.g., specifying textures, multiple accessories, and specific gaits simultaneously), the system can struggle to weigh these features appropriately. In such over-specified cases, the embedding may focus more on a single salient detail (like a bright color) while neglecting more subtle geometric or motion cues, suggesting a limit to the semantic density a single unified embedding can hold.
2. **Visual vs. Motion Dominance:** In many cases, static visual features tend to overpower motion cues within the embedding space. For instance, a query for a "running dog" may prioritize a "sitting dog" over a "running cat" because the visual signal from the multiview image encoder is more consistent than the temporal signal from the video encoder. Since visual attributes like species, color, or texture are present across all rendered views while motion is captured only in the temporal sequence, the system naturally biases toward geometric and appearance-based similarity.
3. **Viewpoint-Dependent Coverage Gaps:**Because our framework relies on six canonical views $(\pm x, \pm y, \pm z)$, small but critical details located at oblique angles, such as a specific logo on a shoulder or a small tool held in a hand, may be missed if they are not captured by the fixed rendering angles.
4. **Resolution of Logical Negations:** The system exhibits a notable negation blindness when processing queries that exclude specific attributes, such as "a man without a hat." In these instances, the underlying encoders often prioritize the salient keyword (e.g., "hat") while ignoring the logical operator "without." This "bag-of-words" behavior is a documented limitation in contrastive vision-language models, where the training distribution primarily emphasizes the presence of objects rather than their absence. As a result, the embedding space effectively treats negated terms as positive anchors, leading to the retrieval of assets containing the very features intended for exclusion. [11, 15, 34, 44]

6.2 Modality and Scaling Trends

Our experiments reveal that on ModelNet40, which features static, textureless geometry, the benefits of multimodality are diminished. Since the video stream provides little additional signal beyond the rendered views in this context, it suggests that motion-aware retrieval is most valuable for models designed to be animated rather than purely geometric datasets. Similarly, while our system demonstrates robustness as the retrieval library scales, accuracy inevitably declines with very large collections, pointing to an open challenge for scalable 3D search.

Another notable finding is that simple averaging of image and video embeddings often outperforms more complex trainable fusion schemes. This suggests that pretrained encoders are already well aligned across modalities and lightweight integration can suffice. However, this may also reflect the limited scale of current training data; with larger and more diverse multimodal datasets, learned fusion strategies could potentially yield greater gains.

6.3 Future Directions

Although our current framework uses six canonical views and a single rendered video per model, there is substantial room to explore richer input modalities. Extending to multiview video could provide more comprehensive coverage of both geometry and motion, particularly for models with complex dynamics. Beyond simply adding more views, an important open question is how to identify which views or motion segments are the most informative to improve robustness and efficiency. Additionally, because we rendered all models against Blender's default neutral grey background, evaluating generalization to real-world environments remains an important next step. Integrating variations in rendering parameters, such as randomized backgrounds, diverse lighting configurations, and environmental noise, during dataset generation could significantly improve the system's robustness against background noise in open-ended deployment contexts.

More broadly, advancing multimodal embeddings of 3D models has implications that extend beyond retrieval. A richer semantic understanding of individual 3D models lays the foundation for reasoning at the scene level. Since scenes are ultimately composed of multiple interacting models, the ability to capture fine-grained appearance and motion at the model level can extend to semantic scene understanding. This opens the possibility of retrieving entire scenes based on high-level descriptions or supporting context-aware composition where models are selected to fit naturally within larger environments.

7 Conclusion

This paper presents a method for representing 3D models that captures both visual appearance and motion. Because reliable 3D captions are scarce, we leverage pretrained image– and video–text encoders and train a multimodal fusion model. Compared to baselines, our approach achieves higher retrieval accuracy, scales more robustly, and is preferred by users over Sketchfab for semantic search, indicating practical value. Simple yet effective, it is suitable for large-scale retrieval and will continue to improve as encoders advance, without additional 3D data. This work takes a step toward making natural language a more effective interface for exploring and retrieving 3D content.

Acknowledgments

This work was supported by the NSF under award CNS-1956095 and Bosch Research. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

References

[1] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li et al. Shapenet: An information-rich 3d model repository, 2015. doi: 10.48550/arXiv.1512.03012 2

[2] Chroma. Chroma - the open-source embedding database, 2023. Accessed: 2025-03-30. 3

[3] B. O. Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 4

[4] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt et al. Objaverse: A universe of annotated 3d objects, 2022. doi: 10.48550/arXiv.2212.08051 2, 3

[5] Epic Games. Unreal engine. 1

[6] M. Fisher and P. Hanrahan. Context-based search for 3d models. In *ACM SIGGRAPH Asia 2010 Papers*, SIGGRAPH ASIA '10, art. no. 182, 10 pp. Association for Computing Machinery, New York, NY, USA, 2010. doi: 10.1145/1866158.1866204 2

[7] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin et al. A search engine for 3d models. 22(1):83–105, Jan. 2003. doi: 10.1145/588272.588279 2

[8] R. Girdhar, A. El-Nouby, Z. Liu, M. Singh, K. V. Alwala, A. Joulin et al. Imagebind: One embedding space to bind them all, 2023. doi: 10.48550/arXiv.2305.05665 2

[9] A. Hamdi, S. Giancola, and B. Ghanem. Mvtn: Multi-view transformation network for 3d shape recognition, 2021. doi: 10.48550/arXiv.2011.13244 2

[10] D. Hegde, J. M. J. Valanarasu, and V. M. Patel. Clip goes 3d: Leveraging prompt tuning for language grounded 3d recognition, 2023. doi: 10.48550/arXiv.2303.11313 2, 4

[11] L. A. Hendricks and A. Nematzadeh. Probing image-language transformers for verb understanding. In C. Zong, F. Xia, W. Li, and R. Navigli, eds., *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3635–3644. Association for Computational Linguistics, Online, Aug. 2021. doi: 10.18653/v1/2021.findings-acl.318 7

[12] T. Huang, B. Dong, Y. Yang, X. Huang, R. W. H. Lau, W. Ouyang et al. Clip2point: Transfer clip to point cloud classification with image-depth pre-training, 2023. doi: 10.48550/arXiv.2210.01055 2

[13] L. Höllein, A. Cao, A. Owens, J. Johnson, and M. Nießner. Text2room: Extracting textured 3d meshes from 2d text-to-image models, 2023. doi: 10.48550/arXiv.2303.11989 2

[14] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham et al. Scaling up visual and vision-language representation learning with noisy text supervision, 2021. doi: 10.48550/arXiv.2102.05918 2

[15] N. Kassner and H. Schütze. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly, 2020. doi: 10.48550/arXiv.1911.03343 7

[16] B. Li, Y. Lu, A. Ghumman, B. Strylowski, M. Gutierrez, S. Sadiq et al. 3d sketch-based 3d model retrieval. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, ICMR '15, pp. 555–558. Association for Computing Machinery, New York, NY, USA, 2015. doi: 10.1145/2671188.2749349 2

[17] H. Li, Y. Zhou, T. Tang, J. Song, Y. Zeng, M. Kampffmeyer et al. Unigs: Unified language-image-3d pretraining with gaussian splatting, 2025. doi: 10.48550/arXiv.2502.17860 2

[18] D. Liu, X. Huang, Y. Hou, Z. Wang, Z. Yin, Y. Gong et al. Uni3d-llm: Unifying point cloud perception, generation and editing with large language models, 2024. doi: 10.48550/arXiv.2402.03327 2

[19] M. Liu, R. Shi, K. Kuang, Y. Zhu, X. Li, S. Han et al. Openshape: Scaling up 3d shape representation towards open-world understanding, 2023. doi: 10.48550/arXiv.2305.10764 2, 4

[20] W. Liu, Y. Du, T. Hermans, S. Chernova, and C. Paxton. Structdiffusion: Language-guided creation of physically-valid structures using unseen objects, 2023. doi: 10.48550/arXiv.2211.04604 2

[21] T. Luo, C. Rockwell, H. Lee, and J. Johnson. Scalable 3d captioning with pretrained models, 2023. doi: 10.48550/arXiv.2306.07279 2, 4

[22] P. Min, J. Chen, and T. Funkhouser. A 2d sketch interface for a 3d model search engine. In *ACM SIGGRAPH 2002 Conference Abstracts and Applications*, SIGGRAPH '02, p. 138. Association for Computing Machinery, New York, NY, USA, 2002. doi: 10.1145/1242073.1242151 2

[23] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas et al. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding, 2018. doi: 10.48550/arXiv.1812.02713 2

[24] R. Ohbuchi, T. Minamitani, and T. Takei. Shape-similarity search of 3d models by using enhanced shape functions. In *Proceedings of Theory and Practice of Computer Graphics, 2003.*, pp. 97–104, 2003. doi: 10.1109/TPCG.2003.1206936 2

[25] Y. Pang, W. Wang, F. E. H. Tay, W. Liu, Y. Tian, and L. Yuan. Masked autoencoders for point cloud self-supervised learning, 2022. doi: 10.48550/arXiv.2203.06604 2

[26] D. Paschalidou, A. Kar, M. Shugrina, K. Kreis, A. Geiger, and S. Fidler. Atiss: Autoregressive transformers for indoor scene synthesis, 2021. doi: 10.48550/arXiv.2110.03675 2

[27] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation, 2017. doi: 10.48550/arXiv.1612.00593 2

[28] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space, 2017. doi: 10.48550/arXiv.1706.02413 2

[29] W. Qian, C. Gao, A. Sathya, R. Suzuki, and K. Nakagaki. Shape-it: Exploring text-to-shape-display for generative shape-changing behaviors with llms. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, UIST '24, art. no. 118, 29 pp. Association for Computing Machinery, New York, NY, USA, 2024. doi: 10.1145/3654777.3676348 2

[30] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal et al. Learning transferable visual models from natural language supervision, 2021. doi: 10.48550/arXiv.2103.00020 2

[31] M. Savva, A. X. Chang, and M. Agrawala. Scenesuggest: Context-driven 3d scene design, 2017. doi: 10.48550/arXiv.1703.00061 2

[32] Sketchfab. Sketchfab — The best 3D viewer on the web. `https://sketchfab.com/`, 2025. Accessed: 9 September 2025. 1, 4

[33] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition, 2015. doi: 10.48550/arXiv.1505.00880 2

[34] T. Thrush, R. Jiang, M. Bartolo, A. Singh, A. Williams, D. Kiela et al. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5238–5248, June 2022. doi: 10.48550/arXiv.2204.03162 7

[35] Unity Technologies. Unity, 2023. Game development platform. 1

[36] Y. Wang, K. Li, Y. Li, Y. He, B. Huang, Z. Zhao et al. Internvideo: General video foundation models via generative and discriminative learning, 2022. doi: 10.48550/arXiv.2212.03191 2

[37] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic graph cnn for learning on point clouds, 2019. doi: 10.48550/arXiv.1801.07829 2

[38] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang et al. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912–1920, 2015. doi: 10.48550/arXiv.1406.5670 2, 3

[39] S. Xie, J. Gu, D. Guo, C. R. Qi, L. J. Guibas, and O. Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding, 2020. doi: 10.48550/arXiv.2007.10985 2

[40] H. Xu, G. Ghosh, P.-Y. Huang, D. Okhonko, A. Aghajanyan, F. Metze et al. Videoclip: Contrastive pre-training for zero-shot video-text understanding, 2021. doi: 10.48550/arXiv.2109.14084 2

[41] L. Xue, M. Gao, C. Xing, R. Martín-Martín, J. Wu, C. Xiong et al. Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding, 2023. doi: 10.48550/arXiv.2212.05171 2, 4

[42] Y. Yang, F.-Y. Sun, L. Weihs, E. VanderBilt, A. Herrasti, W. Han et al. Holodeck: Language guided generation of 3d embodied ai environments, 2024. doi: 10.48550/arXiv.2312.09067 2

[43] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling, 2022. doi: 10.48550/arXiv.2111.14819 2

[44] M. Yuksekgonul, F. Bianchi, P. Kalluri, D. Jurafsky, and J. Zou. When and why vision-language models behave like bags-of-words, and what to do about it?, 2023. doi: 10.48550/arXiv.2210.01936 7

[45] L. Zhang, J. Pan, J. Gettig, S. Oney, and A. Guo. Vrcopilot: Authoring 3d layouts with generative ai models in vr. In *Proceedings of the 37th Annual*

ACM Symposium on User Interface Software and Technology, UIST '24, art. no. 96, 13 pp. Association for Computing Machinery, New York, NY, USA, 2024. doi: 10.1145/3654777.3676451 2

[46] R. Zhang, Z. Guo, W. Zhang, K. Li, X. Miao, B. Cui et al. Pointclip: Point cloud understanding by clip, 2021. doi: 10.48550/arXiv.2112.02413 2

[47] J. Zhou, J. Wang, B. Ma, Y.-S. Liu, T. Huang, and X. Wang. Uni3d: Exploring unified 3d representation at scale, 2023. doi: 10.48550/arXiv.2310.06773 2

[48] Q. Zhou and A. Jacobson. Thingi10k: A dataset of 10,000 3d-printing models, 2016. doi: 10.48550/arXiv.1605.04797 2

[49] B. Zhu, B. Lin, M. Ning, Y. Yan, J. Cui, H. Wang et al. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment, 2024. doi: 10.48550/arXiv.2310.01852 2, 3