MetaFind: Scene-Aware 3D Asset Retrieval for Coherent Metaverse Scene Generation

Zhenyu Pan
Northwestern University
zhenyupan@u.northwestern.edu

Yucheng Lu
New York University
yuchenglu@nyu.edu

Han Liu
Northwestern University
hanliu@northwestern.edu

Abstract

We present **MetaFind**, a scene-aware tri-modal compositional retrieval framework designed to enhance scene generation in the metaverse by retrieving 3D assets from large-scale repositories. MetaFind addresses two core challenges: (i) inconsistent asset retrieval that overlooks spatial, semantic, and stylistic constraints, and (ii) the absence of a standardized retrieval paradigm specifically tailored for 3D asset retrieval, as existing approaches mainly rely on general-purpose 3D shape representation models. Our key innovation is a flexible retrieval mechanism that supports arbitrary combinations of text, image, and 3D modalities as queries, enhancing spatial reasoning and style consistency by jointly modeling object-level features (including appearance) and scene-level layout structures. Methodologically, MetaFind introduces a plug-and-play equivariant layout encoder **ESSGNN** that captures spatial relationships and object appearance features, ensuring retrieved 3D assets are contextually and stylistically coherent with the existing scene, regardless of coordinate frame transformations. The framework supports iterative scene construction by continuously adapting retrieval results to current scene updates. Empirical evaluations demonstrate the improved spatial and stylistic consistency of MetaFind in various retrieval tasks compared to baseline methods.

1 Introduction

This work introduces MetaFind, a novel scene-aware 3D retrieval framework designed to facilitate coherent scene generation within the metaverse by retrieving 3D assets from extensive repositories. Effective scene generation heavily relies on retrieving relevant, consistent, and contextually appropriate 3D assets [26]; however, current methods face significant limitations, primarily due to two key challenges. First, existing retrieval frameworks often overlook critical factors such as spatial relationships, semantic coherence, and stylistic consistency, leading to retrieved assets that are visually and contextually incongruous when integrated into complex scenes [10]. Second, unlike well-established retrieval paradigms in natural language processing (NLP), such as Dense Passage Retrieval (DPR) [8]—which introduced a generalizable dual-encoder architecture—there is currently no standardized retrieval paradigm explicitly tailored to the requirements and characteristics of 3D asset retrieval. Finally, recent retrieval depends on generic 3D shape representation models, which fail to capture scene-specific contextual and stylistic nuances essential for coherent scene layout.

Recent approaches try to address these challenges by introducing various strategies. Early efforts enhance retrieval through 3D representations, focusing on object-level geometric features [7, 25]. Subsequent studies address cross-domain retrieval limitations through advanced techniques. Methods like SPL [27] leverage domain alignment strategies, minimizing inter-domain discrepancies. UCD [24] proposes sample-level weighting combined with domain and class alignment mechanisms, achieving improved performance but still relying on labeled data and introducing prediction bias. More recently, S2Mix [6] and SCA3D [19] introduce style fusion layers and cross-modal data augmentation techniques to enhance retrieval performance. Despite these improvements, the current approaches are

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

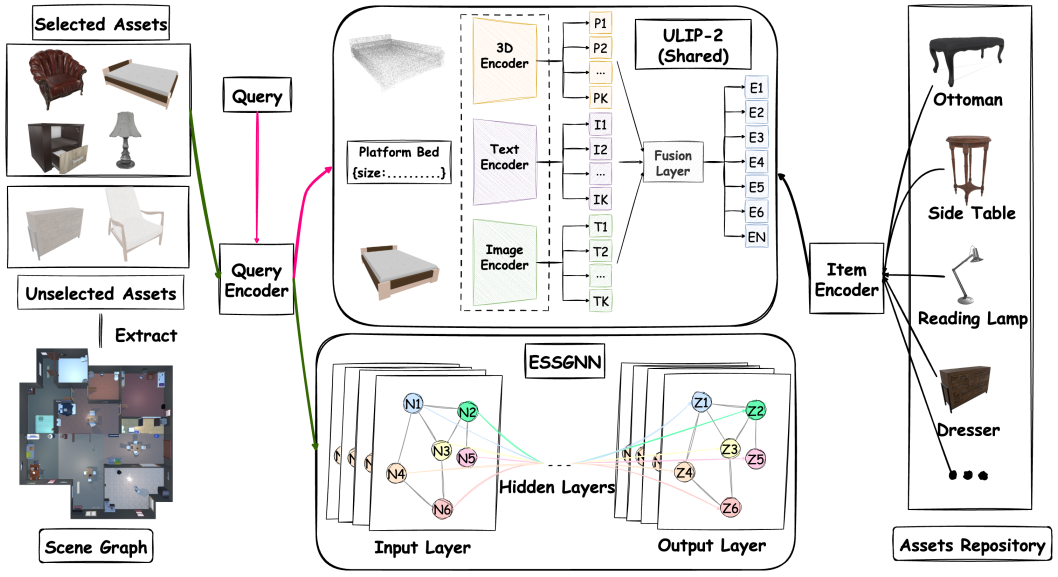


Figure 1: Overall framework. MetaFind adopts a dual-tower design where both the user query and candidate assets are encoded using the ULIP-2 backbone. On the query side, we incorporate a plug-and-play ESSGNN module that encodes the current scene layout into a structured scene graph, which captures spatial relationships and object attributes. The user's input—text, image, point cloud, or any combination—is processed by ULIP-2 and fused with the scene context embedding from the ESSGNN to produce a layout-aware query representation. On the asset side, each 3D asset in the repository is pre-encoded independently by ULIP-2 into a fixed vector. At retrieval time, the similarity between the layout-aware query embedding and the precomputed asset embeddings is computed, and the top-matching asset is selected to be inserted into the scene.

limited as they mainly consider object-centric features without adequately capturing crucial spatial, contextual, and scene-level relationships. Furthermore, they only support single-modality queries (3D-to-3D, text-to-3D, or image-to-3D), lacking the flexibility to handle compositional queries across multiple modalities. To address these limitations, MetaFind introduces a retrieval paradigm that supports compositional multi-modal queries and incorporates spatial reasoning, semantic coherence, and stylistic consistency to ensure seamless integration of retrieved 3D assets into complex scenes.

To this end, we propose MetaFind, a dual-tower retrieval framework that integrates fine-grained object-level semantics with global scene-level spatial reasoning to enable context-aware, multimodal 3D asset retrieval. Unlike prior methods that only rely on object-centric cues (images or 3D shapes or text descriptions), MetaFind incorporates the spatial background by modeling the current scene layout as a structured graph. This layout-aware design allows the retriever to reason about placement constraints, positional dependencies, and contextual fit, enhancing spatial, semantic, and stylistic consistency. Moreover, MetaFind supports flexible multimodal queries, where the input can be any combination of text, image, point cloud, and layout context. This compositional design ensures robustness under missing modality conditions and adaptability to diverse use cases, including interactive scene editing, layout-conditioned asset generation, and large-scale virtual environment construction.

As shown in Figure1, MetaFind builds upon ULIP2 [30], a tri-modal learning framework that aligns text, image, and point cloud into a shared embedding space. We adopt a dual-encoder architecture [8], where the query encoder flexibly encodes any user-provided modality combination, and the gallery encoder precomputes embeddings for all 3D assets to enable efficient retrieval. To supervise this alignment, we annotate 48K 3D assets from the Objaverse-LVIS subset [2], each rendered from 11 views and processed with GPT-4o to generate structured text descriptions. For layout-level reasoning, we introduce the Equivariant Spatial-Semantic Graph Neural Network (ESSGNN), an EGNN-based encoder designed to model rooms as graphs where nodes represent existing objects with 3D coordinates and text features and edges reflect spatial-semantic relationships. Unlike GNNs, ESSGNN maintains equivariance to rotation and translation by separating spatial and semantic

channels, ensuring that scene embeddings remain stable across coordinate shifts and alignments—an essential property for robust layout modeling in unnormalized or dynamic environments. This encoder is trained on ProcTHOR [3], which contains over 10,000 generated houses. The ESSGNN outputs a layout context vector, which is fused with the query embedding to produce a layout-aware retrieval representation. We adopt a two-stage training: (1) pretraining on object-level data for cross-modal grounding and (2) fine-tuning on room-level scenes for layout-aware adaptation. This architecture ensures strong generalization, modularity, and robustness across complex retrieval conditions.

In summary, we contribute on: (1) we present **MetaFind**, a novel layout-aware multimodal 3D asset retrieval framework tailored for coherent scene generation, which jointly considers object-level features and scene-level spatial context; (2) we introduce a plug-and-play **ESSGNN** layout encoder that models the evolving scene as a structured graph, capturing spatial relationships, contextual dependencies, and semantic attributes to guide retrieval decisions, with built-in SE(3) equivariance to prevent degradation under arbitrary scene rotations or global shifts in coordinate systems; (3) we design MetaFind to support flexible and robust multimodal querying, allowing arbitrary combinations of multi-modalities as input, enabling strong performance under diverse and incomplete input conditions; and (4) we demonstrate through comprehensive experiments that MetaFind outperforms baselines in both standard retrieval and layout-aware scene construction, and that our proposed iterative retrieval pipeline enhances contextual consistency and realism compared to current methods.

2 Methodology

In this section, we introduce the MetaFind, formalize the retrieval task, and present our dual-tower architecture with modality-aware fusion and the ESSGNN layout encoder. We describe the training strategy and the iterative scene composition process for contextually coherent 3D asset retrieval.

2.1 Task Definition

We aim to accurately retrieve contextually coherent 3D assets from a large-scale repository, given a user query and optional existing scene layout information. Formally, our retrieval task can be defined as follows: given an input query $Q = \{q_{text}, q_{img}, q_{pc}, q_{layout}\}$, which may include text q_{text}, images q_{img}, 3D point clouds q_{pc}, and optionally layout context q_{layout}, the system retrieves the asset A^* from a pre-encoded asset database $\mathcal{A}$:

$$A^* = \arg\max_{A \in \mathcal{A}} \text{sim}(f_{query}(Q), f_{gallery}(A)), \tag{1}$$

where f_{query} and $f_{gallery}$ represent the query and gallery encoders, and $\text{sim}(\cdot, \cdot)$ denotes the similarity function. The task is challenging due to the multimodal nature of user queries, partial modality absence, and the necessity for accurate layout awareness to ensure spatial coherence and realism.

2.2 Method Overview

To address the above challenge, we introduce MetaFind, as shown in 1, a dual-tower retrieval framework consisting of a query encoder and a gallery encoder, both leveraging the ULIP-2 embedding backbone. The gallery encoder precomputes embeddings for assets using three available modalities, which are then stored for efficient retrieval. On the query side, the encoder is designed to flexibly handle arbitrary combinations of modalities and, optionally, layout information—accommodating partial modality absence through a modality-aware fusion strategy. Specifically, each available modality is independently encoded using the ULIP-2 backbone, and these modality embeddings are subsequently integrated via a fusion layer, such as mean pooling, an MLP, or a Transformer-based module, generating a unified representation. Furthermore, the query encoder optionally integrates a layout encoder (ESSGNN) to capture spatial context from the existing scene layout. The layout is modeled as a structured graph with nodes representing placed objects (each described by spatial coordinates and semantic embeddings) and edges capturing spatial relationships. The layout encoder processes this graph to produce a context-aware layout vector, enhancing the spatial reasoning capability of the retrieval process. Its equivariant property ensure stable and generalizable scene embeddings under varying coordinate frames and unnormalized layouts common in open-world environments.

Our training protocol involves two stages: First, we train the query and gallery encoders to learn fundamental multimodal embedding alignment without spatial context. Subsequently, we fine-tune

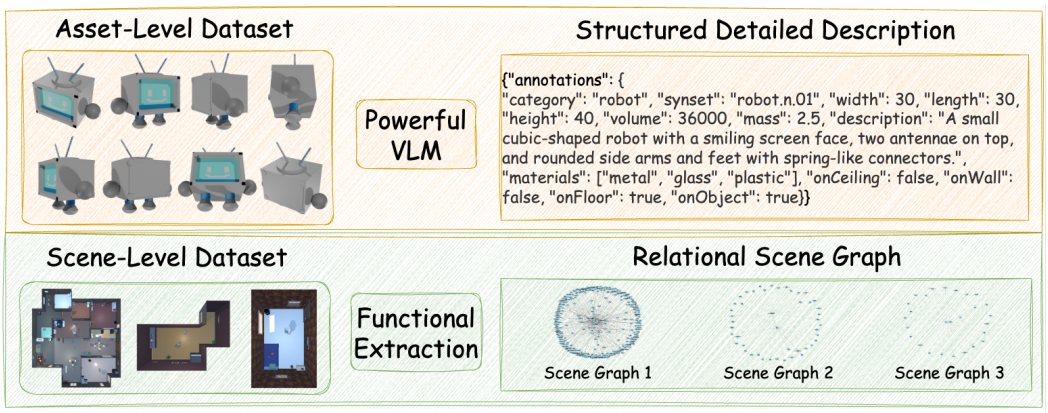


Figure 2: Data preparation pipeline. At the asset level (top), each 3D object from Objaverse-LVIS is rendered from multiple orthogonal views and passed through a VLM to generate structured, detailed annotations, capturing attributes such as category, dimensions, materials, and spatial placement constraints. At the scene-level (bottom), functional extraction is performed on generated rooms from the ProcTHOR, resulting in relational scene graphs encoding the spatial and semantic relationships between placed objects, enabling layout-aware retrieval capabilities in MetaFind.

the query encoder—particularly the fusion module and the layout encoder—using layout-aware room-level datasets. This fine-tuning stage employs adaptive freezing strategies, selectively freezing components like the gallery encoder to balance performance and computational efficiency.

2.3 Data Preparation

Our methodology requires prepared datasets at both object and scene levels to support multimodal and layout-aware retrieval tasks (as illustrated in Figure 2). For object-level representation learning, we utilize the Objaverse-LVIS dataset, which comprises approximately 48,000 distinct 3D assets. Each asset is rendered from 11 orthogonal viewpoints and annotated using GPT-4o. These annotations provide rich textual descriptions detailing attributes such as object category, size dimensions, materials, and placement constraints. For scene-level data, we leverage the ProcTHOR, which includes over 10,000 generated houses constructed from a curated collection of more than 3,000 unique assets. Each room configuration provides precise spatial coordinates and comprehensive semantic metadata for each asset, enabling the extraction of structured graphs representing object-level placements and spatial relationships. The bottom side of Figure 2 illustrates the extraction process of such structured scene graphs. These graphs form the basis for training the layout-aware ESSGNN encoder, effectively capturing spatial coherence and relational context crucial for accurate asset retrieval. They include two types of edges: (i) physical-relation edges that capture spatial dependencies (e.g., "cup on table"); and (ii) semantic-relation edges that capture functional or contextual associations (e.g., "microscope–lab bench"), obtained by prompting an LLM on object pairs. This dual-edge design encodes both physical layout and high-level semantics, enhancing retrieval and layout reasoning.

2.4 Dual-Tower Architecture and Fusion Design

While prior works typically align 3D encoders to a fixed CLIP embedding space by freezing pretrained text and image encoders, our MetaFind framework adopts a more flexible dual-tower design. It enables context-aware, multi-modal queries by training a dedicated query encoder that fuses arbitrary modality subsets—including text, image, and scene-aware 3D inputs.

MetaFind employs a dual-tower architecture with separate encoders for the query and gallery. Each tower leverages ULIP-2 to independently encode available modalities (text, images, and point clouds). A modality-aware fusion module combines these modality embeddings via one of several strategies, such as mean pooling, MLP, masked MLP, gated fusion, or Transformer-based fusion. The gallery encoder is modality-complete and frozen after pretraining, while the query encoder remains flexible: It accepts any subset of modalities and can be augmented with a layout-aware vector. This vector

is extracted using our proposed Equivariant Spatial-Semantic Graph Neural Network (ESSGNN) trained on scene graphs, enabling the model to incorporate spatial context for scene-aware retrieval.

2.5 ESSGNN: Scene-Aware Equivariant Graph Encoder

In this work, we propose the **Equivariant Spatial-Semantic Graph Neural Network (ESSGNN)** to encode 3D scene layouts in a way that is both spatially grounded and semantically expressive. ESSGNN is designed to maintain equivariance to SE(3) transformations during message passing while incorporating semantic relationships between objects through learned edge representations.

We initially experimented with standard Graph Attention Networks (GATs) to model inter-object dependencies based on spatial adjacency. However, we observed that GATs were highly sensitive to global translation and scaling variations across scenes, resulting in unstable layout embeddings and poor generalization. These issues are especially prominent in open-world or metaverse environments, where object positions are defined in large and often unnormalized coordinate systems, with no guarantee that scenes are aligned or centered.

Motivated by recent advances in drug design—where Equivariant Graph Neural Networks (EGNNs [21]) have been effectively applied to model 3D molecular structures invariant to spatial transformations—we design ESSGNN to address these limitations. Our model extends the EGNN formulation to incorporate semantic edge features in addition to geometric ones, allowing message passing to be informed not only by spatial proximity but also by functional or compositional relationships between objects. Given a scene graph $G = (\mathcal{V}, \mathcal{E})$, each node $v_i \in \mathcal{V}$ represents an object with 3D position $x_i \in \mathbb{R}^3$ and a text-derived feature $t_i \in \mathbb{R}^d$. The node feature is initialized as:

$$h_i^{(0)} = \text{Concat}(x_i, t_i).$$

Edges in the graph include both spatial and semantic relationships. Spatial edges are extracted from physical layout constraints (e.g., adjacency, support), while semantic edges are generated by prompting a large language model (LLM) with object descriptions to produce natural language relation sentences. These sentences are then encoded into dense vectors using a frozen text encoder (e.g., CLIP or BERT), resulting in edge embeddings e_{ij} that carry functional and relational meaning.

The message-passing mechanism in ESSGNN follows a modified Equivariant Graph Convolutional Layer (EGCL) structure. For each layer l, node features and positions are updated as:

$$h_i^{l+1} = h_i^l + \sum_{j \in \mathcal{N}(i)} f_h(d_{ij}^l, h_i^l, h_j^l, e_{ij}; \theta_h), \tag{2}$$

$$x_i^{l+1} = x_i^l + \sum_{j \in \mathcal{N}(i)} (x_i^l - x_j^l) \cdot f_x(d_{ij}^l, h_i^{l+1}, h_j^{l+1}, e_{ij}; \theta_x), \tag{3}$$

where $d_{ij}^l = \|x_i^l - x_j^l\|_2$ is the Euclidean distance between nodes, and $f_h : \mathbb{R}^{(2d+1+e)} \to \mathbb{R}^d$, $f_x : \mathbb{R}^{(2d+1+e)} \to \mathbb{R}^3$ are two learnable functions parameterized by θ_h and θ_x, respectively, which we approximate using multilayer perceptrons (MLPs). Here, e denotes the dimension of the semantic edge embedding e_{ij}. After L layers, the node features are aggregated into a global layout embedding:

$$e_{\text{layout}} = \text{Pooling}(\{h_i^{(L)}\}).$$

This embedding is integrated into the query encoder of our dual-tower retrieval framework to provide scene-aware conditioning. ESSGNN generalizes the original EGNN by introducing *semantic-aware edge modulation*, enabling it to operate on multi-relational graphs with heterogeneous object types and to better handle complex spatial-functional layouts found in real-world and virtual 3D scenes.

Our model retains full SE(3)-equivariance concerning input transformations. Specifically, for any rotation operator $R \in SO(3)$ and translation vector $T \in \mathbb{R}^3$, the following condition holds:

$$(Rx^{l+1} + T,\, h^{l+1}) = \text{ESSGNN}(Rx^l + T,\, h^l,\, E). \tag{4}$$

We provide a formal proof of this equivariance property in Appendix C, extending the original EGCL proof to include semantic edge features.

2.6 Training Strategy

We adopt a two-stage training strategy that aligns with the dual-tower architecture and the flexible, multimodal nature of the retrieval task. In the first stage, we focus on learning robust cross-modal representations that can handle arbitrary combinations of query modalities. In the second stage, we incorporate scene layout information through an ESSGNN encoder, enabling the system to perform context-aware retrieval grounded in spatial reasoning.

Stage 1: Cross-Modal Alignment Pretraining.

In the first stage, both query and gallery encoders are trained on large-scale object-level data from Objaverse-LVIS, where each asset has full modality inputs (text, images, and point clouds). We introduce stochastic modality masking to simulate partial-modality queries: each modality in the query has a 30% probability of being independently masked. Rather than zero-padding, we apply masked embeddings to ensure flexibility and prevent model degradation. The goal is to align all available modality combinations into a shared embedding space. The gallery encoder is trained to be modality-complete, and both towers share the contrastive retrieval objective:

$$\mathcal{L}_{\text{pre}} = -\log \frac{\exp(\text{sim}(f_{\text{query}}(Q), f_{\text{gallery}}(A))/\tau)}{\sum_{A' \in \mathcal{B}} \exp(\text{sim}(f_{\text{query}}(Q), f_{\text{gallery}}(A'))/\tau)}, \tag{5}$$

where τ is a temperature hyperparameter and $\mathcal{B}$ denotes the gallery batch.

Stage 2: Layout-Aware Fine-Tuning

In the second training stage, we enhance the query encoder with spatial context derived from the current scene layout. Given available modality embeddings for text e_{text}, image e_{img}, and point cloud e_{pc}, along with the optional layout embedding e_{layout} produced by the ESSGNN module, the final fused query representation is computed as:

$$e_{\text{query}} = \text{Fusion}(e_{\text{text}}, e_{\text{img}}, e_{\text{pc}}) + \lambda \cdot e_{\text{layout}}, \tag{6}$$

where λ is a learnable scalar controlling the contribution of layout information. This residual design allows layout reasoning to enhance retrieval without disrupting the original embedding space.

To ensure robustness in real-world settings where scene layouts may not always be available, we introduce *stochastic scene dropout* during training: the layout vector e_{layout} is omitted in 30% of batches, forcing the model to generalize to layout-free inputs. Only the query-side fusion layer and the ESSGNN module are updated during this stage; the gallery encoder is frozen to reduce training costs and preserve asset embedding consistency.

We adopt a **bidirectional contrastive learning** objective to symmetrically align query and gallery embeddings. Let e_{query} and e_{gallery} denote the fused query and gallery embeddings, respectively. The layout-aware retrieval loss is defined as:

$$\mathcal{L}^{\text{q2g}}_{\text{layout}} = -\log \frac{\exp(\text{sim}(e_{\text{query}}, e_{\text{gallery}})/\tau)}{\sum_{e'_{\text{gallery}} \in \mathcal{B}} \exp(\text{sim}(e_{\text{query}}, e'_{\text{gallery}})/\tau)}, \quad \mathcal{L}^{\text{g2q}}_{\text{layout}} = -\log \frac{\exp(\text{sim}(e_{\text{gallery}}, e_{\text{query}})/\tau)}{\sum_{e'_{\text{query}} \in \mathcal{B}} \exp(\text{sim}(e_{\text{gallery}}, e'_{\text{query}})/\tau)} \tag{7}$$

where τ is a temperature hyperparameter, and $\mathcal{B}$ denotes the batch of negatives. The final loss is the average of the two directions:

$$\mathcal{L}_{\text{layout}} = \frac{1}{2}\left(\mathcal{L}^{\text{q2g}}_{\text{layout}} + \mathcal{L}^{\text{g2q}}_{\text{layout}}\right). \tag{8}$$

This training strategy encourages accurate retrieval of relevant assets (query-to-gallery) and consistent representation of assets retrievable by matching scene context (gallery-to-query). The model improves generalization and robustness by aligning both directions, especially in iterative scene construction where queries and context evolve.

2.7 Inference and Iterative Composition

At inference time, all gallery asset embeddings are precomputed and cached for efficient retrieval. Given an input query—which may consist of any combination of text, image, point cloud, and

Algorithm 1 Iterative Layout-Aware Scene Composition

Require: Precomputed gallery embeddings $\mathcal{E}_{\text{gallery}}$, initial scene graph G_0, asset query list $\{Q_1, Q_2, ..., Q_N\}$

Initialize scene graph $G \leftarrow G_0$
for $i = 1$ to N **do**
  Extract current layout embedding: $e_{\text{layout}} \leftarrow \text{EGNN}(G)$
  Encode available modalities of query Q_i: $e_{\text{text}}, e_{\text{img}}, e_{\text{pc}}$
  Fuse into layout-aware query: $e_{\text{query}} \leftarrow \text{Fusion}(e_{\text{text}}, e_{\text{img}}, e_{\text{pc}}) + \lambda \cdot e_{\text{layout}}$
  Retrieve best-matching asset: $A_i^* \leftarrow \arg\max_{A \in \mathcal{E}_{\text{gallery}}} \text{sim}(e_{\text{query}}, e_{\text{gallery}}(A))$
  Place A_i^* into the scene, update scene graph: $G \leftarrow G \cup \{A_i^*\}$
end for
return Final composed scene G

optional scene layout—the query encoder generates a layout-aware embedding used to identify the most contextually suitable asset from the gallery.

To construct complete scenes, we deploy an iterative composition strategy shown in Algorithm 1. Instead of retrieving all required objects independently in a single step, we retrieve and place one object at a time. After each placement, the scene graph is updated to reflect the new layout, and the ESSGNN module recomputes the layout embedding, allowing subsequent retrievals to account for the evolving spatial context. While this step-by-step process introduces additional computational latency compared to one-shot parallel retrieval, it significantly improves spatial coherence and contextual alignment across placed objects, resulting in more realistic and visually harmonious scenes.

Efficiency considerations. The iterative pipeline incurs extra latency and compute versus one-shot retrieval—especially for multi-object scenes—but this trade-off is use-case dependent and tunable. When global coherence and stylistic consistency matter most, a fully sequential schedule yields the best quality. When efficiency is prioritized, we use parallel retrieval or region-based decomposition: partition a room into semantic/spatial regions (e.g., seating, storage), retrieve sequentially within each region to preserve local coherence, and process regions in parallel to improve throughput. This design flexibility makes the method practical across scenarios, and we have clarified it in the revision.

3 Experiments

We conduct comprehensive experiments to evaluate MetaFind across multiple dimensions, including object-level retrieval, scene-level layout-aware retrieval, and robustness under varying design choices. We begin by introducing our experimental setup, datasets, and baseline adaptations. We then present quantitative results on the Objaverse-LVIS dataset to assess retrieval performance under different modality combinations. Next, we evaluate scene-level quality on the ProcTHOR dataset, highlighting the benefits of layout-aware retrieval using our ESSGNN context encoder. We further perform extensive ablation studies to analyze the contribution of core architectural components and training strategies. Finally, we assess generalization across scene complexities and provide qualitative visualizations to showcase the real-world effectiveness of MetaFind.

3.1 Experimental Setup

Datasets We evaluate MetaFind across both object-level and scene-level retrieval settings. The object-level experiments are conducted on the annotated Objaverse-LVIS dataset containing 48K unique 3D assets. For scene-level layout-aware retrieval, we use the ProcTHOR-10K dataset containing over 10,000 procedurally generated house layouts constructed from over 3,000 curated 3D assets. In both datasets, we allocate 80% of the data for training and reserve the remaining 20% for testing. While our experiments currently use single-room indoor scenes, the framework is designed to generalize to open-world settings; the $\text{SE}(3)$-equivariant design specifically targets robustness to large-scale and dynamic environments.

Baselines **ULIP** [30] is a tri-modal single-tower model that aligns text, image, and point cloud modalities into a unified embedding space through joint representation learning. **OpenShape** [10] adopts a dual-tower contrastive retrieval design, supporting text-to-3D and image-to-3D retrieval

via large-scale vision-language pretraining. **SCA3D** [19] focuses on point cloud-text retrieval and improves robustness using self-augmented contrastive learning, though it lacks multi-modal query fusion capabilities. **Uni3DL** [9] and **Uni3D** [32] present unified architectures for 3D-language-image understanding, supporting multiple modalities inputs. Finally, **OmniBind** [28] offers a scalable omni-modality representation space that supports combinations of text, image, audio, and point cloud inputs, though it is not optimized for layout-aware or scene-conditioned retrieval.

Since most existing retrieval models (e.g., ULIP, OpenShape) are not designed to handle arbitrary combinations of input modalities, we limit our baselines to pre-trained single-tower encoders that support at least one of the three modalities: text, image, and point cloud. To create a fair comparison within a dual-tower retrieval paradigm, we extend each baseline by adding a simple *mean pooling layer* to aggregate available modalities, and use these fused embeddings to retrieve from a pre-encoded gallery. For completeness, we also include our own dual-tower model with a mean fusion layer but without layout context as a direct ablation baseline. The temperature is 0.5 for all experiments.

Metrics We benchmark MetaFind and all variants using standard retrieval metrics, including top-k retrieval accuracy (R@1, R@5). To assess scene-level performance, we further evaluate the compositional quality of generated scenes along two axes: structural coherence and stylistic consistency. These aspects are quantitatively scored using a GPT-4o-based aesthetic and alignment evaluator, and qualitatively validated through human preference studies conducted on a subset of generated scenes. This dual evaluation setup provides a comprehensive assessment of both retrieval accuracy and real-world usability in downstream scene construction.

3.2 Retrieval Performance on Objaverse-LVIS

We first evaluate the object-level retrieval performance on the annotated Objaverse-LVIS dataset, which comprises 48K high-quality 3D assets with structured textual descriptions and multi-view image renders. This evaluation focuses on the core capability of MetaFind to support flexible, modality-compositional retrieval, especially under partial modality conditions. All methods are evaluated under seven query conditions: text-only, image-only, point cloud-only, text+image, text+point cloud, image+point cloud, and full (text+image+point cloud). As shown in Table 1, MetaFind without ESSGNN outperforms all baseline models across different settings. Notably, since other models do not adopt a dual-tower design, their "PC only" performance reflects retrieval using identical embeddings for both query and gallery, leading to inflated accuracy. In contrast, our dual-tower framework introduces more cross-modality retrieval, which results in lower accuracy under the "PC only". Nevertheless, MetaFind demonstrates stronger performance under partial modality conditions, highlighting its capability in multimodal fusion. After integrating the ESSGNN, while the overall scene quality is improved, we observe a drop in accuracy due to the added encoded information. This reflects a temporary and explainable trade-off between object-level precision and scene-level coherence. Stage-1 pretraining on Objaverse-LVIS uses isolated assets (no layout) and no ESSGNN; Stage-2 fine-tuning introduces ESSGNN on ProcTHOR (layout-rich, different asset distribution). Although the retrieval objective is unchanged, the fusion layer becomes partially adapted to layout-conditioned features, creating a feature-attribution mismatch when evaluating on Objaverse-LVIS (which lacks layout and disables ESSGNN). A practical mitigation is to maintain two fusion heads: a layout-free head (Stage-1) and a scene-aware head (Stage-2), selected at inference by context availability. Using the Stage-1 head reproduces the “w/o ESSGNN” numbers (omitted for brevity). In our reported results, we instead explore a single shared head by freezing both encoders in Stage-2, updating only ESSGNN and the fusion, and applying stochastic scene dropout (30%) to expose the model to layout-free inputs; some accuracy loss remains due to residual attribution drift.

3.3 Scene-Level Retrieval with Layout Context

To evaluate the benefit of layout-aware retrieval in realistic scenes, we assess MetaFind on a scene generation pipeline of I-Design [1]. It can generate a 3D scene with a given room description by designing, retrieving, and arranging. In the original paper, they use OpenShape[10] to retrieve the objects. Here, we compare the performance of MetaFind with and without the ESSGNN layout encoder. No retrieval accuracy, we assess the overall quality of composed scenes using both automated and human evaluations across four key dimensions: (1) Overall Aesthetic and Atmosphere: Measures the visual appeal and mood of the composed scene; (2) Color Scheme and Material Choices: Evaluates

Table 1: Retrieval accuracy (R@1 / R@5) on Objaverse-LVIS under different query modality combinations. MetaFind consistently outperforms all baselines across both complete and incomplete query settings. '–' indicates that the method does not support the corresponding modality combination.

Method	Text Only	Image Only	PC Only	T + I	T + PC	I + PC	T + I + PC
ULIP [30]	0.1 / 0.9	0.1 / 1.3	97.9 / 99.4	0 / 0.3	33.9 / 58	22.6 / 41.6	6.4 / 15.9
OpenShape [10]	0.6 / 1.7	0.3 / 1.1	98.4 / 99.7	0 / 0.5	35.1 / 61.4	25.0 / 44.3	7.0 / 17.2
SCA3D [19]	6.9 / 10.4	–	98.1 / 99.3	–	39.7 / 65.2	–	–
Uni3DL [9]	4.5 / 9.2	–	98.5 / **99.8**	–	37.4 / 63.9	–	–
Uni3D [32]	1.7 / 3.9	1.2 / 2.5	98.3 / 99.4	0.5 / 1.1	36.3 / 63.6	26.1 / 44.8	8.2 / 19.1
OmniBind (Base)	1.2 / 2.8	0.6 / 1.4	98.3 / 99.6	0 / 0.4	34.0 / 55.9	21.5 / 38.7	5.5 / 13.8
OmniBind (Large)	2.7 / 4.0	0.9 / 1.8	98.2 / 99.3	0.1 / 0.4	35.2 / 56.7	23.4 / 40.9	6.0 / 16.7
OmniBind (Full)[28]	5.3 / 11.7	2.3 / 3.5	**99.0** / 99.7	0.5 / 1.2	37.5 / 60.8	27.5 / 46.4	11.9 / 23.4
MetaFind w/o ESSGNN	**13.8 / 23.1**	**11.7 / 19.2**	75.1 / 78.0	**17.2 / 21.8**	**44.5 / 71.3**	**45.8 / 73.1**	**51.7 / 76.5**
MetaFind w/ ESSGNN	11.3 / 21.5	10.5 / 15.9	63.2 / 66.5	15.9 / 20.3	41.2 / 68.8	42.0 / 70.4	48.2 / 74.9

consistency in textures, colors, and materials between newly retrieved assets and the existing scene; (3) Scene Coherence: Assesses how well the inserted assets align with the scene's spatial and semantic context; and (4) Realism and 3D Geometric Consistency: Checks for physically plausible placements, avoiding collisions or unnatural geometry. Each dimension is rated on a scale from 1 (poor) to 5 (excellent), independently by GPT-4o and five expert human annotators on a set of 200 randomly sampled scenes. For GPT-4o, we provide scene layouts and rendered views, with prompts aligned to the respective evaluation criteria. Final scores are averaged across annotators and samples.

Table 2: Scene-level quality comparison across four evaluation dimensions. MetaFind (with GSSNN) achieves the highest scores across both GPT-4o and human evaluations, demonstrating superior spatial coherence and aesthetic quality in composed scenes.

Method	Aesthetic		Color & Material		Scene Coherence		Realism & Geometry	
	GPT-4o	Human	GPT-4o	Human	GPT-4o	Human	GPT-4o	Human
ULIP [30]	2.91	3.02	2.84	2.97	2.76	2.89	2.70	2.81
OpenShape [10]	3.14	3.28	3.08	3.19	3.01	3.11	2.95	3.06
MetaFind w/o ESSGNN	3.42	3.55	3.31	3.41	3.26	3.33	3.22	3.30
MetaFind w/ ESSGNN	**4.13**	**4.25**	**4.04**	**4.17**	**4.10**	**4.21**	**4.06**	**4.18**

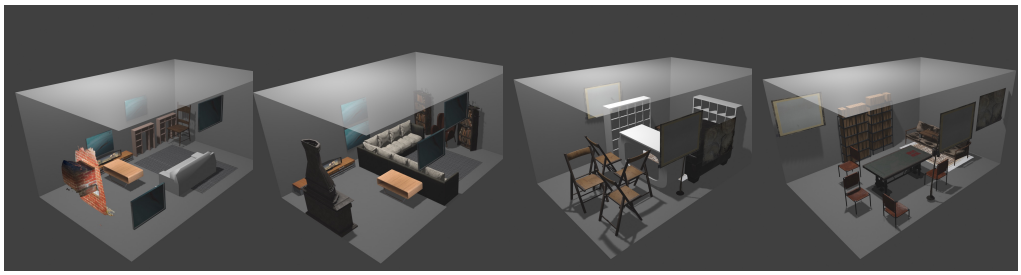
Figures 3, 4 show qualitative comparisons of scene generation with and without the ESSGNN encoder. The first example is a classical-style lounge, which, without ESSGNN, suffers from inconsistent object styles and poor layout organization. With ESSGNN, the scene is more coherent, with well-aligned furniture and logical arrangement for group interaction. The second example is an aged archive room. Without ESSGNN, the objects appear mismatched, while the ESSGNN-generated version offers a more functional and visually consistent space, with well-placed furniture suitable for a reading environment. These results demonstrate that ESSGNN improves both stylistic consistency and layout functionality. This qualitative improvement is also reflected in the quantitative results shown in Table 2, where MetaFind with ESSGNN achieves the highest scores across all evaluation metrics. In particular, the gains in scene coherence and realism highlight the encoder's ability to model spatial relationships and stylistic alignment effectively. Together, these findings confirm the effectiveness of ESSGNN in generating high-quality, semantically grounded 3D scenes.

3.4 Ablation Studies

We conduct ablation studies to evaluate the effectiveness of key architectural components and training strategies in MetaFind, focusing on six dimensions: layout encoding, modality fusion strategies, modality dropout robustness, fusion granularity, gallery encoder flexibility, and missing modality handling. First, removing the ESSGNN layout encoder results in drops in scene realism, underscoring the critical role of spatial context. Regarding fusion strategies, while simple mean pooling offers computational efficiency, MLP and the final selected Transformer outperform others under partial modality conditions by dynamically reweighting available inputs. We also examine modality dropout rates during training, finding that a 30% rate strikes the best balance between robustness and accuracy. Lower rates lead to overfitting on full-modality inputs, whereas higher rates introduce instability. Additionally, we compare fusion granularity strategies, revealing that while training only the fusion

Table 3: Ablation study (Text Only). We report top-1 retrieval accuracy (R@1) on the Object-level task, GPT-4o-based aesthetic score, and scene-level coherence score on the Scene-Level task.

Variant	R@1 (%)	Aesthetic (GPT-4o)	Scene Coherence (GPT-4o)
MetaFind (Full, bidirectional) w/ iterative retrieval & ESSGNN	11.4	**4.1**	**4.2**
w/o iterative retrieval	11.3	<u>4.0</u>	<u>4.1</u>
w/o Layout Context	**13.5**	3.4	3.3
w/ Layout Context (GAT)	11.0	3.4	3.7
Fusion = Mean	9.4	3.2	3.5
Fusion = MLPs	9.9	3.3	3.5
Modality Dropout = 10%	7.3	3.4	3.5
Modality Dropout = 50%	<u>13.2</u>	3.1	3.2
Train fuser only	8.7	3.3	3.2
Padding missing modalities with 0	10.5	3.1	3.1



Room 1 Without ESSGNN | Room 1 With ESSGNN | Room 2 Without ESSGNN | Room 2 With ESSGNN

Figure 3: Visual comparison of scene generation with and without the ESSGNN encoder across two room descriptions. Room 1 — "A classical-style lounge for group leisure and conversation"; Room 2 — “An aged archive room for research and consultation”

module in the query encoder improves efficiency, full encoder fine-tuning yields better performance by allowing earlier layers to adapt to modality-aware supervision. Finally, in handling missing modalities, modality masking outperforms zero-padding by preventing zero embedding interference and promoting robustness through sparsity-aware fusion. Results across these ablations, summarized in Table 3, demonstrate the modularity and resilience of MetaFind under diverse design choices.

4 Summary, Limitation, and Future Work

In this work, we present **MetaFind**, a scene-aware, multimodal 3D asset retrieval framework that unifies object-level semantics and scene-level spatial reasoning through a dual-tower design and a plug-and-play ESSGNN layout encoder. MetaFind demonstrates strong retrieval performance across both complete and partial modality settings, and significantly improves scene coherence and realism in iterative composition tasks. However, asset annotations rely on GPT-4o, which can introduce language bias, hallucinations, and occasional mislabeling (e.g., culturally skewed terms or incorrect attributes), potentially affecting training and evaluation. This work does not explicitly debias these annotations. Looking forward, we plan to extend MetaFind by incorporating real-world human-in-the-loop feedback for adaptive scene refinement, and scaling to open-world settings with dynamic object catalogs and evolving scene goals.

5 Acknowledgments

We gratefully acknowledge support from the NVIDIA Academic Grant ("Interactive Spatial Reasoning and 3D Scene Generation with RL-Enhanced VLMs") and the provision of cloud computing resources, which enabled systematic training and evaluation of our MetaFind and other baselines. This paper is a core component of that project. The views expressed are those of the authors and do not necessarily reflect those of NVIDIA.

References

[1] Ata Çelen, Guo Han, Konrad Schindler, Luc Van Gool, Iro Armeni, Anton Obukhov, and Xi Wang. I-design: Personalized llm interior designer. *arXiv preprint arXiv:2404.02838*, 2024.

[2] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13142–13153, 2023.

[3] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. procthor: Large-scale embodied ai using procedural generation. *Advances in Neural Information Processing Systems*, 35:5982–5994, 2022.

[4] Chuan Fang, Yuan Dong, Kunming Luo, Xiaotao Hu, Rakesh Shrestha, and Ping Tan. Ctrl-room: controllable text-to-3d room meshes generation with layout constraints. *arXiv preprint arXiv:2310.03602*, 2023.

[5] Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. Layoutgpt: Compositional visual planning and generation with large language models. *Advances in Neural Information Processing Systems*, 36:18225–18250, 2023.

[6] Xinwei Fu, Dan Song, Yue Yang, Yuyi Zhang, and Bo Wang. S2mix: Style and semantic mix for cross-domain 3d model retrieval. *Journal of Visual Communication and Image Representation*, 107:104390, 2025.

[7] Haiyun Guo, Jinqiao Wang, Min Xu, Zheng-Jun Zha, and Hanqing Lu. Learning multi-view deep features for small object retrieval in surveillance scenarios. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 859–862, 2015.

[8] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781, 2020.

[9] Xiang Li, Jian Ding, Zhaoyang Chen, and Mohamed Elhoseiny. Uni3dl: Unified model for 3d and language understanding. *arXiv preprint arXiv:2312.03026*, 2023.

[10] Minghua Liu, Ruoxi Shi, Kaiming Kuang, Yinhao Zhu, Xuanlin Li, Shizhong Han, Hong Cai, Fatih Porikli, and Hao Su. Openshape: Scaling up 3d shape representation towards open-world understanding. *Advances in neural information processing systems*, 36:44860–44879, 2023.

[11] Zhenyu Pan, Rongyu Cao, Yongchang Cao, Yingwei Ma, Binhua Li, Fei Huang, Han Liu, and Yongbin Li. Codev-bench: How do llms understand developer-centric code completion?, 2024.

[12] Zhenyu Pan and Han Liu. Metaspatial: Reinforcing 3d spatial reasoning in vlms for the metaverse. *arXiv preprint arXiv:2503.18470*, 2025.

[13] Zhenyu Pan, Haozheng Luo, Manling Li, and Han Liu. Chain-of-action: Faithful and multimodal question answering through large language models. *arXiv preprint arXiv:2403.17359*, 2024.

[14] Zhenyu Pan, Haozheng Luo, Manling Li, and Han Liu. Conv-coa: Improving open-domain question answering in large language models via conversational chain-of-action, 2024.

[15] Zhenyu Pan, Xuefeng Song, Yunkun Wang, Rongyu Cao, Binhua Li, Yongbin Li, and Han Liu. Do code llms understand design patterns? *arXiv preprint arXiv:2501.04835*, 2025.

[16] Zhenyu Pan, Yiting Zhang, Zhuo Liu, Yolo Yunlong Tang, Zeliang Zhang, Haozheng Luo, Yuwei Han, Jianshu Zhang, Dennis Wu, Hong-Yu Chen, Haoran Lu, Haoyang Fang, Manling Li, Chenliang Xu, Philip S. Yu, and Han Liu. Advevo-marl: Shaping internalized safety through adversarial co-evolution in multi-agent reinforcement learning, 2025.

[17] Zhenyu Pan, Yiting Zhang, Yutong Zhang, Jianshu Zhang, Haozheng Luo, Yuwei Han, Dennis Wu, Hong-Yu Chen, Philip S. Yu, Manling Li, and Han Liu. Evo-marl: Co-evolutionary multi-agent reinforcement learning for internalized safety, 2025.

[18] Zhenyu Pan, Yutong Zhang, Jianshu Zhang, Haoran Lu, Haozheng Luo, Yuwei Han, Philip S. Yu, Manling Li, and Han Liu. Fairreason: Balancing reasoning and social bias in mllms, 2025.

[19] Junlong Ren, Hao Wu, Hui Xiong, and Hao Wang. Sca3d: Enhancing cross-modal 3d retrieval via 3d shape and caption paired data augmentation. *arXiv preprint arXiv:2502.19128*, 2025.

[20] Aditya Sanghi, Hang Chu, Joseph G Lambourne, Ye Wang, Chin-Yi Cheng, Marco Fumero, and Kamal Rahimi Malekshan. Clip-forge: Towards zero-shot text-to-shape generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18603–18613, 2022.

[21] Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.

[22] Jonas Schult, Sam Tsai, Lukas Höllein, Bichen Wu, Jialiang Wang, Chih-Yao Ma, Kunpeng Li, Xiaofang Wang, Felix Wimbauer, Zijian He, et al. Controlroom3d: Room generation using semantic proxy rooms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6201–6210, 2024.

[23] Erzhuo Shao, Yifang Wang, Yifan Qian, Zhenyu Pan, Han Liu, and Dashun Wang. Sciscigpt: Advancing human-ai collaboration in the science of science. *arXiv preprint arXiv:2504.05559*, 2025.

[24] Dan Song, Tian-Bao Li, Wen-Hui Li, Wei-Zhi Nie, Wu Liu, and An-An Liu. Universal cross-domain 3d model retrieval. *IEEE Transactions on Multimedia*, 23:2721–2731, 2020.

[25] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015.

[26] Fan-Yun Sun, Weiyu Liu, Siyi Gu, Dylan Lim, Goutam Bhat, Federico Tombari, Manling Li, Nick Haber, and Jiajun Wu. Layoutvlm: Differentiable optimization of 3d layout via vision-language models. *arXiv preprint arXiv:2412.02193*, 2024.

[27] Qian Wang and Toby Breckon. Unsupervised domain adaptation via structured prediction based selective pseudo-labeling. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 6243–6250, 2020.

[28] Zehan Wang, Ziang Zhang, Hang Zhang, Luping Liu, Rongjie Huang, Xize Cheng, Hengshuang Zhao, and Zhou Zhao. Omnibind: Large-scale omni multimodal representation via binding spaces, 2024.

[29] Hao Wu, Ruochong Li, Hao Wang, and Hui Xiong. Com3d: Leveraging cross-view correspondence and cross-modal mining for 3d retrieval. In *2024 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2024.

[30] Le Xue, Ning Yu, Shu Zhang, Artemis Panagopoulou, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos 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*, pages 27091–27101, 2024.

[31] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8552–8562, 2022.

[32] Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3d: Exploring unified 3d representation at scale. *arXiv preprint arXiv:2310.06773*, 2023.

Appendix

A Borader Impacts **14**

B Related Work **14**
B.1 3D Scene Generation Paradigms 14
B.2 3D Object Retrieval 15

C Equivariance Proof of ESSGNN - Extension to Semantic Embedding **15**

D Experimental Analysis **17**

A Borader Impacts

MetaFind facilitates accessible and coherent 3D scene generation, which can benefit fields like virtual reality, education, and game design. By supporting flexible multimodal queries, it lowers the barrier for non-experts to build rich virtual environments. However, risks include potential misuse in generating misleading content, propagation of bias from training data, and intellectual property concerns tied to retrieved assets. We recommend responsible dataset curation and human oversight to ensure ethical deployment.

B Related Work

3D scene generation serves as the broader task context of our work, encompassing both generative and retrieval-based approaches to assembling realistic virtual environments. Within this paradigm, 3D object retrieval plays a critical role by providing high-quality assets that satisfy semantic, stylistic, and spatial constraints. We first review recent advances in scene generation frameworks, followed by an overview of representative models for multimodal 3D object retrieval.

B.1 3D Scene Generation Paradigms

Recent progress in 3D scene generation follows two directions. The first relies on generative models that synthesize entire 3D scenes in mesh, voxel, or neural field formats [22]. While promising, these methods struggle with ensuring object-level realism or semantic fidelity [4]. To address these limitations, a second paradigm emerges that frames scene generation as a layout composition task using retrieved assets from large-scale 3D repositories. LLMs and VLMs exhibit advanced capabilities in various tasks [17, 16, 18]: software engineering [15, 11], question answering systems [13, 14], and scientific discovery [23]. Methods like LayoutGPT [5] and I-Design [1] employ LLMs as planners to generate layouts from text descriptions. More recent techniques, such as LayoutVLM [26], improve physical plausibility through differentiable rendering optimization and layout supervision from image-marked datasets. Despite their advances, they still face two fundamental challenges: (1) limited internalized 3D spatial reasoning within VLMs and (2) the inefficiency and poor generalization of supervised fine-tuning, which relies on scarce and imperfect layout annotations. MetaSpatial [12] addresses these issues via a reinforcement learning-based framework that optimizes 3D spatial layouts in real time using physics-aware constraints and rendered-image evaluations. This significantly enhances scene plausibility and coherence.

While MetaSpatial focuses on improving reasoning in layout generation, another crucial but underexplored dimension is the design of the retrieval mechanism itself. Most prior works rely on general-purpose models, such as OpenShape [10], to fetch 3D assets. However, these models are not specifically trained for multimodal, scene-conditioned retrieval. They struggle to support arbitrary combinations of user inputs (e.g., missing modality scenarios) and treat object retrieval as an independent parallel process, neglecting layout dependencies. To bridge this gap, we propose a retrieval-centric framework that explicitly incorporates layout context into the retrieval loop. Unlike prior work, our method supports arbitrary modality combinations, performs iterative context-aware retrieval, and introduces a plug-and-play ESSGNN module to encode scene layout as a structured graph. This enables spatially consistent and stylistically coherent scene construction.

B.2 3D Object Retrieval

3D object retrieval has traditionally focused on aligning visual and geometric representations of objects with semantic queries in the form of text, image, or point cloud inputs. Early approaches rely on contrastive learning between 2D/3D pairs, such as PointCLIP [31] and CLIP-Forge [20], which repurpose vision-language models for shape retrieval. More recent methods like ULIP [30] and OpenShape [10] extend this to tri-modal alignment, embedding text, image, and 3D point clouds into a unified latent space via either single-tower or dual-tower architectures. However, these models are trained purely on object-centric data and assume complete modality availability, limiting their robustness under missing or partial query inputs. Beyond alignment, retrieval models such as SCA3D [19] and COM3D [29] improve representation quality via self-augmentation or compositional reasoning, yet still lack explicit mechanisms to handle arbitrary modality combinations or incorporate contextual cues. OmniBind [28] offers more flexible modality binding but is not optimized for retrieval tasks involving spatial constraints. In contrast, MetaFind is explicitly designed for context-aware, multimodal 3D asset retrieval. Our model supports free-form modality combinations and is robust to missing inputs through stochastic masking. Most notably, it augments retrieval with scene context by incorporating an ESSGNN-based layout encoder, enabling iterative, layout-aware asset selection that better supports spatial realism and scene consistency.

C Equivariance Proof of ESSGNN - Extension to Semantic Embedding

In this section, we prove that our ESSGNN maintains SE(3) equivariance in 3D space. While the original EGNN [21] formulation allows the inclusion of edge features in the message function, these are typically discrete, task-specific features such as bond types or edge labels. In contrast, our ESSGNN introduces edge embeddings e_{ij} derived from LLM-generated natural language relation descriptions, which are subsequently encoded via a frozen text encoder. Importantly, these semantic edge embeddings are *invariant to the input node positions* x, as they are computed solely from object-level text descriptions and do not depend on spatial coordinates. Therefore, although the semantics encoded in e_{ij} are richer and more expressive, the mathematical property required for equivariance—the independence of e_{ij} from x—remains satisfied. As a result, the message and update equations remain SE(3)-equivariant under our semantic extension, and the original proof structure holds. We now restate and extend the proof below.

Specifically, we show that for any translation vector $g \in \mathbb{R}^3$ and any orthogonal transformation $Q \in \mathbb{R}^{3\times 3}$, the model satisfies:

$$Q\mathrm{x}^{\mathrm{l}+1} + \mathrm{g},\ \mathrm{h}^{\mathrm{l}+1} = \mathrm{ESSGNN}(\mathrm{Q}\mathrm{x}^{\mathrm{l}} + \mathrm{g},\ \mathrm{h}^{\mathrm{l}},\ \mathrm{E}) \tag{9}$$

where x^{l} and h^{l} are the positions and features of all nodes at layer l, and E contains edge features including learned semantic embeddings e_{ij}. We begin by assuming that h^0 is invariant to SE(3) transformations on x, and that semantic edge embeddings e_{ij} are derived solely from object-level textual descriptions and thus independent of spatial coordinates. Under these assumptions, the edge message computation remains SE(3) invariant. Let us denote the pairwise edge message as:

$$\mathrm{m}_{\mathrm{ij}} = \phi_{\mathrm{e}}\left(\mathrm{h}_{\mathrm{i}}^{\mathrm{l}}, \mathrm{h}_{\mathrm{j}}^{\mathrm{l}}, \|\mathrm{x}_{\mathrm{i}}^{\mathrm{l}} - \mathrm{x}_{\mathrm{j}}^{\mathrm{l}}\|^2, \mathrm{e}_{\mathrm{ij}}\right) \tag{10}$$

Now consider a translation and rotation of all node positions: $\mathrm{x}_{\mathrm{i}}^{\mathrm{l}} \mapsto \mathrm{Q}\mathrm{x}_{\mathrm{i}}^{\mathrm{l}} + \mathrm{g}$. The Euclidean distance term becomes:

$$\|Q\mathrm{x}_{\mathrm{i}}^{\mathrm{l}} + \mathrm{g} - (\mathrm{Q}\mathrm{x}_{\mathrm{j}}^{\mathrm{l}} + \mathrm{g})\|^2 = \|\mathrm{Q}(\mathrm{x}_{\mathrm{i}}^{\mathrm{l}} - \mathrm{x}_{\mathrm{j}}^{\mathrm{l}})\|^2 = \|\mathrm{x}_{\mathrm{i}}^{\mathrm{l}} - \mathrm{x}_{\mathrm{j}}^{\mathrm{l}}\|^2 \tag{11}$$

Hence the edge message is preserved:

$$\mathrm{m}'_{\mathrm{ij}} = \phi_{\mathrm{e}}\left(\mathrm{h}_{\mathrm{i}}^{\mathrm{l}}, \mathrm{h}_{\mathrm{j}}^{\mathrm{l}}, \|\mathrm{Q}\mathrm{x}_{\mathrm{i}}^{\mathrm{l}} + \mathrm{g} - \mathrm{Q}\mathrm{x}_{\mathrm{j}}^{\mathrm{l}} - \mathrm{g}\|^2, \mathrm{e}_{\mathrm{ij}}\right) = \mathrm{m}_{\mathrm{ij}} \tag{12}$$

The position update in ESSGNN (adapted from EGNN) is defined as:

$$\mathrm{x_i^{l+1}} = \mathrm{x_i^l} + \sum_{\mathrm{j}\neq\mathrm{i}} (\mathrm{x_i^l} - \mathrm{x_j^l}) \cdot \phi_{\mathrm{x}}(\mathrm{m_{ij}}) \tag{13}$$

We now show that this equation is SE(3) equivariant. Applying the transformation:

$$\begin{aligned} Q\mathrm{x_i^l} + \mathrm{g} + \sum_{\mathrm{j}\neq\mathrm{i}} \left(\mathrm{Qx_i^l} + \mathrm{g} - \mathrm{Qx_j^l} - \mathrm{g}\right) \cdot \phi_{\mathrm{x}}(\mathrm{m_{ij}}) &= Q\mathrm{x_i^l} + \mathrm{g} + \mathrm{Q} \sum_{\mathrm{j}\neq\mathrm{i}} (\mathrm{x_i^l} - \mathrm{x_j^l}) \cdot \phi_{\mathrm{x}}(\mathrm{m_{ij}}) \\ &= Q \left(\mathrm{x_i^l} + \sum_{\mathrm{j}\neq\mathrm{i}} (\mathrm{x_i^l} - \mathrm{x_j^l}) \cdot \phi_{\mathrm{x}}(\mathrm{m_{ij}}) \right) + g \\ &= Q\mathrm{x_i^{l+1}} + \mathrm{g} \end{aligned}$$

Thus, the coordinate update is SE(3) equivariant.

For the feature update:

$$\mathrm{h_i^{l+1}} = \mathrm{h_i^l} + \sum_{\mathrm{j}\neq\mathrm{i}} \phi_{\mathrm{h}}(\mathrm{m_{ij}}) \tag{14}$$

Since $\mathrm{m_{ij}}$ is invariant to transformations of x, and both $\mathrm{h_i^l}$, $\mathrm{h_j^l}$ and e_{ij} are independent of the global pose, the feature update is invariant to SE(3) transformations of positions.

Therefore, the ESSGNN update satisfies:

$$Q\mathrm{x^{l+1}} + \mathrm{g},\ \mathrm{h^{l+1}} = \mathrm{ESSGNN}(\mathrm{Qx^l} + \mathrm{g},\ \mathrm{h^l},\ \mathrm{E}) \tag{15}$$

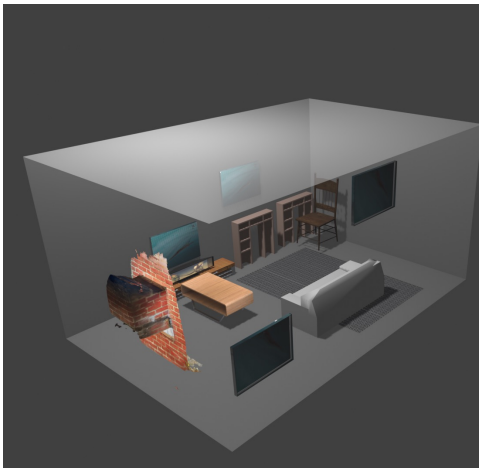
This completes the proof that ESSGNN preserves SE(3) equivariance despite the inclusion of semantic edge embeddings.

D Experimental Analysis

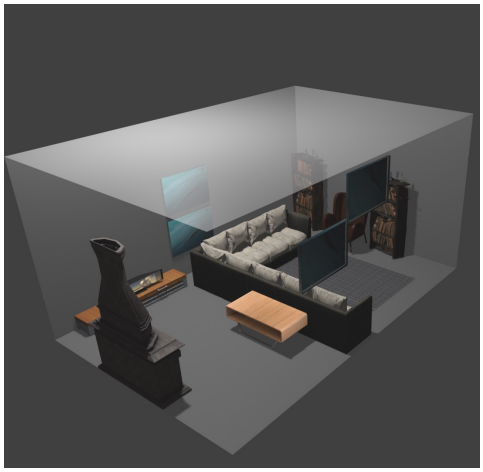
As shown in Figure 4:

Room 1: Without ESSGNN, the room lacks stylistic coherence—the metallic fireplace and mismatched furniture deviate from the classical theme. With ESSGNN, the scene adopts a unified classical aesthetic with a dark-toned fireplace, matching sofa, and bookshelf.

Room 2: Without ESSGNN, modern office furniture and cluttered seating break the archive theme and hinder functionality. With ESSGNN, compact wooden chairs are arranged around the table, better fitting the aged archive context and improving usability.



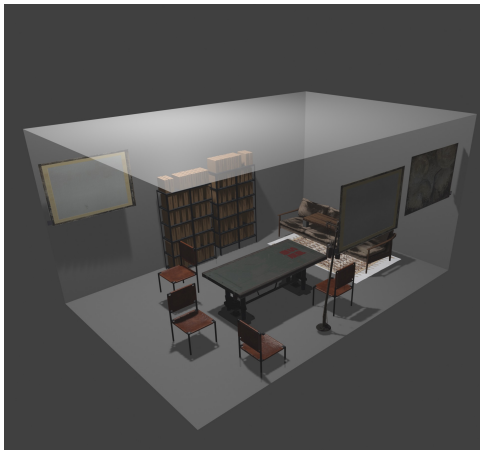
(a) Without ESSGNN encoder



(b) With ESSGNN encoder

Room 1 Description: A classical-style lounge for group leisure and conversation

(c) Without ESSGNN encoder



(d) With ESSGNN encoder

Room 2 Description: An aged archive room for research and consultation

Figure 4: Comparison of scene generation with and without ESSGNN encoder.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Section 1

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 4

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Appendix C

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 2

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We have already released a simplified version of the framework. For the final version, we plan to build a startup based on it; therefore, we do not intend to provide open access at this time.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 3

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Section 3

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 3

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper follows the Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Section A

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [No]

    Justification: We don't have any risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: In references, we cite all of them.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Appendix 5

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We are not crowd sourcing.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We are not including humans.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Appendix 5

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.