ALCER3D: Adaptive Learning Constraints for Enhanced Retrieval of Complex Indoor 3D Scenarios

Alex Falcon, Ali Abdari, Giuseppe Serra

***Abstract*—The Metaverse is growing rapidly, resulting in thousands of rich virtual universes. This results in a difficult search process for the user, making advanced search tools a necessity. Existing methods leverage contrastive learning to obtain a function mapping a 3D scene and its textual descriptions into similar representations. However, Metaverse scenarios are complex, multimedia-rich 3D scenes containing many elements, making cross-modal alignment difficult. For instance, a museum dedicated to Van Gogh is unrelated to Warhol, yet it shares similarities with Matisse or Monet. To make the mapping functions aware of these nuances, we propose a novel learning strategy to integrate Adaptive Optimization Constraints, computing data-dependent distances using a language-based method we design and enforcing them between the representations at training time. This novelty sets our approach apart from standard procedures enforcing the same distance. We validate the effectiveness of two datasets, one including 6000 apartments, and a novel dataset of 3000 museums that we collect. We observe consistent improvements compared to existing methods. Moreover, we obtain better generalization when with very complex scenarios, e.g. on the museums dataset it obtains an average R@1 of 5.2% compared to 1.2% obtained by existing methods. Finally, the source code is available at https://github.com/aliabdari/ALCER3D.**

***Index Terms*—Text-Metaverse Retrieval, Metaverse, Complex 3D scenes, Contrastive learning, Custom loss function.**

I. Introduction

THE concept of Metaverse traces back to the 90s and the novel *Snow Crash* written by Neal Stephenson, in which it represents a virtual three-dimensional world populated by avatars, controlled either by real users or by system programs [1]. Nowadays, the Metaverse and the boundaries of reality are pushed by several tech leaders, bringing increasingly more investments and attention towards Extended Reality (XR) technologies, while aiming to create immersive experiences and increase the interactivity with the augmented (or fully virtual) environment [2]. This resulted in Metaverses created for many different applications, related to shopping [3], [4], education [5], [6], and industry [7]. Given the ever growing amount of Metaverse scenarios and the difficulty for users to efficiently find those they are interested in, it becomes urgent to implement effective tools to support the users in filtering the Metaverses based on their current needs by formulating them in a textual query.

The Metaverses are highly complex 3D scenarios containing many 3D objects, such as furniture in apartments, and multimedia elements, such as paintings in digital museums. Historically, the closest approach to automating the processing of Metaverses focused on individual 3D objects. The retrieval of 3D objects guided by textual queries has attracted ample interest in research, leading to the implementation of retrieval systems for 3D objects [8]–[12]. However, this branch of research has focused on the retrieval of *single* 3D objects. Following this direction, some works started to deal with textual-guided retrieval of Metaverses, i.e. *complex and rich 3D scenarios* composed of *multiple* 3D objects, leveraging contrastive loss functions to realize systems using text to retrieve indoor scenarios with one or more furnished rooms [13]–[15], and multimedia-rich 3D scenarios [16], [17].

Contrastive loss functions [18]–[21] are commonly used in state-of-the-art approaches to implement cross-modal retrieval. By using them, it is possible to learn a joint embedding space and the respective functions mapping elements coming from different modalities into the same space. In the context of text-Metaverse retrieval, the data used to perform such a training process consist of pairs made of the Metaverse scenario and the textual description of its contents. Then, the similarity of the representations learned by the two functions is maximized. In this way, the mapping function for the textual descriptions can be used to map user queries, enabling an efficient retrieval of the desired Metaverses.

However, we identify an important shortcoming in this approach: while maximizing the similarity of the representations of Metaverse-description pairs, the other pairs are always treated as completely unrelated with nothing being shared between them. Yet, this can lead to contradictions, as Metaverses represent complex and heterogeneous scenes, hence it is possible that they have some similarities with each other. For instance, consider the apartments in Figure 1. The first two, (a) and (b), are very similar in terms of rooms, with the only difference represented by (a) having separate dining and living rooms. In comparison with (c) and (d), such

Manuscript received April 19, 2021; revised August 16, 2021.

This work was supported by MUR Progetti di Ricerca di Rilevante Interesse Nazionale (PRIN) 2022 “Multimedia Understanding meets Social Media Analysis” (code 2022YTE579, CUP G53D23002930006) funded by EU - Next-Generation EU - M4 C2 I1.1, and by the Department Strategic Plan (PSD) of the University of Udine – Interdepartmental Project on Artificial Intelligence (2020-25).

Ali Abdari is with University of Udine, Italy and University of Naples Federico II, Italy (email: abdari.ali@spes.uniud.it). Alex Falcon and Giuseppe Serra are with University of Udine, Italy. (email: falcon.alex@spes.uniud.it; giuseppe.serra@uniud.it).

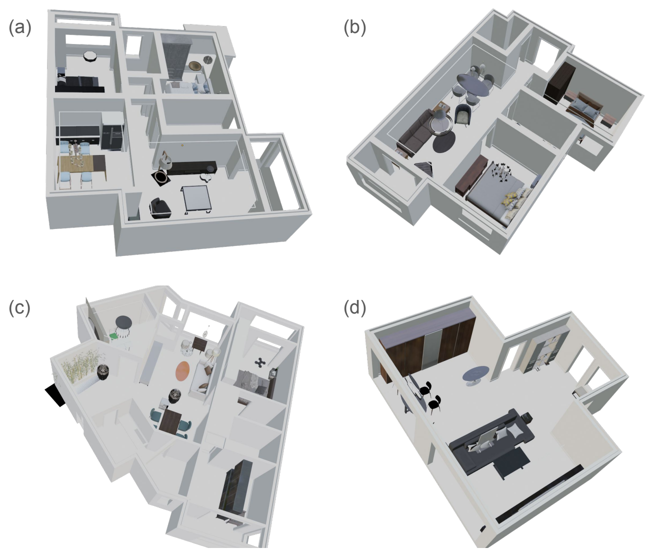


Fig. 1. Four complex 3D scenes found within the 3D-FRONT dataset [22].

a difference is negligible, since (c) only has one bedroom, and includes an outer space with few plants, whereas (d) is a simple studio. Yet, using a standard contrastive learning framework, given the apartment (a), all the other ones—(b), (c), and (d)—are considered equally different and the learning constraints lead to entirely different representations for all of them. Hence, these differences are not given the right importance at training time, and the trained model may fail at catching them.

To overcome this limitation, in this paper we introduce a *novel contrastive learning framework* designed for text-Metaverse retrieval which makes use of this observation. Specifically, given the richness of the Metaverses and the wide range of possible variations between any of them, we use their textual descriptions as a proxy for measuring the difference and quantify the desired distance to be enforced among them in the embedding space. To deal with these textual descriptions, usually very long because of the abundance of objects and multimedia elements contained in the Metaverses, we devise a method which analyzes the descriptions through multiple models and makes a decision by collecting the individual opinions. This results in highly adaptive optimization constraints which are not pursued in standard frameworks.

In summary, the important research question investigated in this paper is the following: *does enforcing customized learning constraints help obtaining better generalization?* We investigate this question on two datasets. The first contains more than 6000 furnished apartments annotated with descriptions describing the structure of the apartment (number and type of rooms) and the furniture in detail. We create the second dataset by populating it with 3000 museums, establishing the first step towards the analysis of museums in the Metaverse. Each of them is annotated by a very long description, covering the details of the paintings in each room in the museum. The experimental results confirm our hypothesis and we observe a relative improvement of 24.5% and 21.5% in median rank on Apartments and Museums datasets, respectively. Further experiments are made to support and verify the robustness of our design decisions.

The contributions of this paper can be summarized as follows:

- Given a Metaverse scenario and its description, current methodologies learn a joint embedding space through optimization constraints which distance all the other samples equally from it [13], [17]. Considering the complexity and richness of Metaverse scenarios, we highlight this procedure as flawed when it comes to such a complex type of data. Therefore, we propose a general training methodology to overcome it by creating optimization constraints which adapt to the data at hand.
- Due to the inherent complexity of Metaverses, the descriptions detailing the contents of the scenarios are also fairly complex and long. Therefore, we implement a method involving multiple language models to collectively analyze the descriptions and make decisions in relation to the proposed constraints.
- We validate our proposed approach on two large scale datasets of complex 3D scenarios. The first dataset consists of more than 6000 apartments, whereas the second one consists of 3000 museums that we collect and annotate, in order to establish a starting point for realistic Metaverse museums. With our approach, absolute improvements of +2.0% R@5 and +4.4% in R@5 and R@10 (relative improvement of 4.7% and 9.0%) are obtained compared to previous state-of-the-art results on the first dataset. In the second dataset, compared to a CLIP-based baseline, we obtain absolute improvements of +4.1% R@1, +11.4% R@5, +13.7% R@10 (relative improvements of 585%, 242%, and 142%, respectively).

A preliminary version of this manuscript was published at ACM ICMR 2024 [14]. Here, we extend it by working in three main research directions. First, the approach proposed here is more general, as it can be applied to more diverse and rich scenarios, whereas the previous approach—especially, the function for identifying the distance to be enforced—was specifically designed for the apartments. In this step, we introduce a novel method based on multiple models to process the very long descriptions of the museums and support the decision of the distance. Second, we generalize the previous approach to use multiple likeness classes, as we hypothesize that more complex and rich scenarios, such as the museums, perform better when more classes at training time are used compared to simpler scenarios, such as the apartments. Finally, we collect a novel dataset of 3000 digital museums, introducing a much more challenging benchmark and presenting evidence that our approach leads to better performance than existing methods.

The rest of the paper is organized as follows. Related works are highlighted and contextualized in Section II. The proposed methodology is thoroughly described in Section III. Section IV explains the experimental setting and the research questions identified and discussed in this work, along with those which remain open. Finally, Section V draws the conclusion and plans the future work.

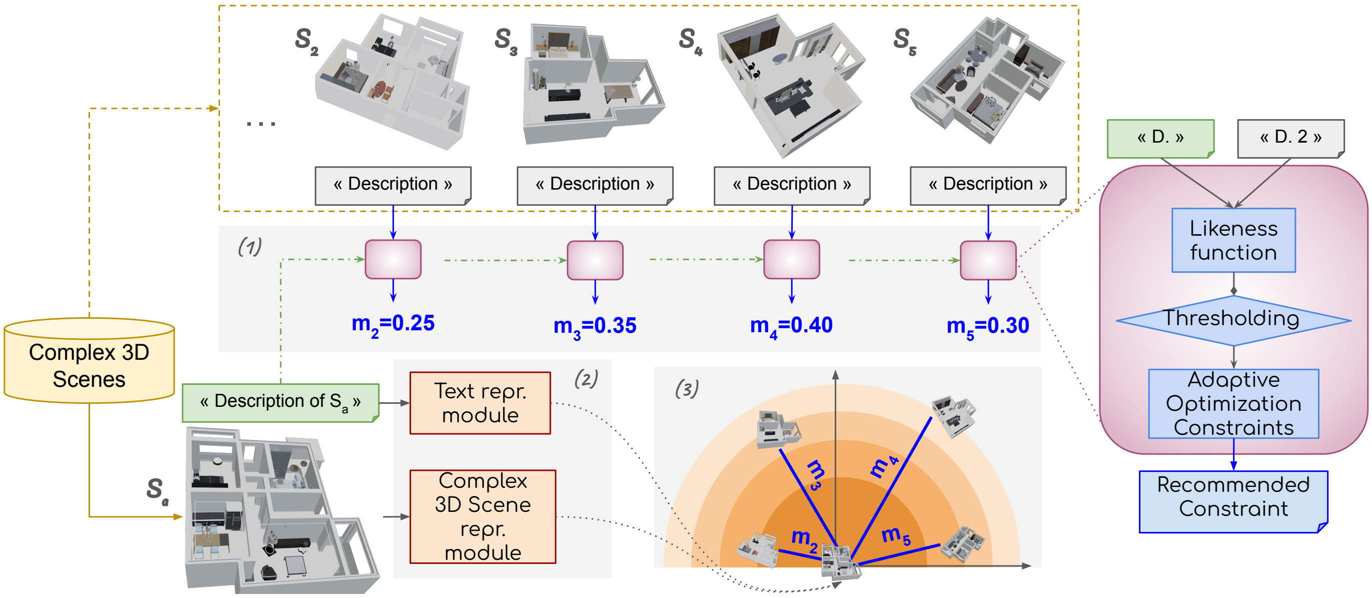


Fig. 2. The proposed methodology learns a joint (complex 3D scene and text) embedding space using Adaptive Optimization Constraints. There are three main steps. (1) Given a complex 3D scene S_a and its description D_a, a customized constraint is defined in terms of D_a and the description of other scenes (S_2, S_3, etc). The constraint is visualized as a margin (m_1, m_2, etc) representing the distance enforced between their representations in the embedding space. (2) The representations of D_a and S_a are computed through trainable methods. (3) The customized constraints are enforced on the joint embedding space. For simplicity, we visualize only the 3D scene for each pair of inputs and connect to it the output of both previous representation modules.

II. Related work

A. Ranking 3D objects and complex scenes based on text

The retrieval of 3D objects has sparked increasing interest during the past few years. The yearly Shape Retrieval Contest (SHREC) started in 2013 with four tracks differing in terms of query (sketch, image, or 3D object) and 3D format used in the collection [23]–[26]. Over time, the datasets became larger, with more complex 3D models and varied data formats, spanning from generic and open-domain objects [27]–[29] to very specific themes, such as 3D models of animals [30], [31], hand gestures [32], [33], and protein surfaces [34], [35]. Nowadays, most methods performing cross-modal retrieval of 3D objects rely on CLIP [20], and then either apply it to the point-of-view images, leveraging the vault of knowledge obtained from the large scale multimodal pretraining [36]–[39], or performing additional pretraining to align 3D point clouds or meshes to point-of-view images and text [9], [11], [12], [40]. However, all these studies work on single 3D objects. Only recently, few works started to investigate the more challenging problem given by complex 3D scenes (furnished indoor scenes [13]–[15] and multimedia-rich 3D scenes [16], [17]). In this work, we focus on the retrieval of complex 3D scenes such as fully furnished virtual apartments and digital museums, for which we collect a novel dataset of 3000 museums including multiple rooms and several multimedia elements per room.

B. Contrastive loss functions

In recent years, contrastive learning became a popular way for learning how to solve cross-modal tasks relying on deep learning and annotated vision-language datasets [20], [41], [42]. Contrastive learning aims at teaching models to output similar representations for paired multimodal inputs (e.g. a 3D scene and its description) and dissimilar for unpaired inputs. While early approaches used only two inputs at a time [19], subsequent works used multiple inputs [21], [43], [44] to model intra-class and inter-class relations at once. A common aspect among these functions is an hyperparameter, called margin or distance, which defines the distance to be enforced between unpaired representations in the embedding space. Typically, it is fixed, i.e. the distance enforced between unpaired representations is always the same. Adaptive solutions were also proposed, e.g. increasing it monotonically as the training goes on [45], varying it based on paired-unpaired distance [46], [47], scheduling the use of different models to compute it in subsequent phases of the training process [48], or defining it in terms of part-of-speech-based relevance scores [49]. Differently, we design a method to automatically quantify their "likeness" and, based on that, define a proper margin to be enforced. By doing so, two scenes which are very similar will be separated by a small margin, and the margin increases as their similarity decreases (see Fig. 2 for an example).

III. Proposed methodology

Figure 2 presents an overview of the proposed methodology, ALCER3D, which is made of three main components. The first component presents the procedure we implement to automatically define customized constraints based on the input data. This novel component allows to customize the training procedure, making it possible to capture fine-grained details of the data which are not considered in standard procedures (more details in Sec. III-A). Once the constraints are defined, the representation of each input scene/description is obtained through state-of-the-art neural components (Sec. III-B). Finally, standard contrastive learning techniques are modified in order to integrate the proposed adaptive constraints, implementing a customized contrastive learning framework (Sec. III-C).

A. Automatically recommending Adaptive Optimization Constraints

As shown in Figure 2 (1), given the complex 3D scene S_a and its description, a value m_i is computed against the other complex 3D scenes (S_i with $i = 2, 3, \dots$). The procedure computing this value is shown on the right side of Fig. 2 and is a core mechanism of the proposed methodology. In fact, the value m_i captures the desired distance to be enforced in the joint embedding space between the representations of S_a and S_i. This is different than standard procedures which enforce the same distance between all unpaired samples. Yet, by learning to organize the embedding space enforcing the same distance between any pair of samples, fine-grained details discriminating different scenes are neglected.

The procedure that we design for recommending distance constraints, RecCon (Alg. 1), consists of three main sub-components: a likeness function (Sec. III-A1), a thresholding step (Sec. III-A2), and the module used to recommend the adaptive constraint based on the information obtained by the other two sub-components (Sec. III-A3).

Algorithm 1 Algorithmic description of our procedure $RecCon$ for recommending adaptive optimization constraints at *training* time.

Require: The thresholds $\tau_1, \dots, \tau_{N-1}$, the margin values $\Delta_1, \dots, \Delta_N$, class-to-constraint mapping function M
Input: Description of two complex 3D scenes, D and D_2
Output: Recommended distance constraint m
$l \leftarrow L(D, D_2)$ ▷ (1) Computation of likeness
if $l \leq \tau_1$ **then** ▷ (2) Thresholding step
  $c \leftarrow c_1$ ▷ Very low likeness class
else if $l > \tau_{N-1}$ **then**
  $c \leftarrow c_{N-1}$ ▷ Very high likeness class
else
  for $i = 1$ **to** $N-2$ **do**
    if $l > \tau_i \wedge l \leq \tau_{i+1}$ **then**
      $c \leftarrow c_{i+1}$ ▷ Intermediate likeness class
    end if
  end for
end if
$m \leftarrow M(c)$ ▷ (3) Compute recommended distance
return m

1) The likeness function L : The first step in our procedure RecCon computes a score representing the correspondence of two input complex 3D scenes using a likeness function L. When the scenes share similar rooms and objects they will get high likeness scores, whereas low scores are expected otherwise. To implement this function, we process the textual descriptions with a neural model. However, the description of a complex 3D scene can be very long and rich, so it is challenging to compute the likeness of pairs of descriptions using a single language model. Therefore, we define the function L as a novel ensemble of pretrained language models aggregating their opinions on the likeness of the descriptions to make a more informed decision. Specifically, we use three models, balancing complexity and empirical capabilities. For every model, we compute the similarity for the pair of descriptions following its methodology, and then normalize the values in the range [0, 1] before the aggregation step. In fact, they are distributed in different ranges, depending both on the data and on the embedding space captured by the model, making it difficult to compare them fairly. To account for this, we normalize the values using minmax with extremes computed as a preprocessing step on all possible training pairs. This step is only done for computing the likeness values at training time, hence it does not influence the textual representations on the test set. Then, the individual values are pooled to make a collective decision. Notably, as this step is performed only during training, no decrease in inference times is observed.

2) The thresholding step : In the second step, a "class" of likeness for the input scenes is identified based on the likeness score. The intuition is that their representations should be separated by a different distance in the embedding space based on which class captures them: if the input scenes are similar (hence, L outputs a high score), the distance should be low, and vice versa, with varying degrees standing in between. To capture these nuances, we introduce the hyperparameter N representing the amount of likeness classes. We define N-1 thresholds, $\tau_1 < \tau_2 < \dots < \tau_{N-1}$, acting as the borders between the different classes. For instance, consider $N = 3$ and the scenes shown in Fig. 2. With $N = 3$, the classes could be interpreted as considering the two descriptions to be "very similar", "slightly similar", or "dissimilar". Since S_a and S_2 are very similar, their likeness score is high (greater than τ_2) meaning that S_2 falls in the "very similar" class compared to S_a, whereas S_4 likely falls in the "dissimilar" class, given that they are very different (hence, their likeness is lower than τ_1).

Formally, given the likeness score $l = L(D_a, D_i)$ and the thresholds $\tau_{i,i \leq N-1}$, we consider a thresholding function T defined as follows:

$$T_{\tau_1,\dots,\tau_{N-1}}(l) = \begin{cases} c_1, & \text{if } l \leq \tau_1 \\ c_2, & \text{if } l > \tau_1 \wedge l \leq \tau_2 \\ \dots & \\ c_{N-1}, & \text{if } l > \tau_{N-2} \wedge l \leq \tau_{N-1} \\ c_N, & \text{if } l > \tau_{N-1} \end{cases} \quad (1)$$

where $c_1, \dots, c_N$ identify the classes ordered from low to high likeness.

3) Mapping likeness classes to customized margins: The third and final step of our procedure RecCon maps the information obtained from previous steps to customized constraints. As mentioned before, the aim is to enforce a distance between representations in the embedding space, and such a distance is related to the likeness class. Therefore, taking inspiration from the triplet loss function [21] commonly used in contrastive learning settings, we represent the distance as the margin hyperparameter. Different from the standard approach, we introduce one customized margin for each likeness class, namely Δ_1 to Δ_N. Then, the function M in Alg. 1 simply maps c_i to Δ_i. Note that the biggest margin is assigned to the lowest likeness class, hence the margins follow a descendant order, i.e. $\Delta_1 > \Delta_2 > \cdots > \Delta_N$.

B. Latent representations for complex 3D scenes

To extract a representation for the complex 3D scenes and the related descriptions, we build on top of the state-of-the-art cross-modal retrieval approach CLIP [20]. The visual and textual backbones, implemented by means of a Vision Transformer [50] and a standard Transformer [51], are jointly pretrained using the contrastive loss function introduced by CLIP, and then frozen so that the knowledge learned can be transferred to new data. As we extract multiple images from each complex 3D scene, the Vision Transformer is used to encode the content of each into a compact vector. Then, a one-dimensional convolutional neural network and a MLP are applied to learn a task-dependent representation of it. For the descriptions, the Transformer is used to extract a vector for each sentence as the descriptions are very long. Then, a bidirectional GRU is applied to learn a contextual representation leveraging the sequential nature of the sentences within the paragraph. The same methodology for extracting the representations of scenes and descriptions was also used in previous works on the topic [13], [14].

C. Adapting a standard contrastive learning framework to use customized distance constraints

After identifying the recommended constraints based on the likeness function (Sec. III-A) and extracting the representations for the input data (Sec. III-B), the constraints are integrated into the training objective so that the representation functions are adjusted to better satisfy them. In our methodology, we start from the triplet loss function and adjust it to integrate our customized constraints, obtaining a novel adaptive version of the triplet loss function. Formally, Equation 2 illustrates the triplet loss calculation where s is the cosine similarity, a, p, and n are used to represent the anchor, positive, and negative elements of the triplet:

$$L(a,p,n) = max(0, RecCon(a,n) + s(n,a) - s(p,a)) \quad (2)$$

Equation 3 describes the loss calculation across a batch $\mathcal{B}$, consisting of pairs made by a complex 3D scene and its textual description, i.e. $\mathcal{B} = \{(s_i, d_i) \mid s_i \in S, d_i \in D\}$ where S and D represent the set of scenes and descriptions, respectively.

$$\mathcal{L}_{AT}(\mathcal{B}) = \frac{1}{|S|}\Bigg(\sum_{(S_a,D_a)\in\mathcal{B}} \sum_{\substack{D_n\in B\\ D_n\neq D_a}} L(S_a, D_a, D_n) + \sum_{(S_a,D_a)\in\mathcal{B}} \sum_{\substack{S_n\in S\\ S_n\neq S_a}} L(D_a, S_a, S_n)\Bigg) \quad (3)$$

Note that in standard triplet loss [21], the margin Δ is fixed, whereas here $RecCon(a,n)$ customizes the margin based on the input. This means that all the negatives are considered equally, neglecting that the complex 3D scenes at hand may have very nuanced degrees of shared content. Therefore, at training time, many fine-grained details are lost and not properly understood by the model, whereas with our procedure the aim is to support the model in learning them.

IV. Experimental results

A. Datasets

We validate our approach on two large datasets. The first one (referred to as **Apartments** later on) is a public dataset consisting of more than 6000 apartments [13], each described by a paragraph detailing each room and the furniture contained in it. The second dataset (we refer to it as **Museums**) is novel and collected for this manuscript. It consists of 3000 museums, each containing between 5 and 8 rooms (6.5 on average) decorated with multiple paintings taken from SemArt [52]. Each museum is described by a paragraph containing: the number of rooms, then for each room the amount of paintings and their school, followed by a description of each painting. Given the complexity and richness of the two scenarios, the paragraphs are very long, averaging 16 sentences and 319 words for the Apartments, 171 sentences and 3447 words for the Museums. An example of these data is reported in Figure 3. For the Apartments, we follow the split used in [13], consisting of 4256, 912, and 913 apartments for train, validation, and test set. For the Museums, we split them with 70/15/15 ratio, resulting in 2100, 450, and 450 museums.

B. Existing methods

We consider the following methods for performance comparison. **CNV** uses a Transformer and a Vision Transformer, jointly pretrained with CLIP [20], for feature extraction from the sentences forming the descriptions and the multi-view images taken within the 3D scenes, respectively. Then, using standard contrastive learning, it learns optimal weights for a bidirectional GRU processing the input sentences, and a neural network composed by a convolutional layer and a MLP. **FArMARe** [13] extends CNV by implementing a multi-task learning setting with the addition of a classification objective aimed at injecting into the model better knowledge of furniture objects. We also include two other simpler baselines from [13]: **NLB**, which is a zero-shot baseline taking the mean of CLIP-based vectors and then ranking using cosine similarity, and **AFN**, which learns an MLP on top of the mean pooled vectors. Finally, **AdOCTeRA** [14] is the early implementation of the method proposed in this study. It is less general, as the procedure for recommending constraints is designed on the structure of the Apartments descriptions, and has more difficulties in dealing with longer and complex descriptions, such as those found in the Museums dataset, since it only uses one language model for the likeness function.

C. Evaluation protocol

The evaluation protocol consists of the following steps. First, the training procedure is performed and a trained model is obtained by selecting the best performer on the validation

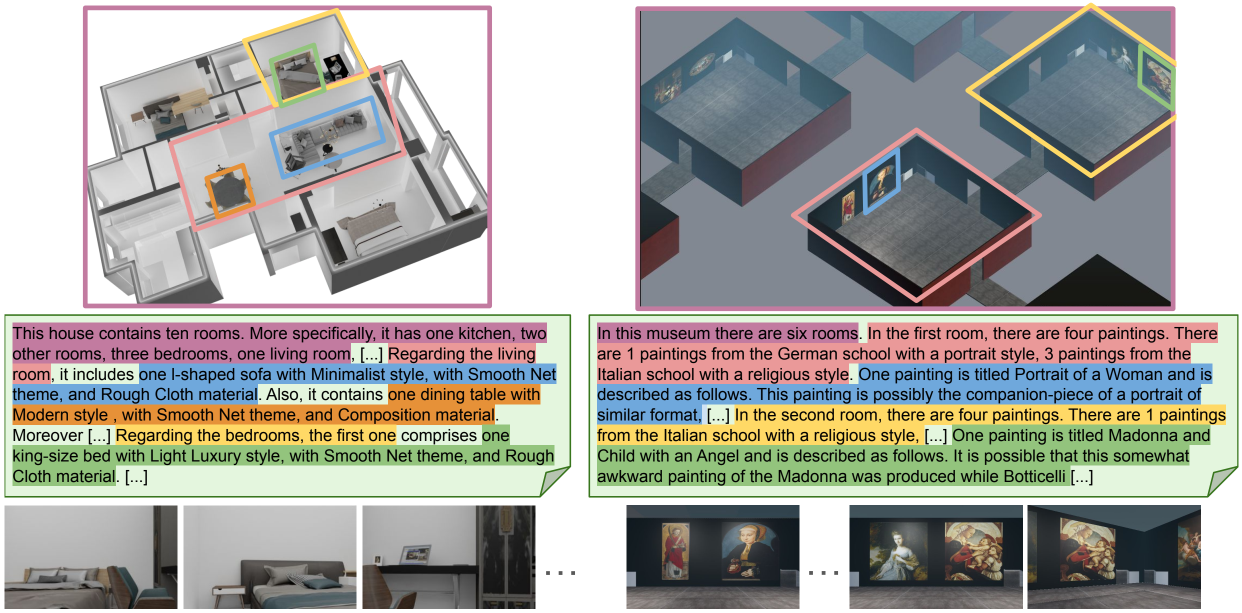


Fig. 3. Examples taken from the datasets considered in this study. For both datasets (Apartments on left, Museums on right), we show the 3D scene, part of the description, and multi-view images extracted from the scene. We include colored boxes to show how the descriptions and the visual data are aligned.

set. Then, this model is executed on the testing set, allowing to assess its performance on unseen data. This process is repeated four times with random seeds and the performance metrics are reported as average across the different runs. The performance metrics include the recall rates, the sum of recalls, and the median rank. The recall at K (or **R@K**) measures the amount of times the groundtruth was observed in the top K of the ranking list. Common values for K are 1, 5, and 10. The sum of recalls (**Rsum**) is also frequently reported in cross-modal retrieval works, and consists in the sum of R@1, R@5, and R@10 in both directions of retrieval (i.e., both scene-to-text and text-to-scene) for a more comprehensive view on the results. Finally, the median rank (**MedR**) assesses the median position of the groundtruth in the ranking list.

D. Implementation details

We use ViT-B-32 [50] to extract visual features from the multi-view images taken within the 3D scenes. For the descriptions, a 12-layer Transformer [51] was used. The Vision Transformer and the Transformer are jointly pretrained using CLIP on LAION-2B [53]. The models used for the likeness computation are `paraphrase-distilroberta-base-v2` for DistilRoBERTa [54], [55], `all-MiniLM-L6-v2` for MiniLM [56], and `gte-large-en-v1.5` for GTE-Large [57], all obtained from the SentenceTransformers environment. Each training run uses a batch of 64 and lasts up to 50 epochs, early stopping if the validation loss does not decrease by at least .0001 in 25 epochs. The weights and biases are optimized using Adam, with initial learning rate of .008 and decay factor of 25% after 27 epochs. In the comparison with state-of-the-art, we use one threshold ($\tau = 0.25$) with margins set to 0.25 and 0.40 for the Apartments dataset, and two thresholds ($\tau_1 = 0.45, \tau_2 = 0.75$) with margins set to 0.25, 0.40, and 0.55 for the Museums dataset. The implementation uses PyTorch 2.2.0 and the experiments are run on a machine using one RTX A5000 GPU, 32 GB of RAM, and an Intel Core i7-9700K.

E. How does our approach compare to existing works?

Here, we compare the performance obtained using ALCER3D to existing solutions, namely NLB, AFN, CNV, FArMARe [13], and AdOCTeRA [14]. More details on these models are presented in Sec. IV-B. In Table I, we present the results. Two major observations are made.

Compared to the baseline, the proposed approach leads to consistent improvements in both scenarios considered. Specifically, for the Apartments dataset, +0.9% R@1, +3.7% R@5, and +4.6% text-to-Apartment R@10 are obtained with ALCER3D, whereas for Museums, the improvements measure +4.1%, +11.4%, +13.7%, respectively. These results show both the effectiveness of the proposed approach, and its increase in effectiveness when more complex descriptions are involved. In fact, Museums' descriptions are considerably longer and richer than the Apartments' descriptions (see Sec. IV-A), and the use of multiple models when computing the likeness becomes fundamental, resulting in four times the performance compared to AdOCTeRA (from 1.2% to 4.8% R@1) and almost seven times the baseline performance (0.7% to 4.8% R@1).

Compared to the early prototype, AdOCTeRA, the method proposed in this manuscript has several benefits. First, while on the Apartments dataset it leads to only slightly better performance (overall, +2.1 Rsum), the proposed approach is more general, as it uses the full description to adapt the optimization constraint. Second, when working on more complex descriptions, it obtains considerably higher performance. We adapted AdOCTeRA to the new scenario, using the same model previously used for the likeness computation on the full description, with two thresholds and three similarity classes. The proposed approach achieves 4.8% R@1, 16.1% R@5, and

TABLE I
COMPARISON OF EXISTING METHODS TESTED ON THE TWO DATASETS UNDER ANALYSIS. POSITIVE CHANGES (Δ) IN PERFORMANCE ARE HIGHLIGHTED IN GREEN. T-A AND T-M STAND FOR TEXT-TO-APARTMENT AND TEXT-TO-MUSEUM (AND VICE VERSA FOR A-T AND M-T). †: REPRODUCED USING THE CODE FROM [13]. ⋆: TESTED WITH DISTIL-ROBERTA, TWO THRESHOLDS, AS IN [14]. DETAILS IN SEC. IV-E.

	Apartments								
	Text-to-Apartment			Apartment-to-Text			T-A	A-T	
Method	R@1	R@5	R@10	R@1	R@5	R@10	MedR	MedR	Rsum
NLB [13]	0.1	0.6	1.3	0.6	1.9	3.7	413	339	8.2
AFN [13]	10.6	25.5	34.2	8.9	23.2	31.3	29	31	133.7
CNV [13]	23.7	40.6	48.9	23.1	40.7	48.9	11	12	225.9
FArMARe [13]	**24.8**	42.6	49.1	**25.7**	41.4	49.1	11	11	232.7
Ours									
AdOCTeRA [14]	24.5	**44.3**	53.0	25.0	42.9	52.0	**8.3**	9	241.8
ALCER3D	24.6	**44.3**	**53.5**	24.6	**44.1**	**52.1**	**8.3**	**8.5**	**243.1**
Δ baseline	0.9	3.7	4.6	1.5	3.4	3.2	-2.8	-3.5	17.2
Δ FArMARe	-0.2	1.7	4.4	-1.1	2.7	3.0	-2.8	-2.5	10.4
Δ AdOCTeRA	0.1	0.0	0.5	-0.4	1.2	0.1	-0.1	-0.5	1.3
	Museums								
	Text-to-Museum			Museum-to-Text			T-M	M-T	
Method	R@1	R@5	R@10	R@1	R@5	R@10	MedR	MedR	Rsum
NLB	0.4	2.4	4.0	0.2	1.6	3.1	171.0	199.5	11.8
CNV †	0.7	4.7	9.6	0.8	5.6	9.2	55.3	57.3	30.6
Ours									
∼AdOCTeRA ⋆	1.2	7.8	14.8	1.1	7.9	14.7	43.9	45.9	47.6
ALCER3D	**4.8**	**16.1**	**23.3**	**5.6**	**15.9**	**23.7**	**43.4**	**44.5**	**89.5**
Δ baseline	4.1	11.4	13.7	4.8	10.3	14.5	-12.0	-12.8	58.9
Δ ∼AdOCTeRA	3.6	8.3	8.5	4.5	8.0	9.0	-0.5	-1.4	41.9

23.3% R@10 whereas AdOCTeRA stops at 1.2%, 7.8%, and 14.8%, respectively.

F. How many models should we use to compute the likeness?

The second question investigates the number of models used for the likeness computation. Three models, both small and large while still achieving good performance, are considered. Distil-RoBERTa (82M parameters) is obtained through the distillation of RoBERTa-base [54] following the procedure of DistilBERT [55]. MiniLM [56] (22.7M parameters) introduces several technical improvements resulting in a more flexible approach, which obtains better generalization through the use of an additional "teacher assistant" for the distillation phase. Finally, GTE-Large [57] is a large model (434M parameters) pretrained with data coming from multiple sources, including data from the CodeSearchNet benchmark [58].

In this experiment, the results reported in Table II are obtained using at training time a single threshold, i.e. two likeness classes, for easier comparison. We observe that the answer to the main question depends on the richness of the descriptions. In fact, for Apartments, a single model seems to suffice, with MiniLM leading to 24.8% R@1 and a Rsum of 243.9. Nevertheless, using three models does not lead to significantly worse results (e.g., -0.2% R@1). However, as the descriptions become more complex, using more models to support the decision results in better performance. Looking at *average* R@1, the lowest performance is obtained with a single model (MiniLM with 2.3%, average of 2.2% text-to-apartment and 2.4% apartment-to-text R@1), whereas 2.7% (2.6% and 2.8%) is obtained adding GTE-Large, and finally 3.2% (2.6% and 3.8%) using all three models together.

TABLE II
PERFORMANCE COMPARISON WHEN VARYING THE APPROACH FOLLOWED FOR COMPUTING THE LIKENESS. M = MINILM [56], R = DISTILROBERTA [54], [55], G = GTE-LARGE [57]. DETAILS IN SEC. IV-F.

	Apartments				
	Text-to-Apartment		Apartment-to-Text		
Method	R@1	R@10	R@1	R@10	Rsum
R	24.6	53.9	24.0	52.3	241.8
G	23.7	52.4	23.9	52.0	238.7
M	**24.8**	53.1	**24.8**	**52.8**	**243.9**
M+G	24.5	**54.0**	24.1	52.0	241.0
R+G	24.6	53.4	24.2	52.7	242.1
R+M	24.6	53.7	23.3	52.6	241.7
M+R+G	24.6	53.5	24.6	52.1	243.1
	Museums				
	Text-to-Museum		Museum-to-Text		
Method	R@1	R@10	R@1	R@10	Rsum
R	1.4	12.2	1.2	13.1	40.7
G	1.4	13.1	1.3	14.4	44.8
M	2.2	17.9	2.4	**19.4**	62.2
M+G	**2.6**	17.4	2.8	17.7	61.1
R+G	1.3	12.7	2.1	13.0	42.7
R+M	1.4	12.7	1.8	12.3	42.1
M+R+G	**2.6**	**18.3**	**3.8**	**18.1**	**65.2**

G. How many likeness classes should we use for training?

The third question explores the effect of changing the amount of likeness classes. Notably, as low as one class could be considered, similar to the baseline where the same margin is used for all samples, to an infinite number of classes, where the margin is directly proportional to the likeness value.

We explore this question across single- to multi-model solutions for likeness computation, starting with MiniLM, since it resulted in the best performance in the previous experiments. We consider a number of classes varying from one to four, as the performance drops off already with three

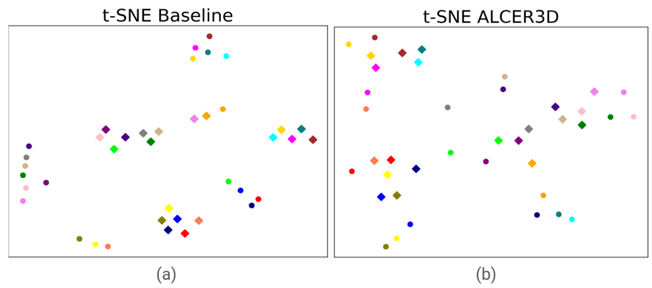


Fig. 4. t-SNE plots of the embedding space comprising 20 random samples (pairs of scenes/circles and descriptions/squares) from the test set of the Museums dataset. These 20 pairs are then projected using the trained (a) baseline approach, and the (b) proposed ALCER3D approach.

classes. The results are reported in Table III. It can be seen that two classes represent a generally viable solution, leading for instance to the highest R@1 in all combinations of methods tested for the Apartments scenario. However, in the Museums dataset, we see that the best results are obtained when three classes are considered (4.8% R@1 and 89.5 Rsum with all three models used for likeness computation, 2.9% R@1 and 72.4 Rsum when MiniLM and GTE-Large are used). This result further corroborates the hypothesis that with more complex scenarios (on average, Museums have 3447 words per description whereas Apartments have 319), having more likeness classes positively influences the final performance.

To better illustrate the effectiveness of using multiple language models together with different margin settings, the t-SNE plot of the embedding space for 20 random samples from the test set of the Museums dataset is presented in Figure 4. The comparison includes both the baseline and ALCER3D, which utilizes all language models and three classes. In these plots, the same colors represent a unique pair (scene and description), while circles denote scenes, and squares represent descriptions. As shown in Figure 4(b), corresponding pairs are much closer to each other compared to Figure 4(a), which does not incorporate any customized margin.

TABLE III
PERFORMANCE COMPARISON WHEN VARYING THE NUMBER OF LIKENESS CLASSES. T-A AND T-M STAND FOR TEXT-TO-APARTMENT AND TEXT-TO-MUSEUM (AND VICE VERSA FOR A-T AND M-T). DETAILS IN SEC. IV-G.

		Apartments				
	Num.	T-A		A-T		
Method	*Classes*	R@1	R@10	R@1	R@10	Rsum
(base)	1	23.7	48.9	23.1	48.9	225.9
M	2	**24.8**	**53.1**	**24.8**	**52.8**	**243.9**
M	3	24.7	52.8	24.5	52.2	240.4
M	4	**24.8**	52.5	23.9	51.9	239.1
M+G	2	**24.5**	**54.0**	24.1	**52.0**	**241.0**
M+G	3	24.3	53.1	**24.5**	51.1	238.8
M+G	4	24.0	53.1	23.2	52.1	239.0
R+M	2	**24.6**	**53.7**	23.3	**52.6**	**241.7**
R+M	3	23.9	52.5	23.7	51.7	238.6
R+M	4	24.0	53.3	**24.3**	52.2	241.5
M+R+G	2	**24.6**	53.5	**24.6**	**52.1**	**243.1**
M+R+G	3	24.0	**53.6**	24.5	51.9	240.7
M+R+G	4	24.1	52.8	23.4	51.8	237.8
		Museums				
	Num.	T-M		M-T		
Method	*Classes*	R@1	R@10	R@1	R@10	Rsum
(base)	1	0.7	9.6	0.8	9.2	30.6
M	2	**2.2**	**17.9**	**2.4**	**19.4**	**62.2**
M	3	0.9	11.6	1.4	12.3	38.3
M	4	1.5	11.8	1.4	12.1	40.8
M+G	2	2.6	17.4	2.8	17.7	61.1
M+G	3	**2.9**	**20.6**	**3.9**	**19.4**	**72.4**
M+G	4	0.8	11.6	1.4	11.4	37.4
R+M	2	**1.4**	**12.7**	**1.8**	**12.3**	**42.1**
R+M	3	1.1	11.9	1.1	11.7	37.9
R+M	4	1.3	11.3	1.1	11.2	36.6
M+R+G	2	2.6	18.3	3.8	18.1	65.2
M+R+G	3	**4.8**	**23.3**	**5.6**	**23.7**	**89.5**
M+R+G	4	1.5	12.3	1.6	11.9	39.8

H. What would happen if we used CLIP to compute the likeness?

In the proposed method, we use an ensemble of language models to analyze the input descriptions and compute a likeness value. Here, we test whether using the multimodal knowledge encoded in CLIP to compute the likeness leads to better generalization. To do so, given the representation of the descriptions and the images, we separately compute the textual and visual likeness, and then we take their average to obtain the final likeness value. The representations are given by the mean-pooled image-level and sentence-level vectors.

We compare the usage of CLIP to the four combinations of the textual models that we used before, i.e. MiniLM (M), MiniLM+GTE (M+G), RoBERTa+MiniLM (R+M), and finally all the three models (M+R+G). In the eight comparisons, we use the hyperparameters—number of classes, thresholds, and margins—from the previous experiment. For the Apartments, we used two classes, with thresholds set to 0.85, 0.55, 0.85, and 0.25, while the margins were set to (0.25, 0.40), (0.25, 0.30), (0.25, 0.55), and (0.25, 0.40), respectively. On the other hand, for the Museums, we used two classes for M, with the threshold set to $\tau = 0.25$ and margins (0.25, 0.55); $N = 3$ for M+G with $\tau_1 = 0.35, \tau_2 = 0.75$, and margins (0.25, 0.30, 0.40); $N = 2$ for R+M with $\tau = 0.55$, and margins (0.25, 0.45); finally, $N = 3$ for M+R+G with $\tau_1 = 0.45, \tau_2 = 0.75$, and margins set to (0.25, 0.40, 0.55).

The following observations are made. On the Apartments dataset, using CLIP leads to better performance than the baseline (225.9 Rsum), obtaining up to 238.8 Rsum (+13 over the baseline). This improvement is more marked on the R@10 where up to +4% (52.9% T-A R@10 compared to 48.9%) is observed in setting (b). However, the performance obtained in these scenarios does not reach the same level obtained by the proposed method. On the Museums dataset, the proposed method obtains far better results than using CLIP for computing the likeness. Specifically, up to 89.5 Rsum is obtained whereas using CLIP the performance is similar to the baseline (up to 31.5 Rsum in setting (g) compared to 30.6).

I. How do the thresholds and margins value affect the performance?

As part of the proposed methodology, the value of the margins and the thresholds separating the likeness classes represent key hyperparameters. Therefore, we investigate how they affect the final performance in the Apartments dataset,

TABLE IV
PERFORMANCE COMPARISON BETWEEN TEXT-ONLY METHODS (MINILM, GTE-LARGE, DISTIL-ROBERTA) FOR COMPUTING THE LIKENESS AND MULTIMODAL METHODS (CLIP). T-A AND T-M STAND FOR TEXT-TO-APARTMENT AND TEXT-TO-MUSEUM (AND VICE VERSA FOR A-T AND M-T). DETAILS IN SEC. IV-H.

		Apartments				
		T-A		A-T		
	Method	R@1	R@10	R@1	R@10	Rsum
	(base)	23.7	48.9	23.1	48.9	225.9
(a)	M	**24.8**	**53.1**	**24.8**	**52.8**	**243.9**
	C	22.7	49.9	22.8	49.7	227.4
(b)	M+G	**24.5**	**54.0**	**24.1**	**52.0**	**241.0**
	C	23.8	52.9	23.9	**52.0**	238.8
(c)	R+M	**24.6**	**53.7**	**23.3**	**52.6**	**241.7**
	C	22.5	50.2	22.6	50.0	227.2
(d)	M+R+G	**24.6**	**53.5**	**24.6**	**52.1**	**243.1**
	C	23.8	52.0	24.1	51.3	236.8
		Museums				
		T-M		M-T		
	Method	R@1	R@10	R@1	R@10	Rsum
	(base)	0.7	9.6	0.8	9.2	30.6
(e)	M	**2.2**	**17.9**	**2.4**	**19.4**	**62.2**
	C	0.8	8.9	0.7	9.8	28.9
(f)	M+G	**2.9**	**20.6**	**3.9**	**19.4**	**72.4**
	C	0.9	8.6	1.0	8.9	28.7
(g)	R+M	**1.4**	**12.7**	**1.8**	**12.3**	**42.1**
	C	0.8	9.7	1.5	10.0	31.5
(h)	M+R+G	**4.8**	**23.3**	**5.6**	**23.7**	**89.5**
	C	0.8	8.6	0.9	9.4	28.5

TABLE V
ABLATION STUDY ON THE SENSITIVITY OF TWO HYPERPARAMETERS (THRESHOLDS AND MARGINS) FOR OUR METHOD. EXPERIMENTS PERFORMED ON THE APARTMENTS DATASET. DISCUSSION IN SECTION IV-I.

	Thresholds		*Margins*				
	Low	High	m_1	m_2	m_3	R@1	R@10
(a)	**0.25**	0.75	0.25	0.30	0.35	23.9	50.5
	0.35	0.75	0.25	0.30	0.35	**24.0**	51.6
	0.45	0.75	0.25	0.30	0.35	**24.0**	**53.6**
	0.55	0.75	0.25	0.30	0.35	22.5	50.4
(b)	0.45	**0.65**	0.25	0.30	0.35	23.6	51.9
	0.45	**0.75**	0.25	0.30	0.35	**24.0**	**53.6**
	0.45	**0.85**	0.25	0.30	0.35	23.6	52.6
(c)	0.45	0.75	0.25	**0.28**	0.35	23.9	51.9
	0.45	0.75	0.25	**0.30**	0.35	**24.0**	**53.6**
	0.45	0.75	0.25	**0.32**	0.35	23.0	52.1
(d)	0.45	0.75	0.25	0.30	**0.33**	23.6	49.6
	0.45	0.75	0.25	0.30	**0.35**	**24.0**	**53.6**
	0.45	0.75	0.25	0.30	**0.37**	22.4	50.3
	0.45	0.75	0.25	0.30	**0.40**	22.7	50.7
(e)	0.45	0.75	0.25	**0.35**	**0.40**	23.0	51.6
	0.45	0.75	0.25	**0.35**	**0.45**	22.6	51.0
	0.45	0.75	0.25	**0.35**	**0.50**	22.2	51.7

considering a setting with three likeness classes. Specifically, the following procedure is used to select the margins and thresholds. For the thresholds, we evaluated several values ranging from 0.25 to 0.55 for the lower threshold while fixing the upper threshold to 0.75; then, after fixing the lower threshold to 0.45, we evaluated several values in the remaining interval (0.65 to 0.85) for the upper threshold. After fixing the thresholds, we set the first margin to 0.25, as done in existing cross-modal retrieval works [14], [16], [59], [60], whereas for the other margins, we performed a grid search varying the values to increase or decrease the distance between embeddings, with increments ranging between 0.02 and 0.05. The results are reported in Table V.

Overall, the performance is sensibly affected by the choice of these hyperparameters, with the R@1 ranging from 22.5% to 24.0% and the R@10 from 49.6% and 53.6%. In (a) and (b), we consider a fixed set of margin values, while varying the lower and upper thresholds, respectively. In both cases, we fix one of the thresholds while varying the other in increments of 0.10 in the remaining interval. When increasing the lower threshold from 0.25 to 0.55, the performance increases steadily, especially in terms of R@10, increasing to 3.1% R@10 when it is set to 0.45. Similarly, in Table V(b), we observe that varying the upper threshold in the remaining interval also leads to sensible variations in R@10 compared to setting it to 0.75. For the margins, we fix $m_1 = 0.25$ and vary m_2 and m_3. First, in (c), we vary m_2 with increments of 0.02, but the performance decreases in both directions. Similarly, when varying m_3 in (d), we also observe negative variations in the performance. In (e), we try both to set $m_2 = 0.35$ and vary m_3 in [0.40, 0.50], but the performance further decreases. These results show that carefully finetuning these hyperparameters is important to improve on the final task. In our experiments, the best combination sets the thresholds to 0.45 and 0.75, and the three margins to 0.25, 0.30, and 0.35.

J. Limitations

We highlight two main limitations in our work.

First, the use of unimodal representations for computing the likeness function. We showed that leveraging large vision-and-language models to compute a multimodal likeness leads to comparable yet slightly lower performance on the Apartments scenario compared to the proposed approach. However, unfavorable results are obtained when using very complex scenarios, such as those in Museums (see Sec. IV-H for details). Nonetheless, it is still limiting that at training time our methodology sees the textual descriptions to compute the likeness and identify a proper margin. Further research is required to test more complex multimodal models and understand how to adapt them to the task.

Second, while with the proposed approach we obtain considerable improvements, especially when dealing with complex scenarios and descriptions, the use of multiple models has notable impact either on training time and GPU memory, or on disk space usage and RAM memory. If the pretrained language models are used for creating a representation of the descriptions *while training*, then the models substantially uses the GPU memory (about 2.7 GB in our setting), limiting other hyperparameters such as the batch size, while also slowing down the process, since the extraction process takes a considerable amount of time (e.g. about 2.8s when using MiniLM on a batch of 64 descriptions taken from the Museums dataset). If the feature extraction and the computation of the likeness values are performed *before training*, as preprocessing steps, the training time is almost unaffected by the additional processing required to compute the modified loss, however there is an increase in disk space (about 1 GB) and RAM usage (about

5.7 GB) to keep the matrices containing the precomputed likeness values in memory and to avoid reading them from disk when needed, which would increase the training time. While these aspects do not affect inference time, which is more important when it comes to practical applications, they still represent some limitations we could not overcome with existing technologies.

V. Conclusions

With the Metaverse and its immersive experiences continuously growing in size, the problem of finding those which satisfy the constraints and desires expressed by the user is becoming urgent. Previous works on the topic, which focused on retrieving complex 3D indoor [13], [15] and multimedia-rich scenes [16], [17], inherited the state-of-the-art cross-modal approach CLIP [20] and built dedicated solutions on top of it. This approach is not optimal: when contrasting the representation of a 3D scene and its description against that of other scenes/descriptions, standard approaches neglect the many degrees of shared content between them. Therefore, to capture these nuances and instill them into the model, we devised a novel learning strategy with personalized and automatically recommended distances to be enforced on the joint embedding space. We validated our approach on two datasets and observed improved retrieval performance, both when comparing to our baseline (+3.7% R@5 on Apartments, +11.4% R@5 on Museums) and to existing methods, obtaining +1.7% and +8.3% R@5, respectively. We confirmed the effectiveness of aggregating the opinions given by multiple models to support the procedure for personalizing the learning constraints, especially when dealing with very complex scenarios. Furthermore, we showed that the amount of likeness classes to be used varies based on the dataset, as two classes are optimal for Apartments but three are generally better for Museums. Finally, we highlighted two limitations of our approach, suggesting possible future directions.

References

[1] J. Joshua, "Information bodies: computational anxiety in neal stephenson's snow crash," *Interdisciplinary Literary Studies*, vol. 19, no. 1, pp. 17–47, 2017.

[2] J. Guan, J. Irizawa, and A. Morris, "Extended reality and internet of things for hyper-connected metaverse environments," in *IEEE VRW*, 2022, pp. 163–168.

[3] Y. A. Sekhavat, "Privacy preserving cloth try-on using mobile augmented reality," *IEEE Transactions on Multimedia*, vol. 19, no. 5, pp. 1041–1049, 2016.

[4] M. Yuan, I. R. Khan, F. Farbiz, S. Yao, A. Niswar, and M.-H. Foo, "A mixed reality virtual clothes try-on system," *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 1958–1968, 2013.

[5] X. Chen, D. Zou, H. Xie, and F. L. Wang, "Metaverse in education: contributors, cooperations, and research themes," *IEEE Transactions on Learning Technologies*, vol. 16, no. 6, pp. 1111–1129, 2023.

[6] X. Chen, Z. Zhong, and D. Wu, "Metaverse for education: technical framework and design criteria," *IEEE Transactions on Learning Technologies*, vol. 16, no. 6, pp. 1034–1044, 2023.

[7] L. Ren, J. Dong, L. Zhang, Y. Laili, X. Wang, Y. Qi, B. H. Li, L. Wang, L. T. Yang, and M. J. Deen, "Industrial metaverse for smart manufacturing: Model, architecture, and applications," *IEEE Transactions on Cybernetics*, 2024.

[8] K. Chen, C. B. Choy, M. Savva, A. X. Chang, T. Funkhouser, and S. Savarese, "Text2shape: Generating shapes from natural language by learning joint embeddings," in *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*. Springer, 2019, pp. 100–116.

[9] D. Hegde, J. M. J. Valanarasu, and V. Patel, "Clip goes 3d: Leveraging prompt tuning for language grounded 3d recognition," in *IEEE/CVF International Conference on Computer Vision*, 2023, pp. 2028–2038.

[10] Y. Ruan, H.-H. Lee, Y. Zhang, K. Zhang, and A. X. Chang, "Tricolo: Trimodal contrastive loss for text to shape retrieval," in *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 5815–5825.

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

[12] J. Zhou, J. Wang, B. Ma, Y.-S. Liu, T. Huang, and X. Wang, "Uni3d: Exploring unified 3d representation at scale," *arXiv preprint arXiv:2310.06773*, 2023.

[13] A. Abdari, A. Falcon, and G. Serra, "Farmare: a furniture-aware multi-task methodology for recommending apartments based on the user interests," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4293–4303.

[14] ——, "Adoctera: Adaptive optimization constraints for improved text-guided retrieval of apartments," in *Proceedings of the ACM International Conference on Multimedia Retrieval*, 2024.

[15] F. Yu, Z. Wang, D. Li, P. Zhu, X. Liang, X. Wang, and M. Okumura, "Towards cross-modal point cloud retrieval for indoor scenes," in *International Conference on Multimedia Modeling*. Springer, 2024, pp. 89–102.

[16] A. Abdari, A. Falcon, and G. Serra, "Metaverse retrieval: Finding the best metaverse environment via language," in *Proceedings of the 1st International Workshop on Deep Multimodal Learning for Information Retrieval*, 2023, pp. 1–9.

[17] ——, "A language-based solution to enable metaverse retrieval," in *International Conference on Multimedia Modeling*. Springer, 2024, pp. 477–488.

[18] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," in *6th Int'l Conference on Neural Information Processing Systems*, 1993, p. 737–744.

[19] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *IEEE computer society CVPR*, vol. 2. IEEE, 2006, pp. 1735–1742.

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

[21] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[22] H. Fu, B. Cai, L. Gao, L.-X. Zhang, J. Wang, C. Li, Q. Zeng, C. Sun, R. Jia, B. Zhao *et al.*, "3d-front: 3d furnished rooms with layouts and semantics," in *Proc. of IEEE/CVF ICCV*, 2021, pp. 10 933–10 942.

[23] A. Cerri, S. Biasotti, M. Abdelrahman, J. Angulo, K. Berger, L. Chevallier, M. T. El-Melegy, A. A. Farag, F. Lefebvre, A. Giachetti *et al.*, "Shrec'13 track: Retrieval on textured 3d models." in *3DOR@ Eurographics*, 2013, pp. 73–80.

[24] B. Li, Y. Lu, A. Godil, T. Schreck, M. Aono, H. Johan, J. M. Saavedra, and S. Tashiro, "Shrec'13 track: large scale sketch-based 3d shape retrieval," in *3DOR@ Eurographics*, 2013, pp. 89–96.

[25] J. Machado, A. Ferreira, P. B. Pascoal, M. Abdelrahman, M. Aono, M. T. El-Melegy, A. A. Farag, H. Johan, B. Li, Y. Lu *et al.*, "Shrec'13 track: Retrieval of objects captured with low-cost depth-sensing cameras." in *3DOR@ Eurographics*, 2013, pp. 65–71.

[26] I. Sipiran, R. Meruane, B. Bustos, T. Schreck, H. Johan, B. Li, and Y. Lu, "Shrec'13 track: large-scale partial shape retrieval using simulated range images," in *3DOR@ Eurographics*, 2013, pp. 81–88.

[27] H. Abdul-Rashid, J. Yuan, B. Li, Y. Lu, T. Schreck, N.-M. Bui, T.-L. Do, M. Holenderski, D. Jarnikov, K. T. Le *et al.*, "Shrec'19 track: Extended 2d scene image-based 3d scene retrieval," *Training (per class)*, vol. 700, p. 70, 2019.

[28] J. Collins, S. Goel, K. Deng, A. Luthra, L. Xu, E. Gundogdu, X. Zhang, T. F. Y. Vicente, T. Dideriksen, H. Arora *et al.*, "Abo: Dataset and benchmarks for real-world 3d object understanding," in *Proceedings of the IEEE/CVF CVPR*, 2022, pp. 21 126–21 136.

[29] Y. Feng, Y. Gao, X. Zhao, Y. Guo, N. Bagewadi, N.-T. Bui, H. Dao, S. Gangisetty, R. Guan, X. Han *et al.*, "Shrec'22 track: Open-set 3d object retrieval," *Computers & Graphics*, vol. 107, pp. 231–240, 2022.

[30] T.-N. Le, T. V. Nguyen, M.-Q. Le, T.-T. Nguyen, V.-T. Huynh, T.-L. Do, K.-D. Le, M.-K. Tran, N. Hoang-Xuan, T.-L. Nguyen-Ho *et al.*,

"Textanimar: text-based 3d animal fine-grained retrieval," *Computers & Graphics*, vol. 116, pp. 162–172, 2023.

[31] ——, "Sketchanimar: sketch-based 3d animal fine-grained retrieval," *Computers & Graphics*, vol. 116, pp. 150–161, 2023.

[32] A. Caputo, A. Giachetti, S. Soso, D. Pintani, A. D'Eusanio, S. Pini, G. Borghi, A. Simoni, R. Vezzani, R. Cucchiara *et al.*, "Shrec 2021: Skeleton-based hand gesture recognition in the wild," *Computers & Graphics*, vol. 99, pp. 201–211, 2021.

[33] M. Emporio, A. Caputo, A. Giachetti, M. Cristani, G. Borghi, A. D'Eusanio, M.-Q. Le, H.-D. Nguyen, M.-T. Tran, F. Ambellan *et al.*, "Shrec 2022 track on online detection of heterogeneous gestures," *Computers & Graphics*, vol. 107, pp. 241–251, 2022.

[34] F. Langenfeld, T. Aderinwale, C. Christoffer, W.-H. Shin, G. Terashi, X. Wang, D. Kihara, H. Benhabiles, K. Hammoudi, A. Cabani *et al.*, "Surface-based protein domains retrieval methods from a shrec2021 challenge," *Journal of Molecular Graphics and Modelling*, vol. 111, p. 108103, 2022.

[35] A. Raffo, U. Fugacci, S. Biasotti, W. Rocchia, Y. Liu, E. Otu, R. Zwiggelaar, D. Hunter, E. I. Zacharaki, E. Psatha *et al.*, "Shrec 2021: Retrieval and classification of protein surfaces equipped with physical and chemical properties," *Computers & Graphics*, vol. 99, pp. 1–21, 2021.

[36] A. Hamdi, S. Giancola, and B. Ghanem, "Mvtn: Multi-view transformation network for 3d shape recognition," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1–11.

[37] D. Lin, Y. Cheng, S. Mao, A. Guo, and Y. Li, "Peva-net: Prompt-enhanced view aggregation network for zero/few-shot multi-view 3d shape recognition," *arXiv preprint arXiv:2404.19168*, 2024.

[38] X. Ning, Z. Yu, L. Li, W. Li, and P. Tiwari, "Dilf: Differentiable rendering-based multi-view image–language fusion for zero-shot 3d shape understanding," *Information Fusion*, vol. 102, p. 102033, 2024.

[39] D. Song, X. Fu, W. Nie, W. Li, and A. Liu, "Mv-clip: Multi-view clip for zero-shot 3d shape recognition," *arXiv preprint arXiv:2311.18402*, 2023.

[40] M. Liu, R. Shi, K. Kuang, Y. Zhu, X. Li, S. Han, H. Cai, F. Porikli, and H. Su, "Openshape: Scaling up 3d shape representation towards open-world understanding," *Advances in neural information processing systems*, vol. 36, 2024.

[41] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[42] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.

[43] W. Chen, X. Chen, J. Zhang, and K. Huang, "Beyond triplet loss: a deep quadruplet network for person re-identification," in *IEEE conference on computer vision and pattern recognition*, 2017, pp. 403–412.

[44] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," *Advances in neural information processing systems*, vol. 29, 2016.

[45] Y. Zhang, Q. Zhong, L. Ma, D. Xie, and S. Pu, "Learning incremental triplet margin for person re-identification," in *AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 9243–9250.

[46] S. Hu, M. Feng, R. M. Nguyen, and G. H. Lee, "Cvm-net: Cross-view matching network for image-based ground-to-aerial geo-localization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7258–7267.

[47] D. Semedo and J. Magalhães, "Cross-modal subspace learning with scheduled adaptive margin constraints," in *27th ACM Int'l Conference on Multimedia*, 2019, pp. 75–83.

[48] F. He, Q. Wang, Z. Feng, W. Jiang, Y. Lü, Y. Zhu, and X. Tan, "Improving video retrieval by adaptive margin," in *44th Int'l ACM SIGIR*, 2021, pp. 1359–1368.

[49] A. Falcon, S. Sudhakaran, G. Serra, S. Escalera, and O. Lanz, "Relevance-based margin for contrastively-trained video retrieval models," in *Proceedings of the 2022 International Conference on Multimedia Retrieval*, 2022, pp. 146–157.

[50] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2020.

[51] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017.

[52] N. Garcia and G. Vogiatzis, "How to read paintings: semantic art understanding with multi-modal retrieval," in *European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0.

[53] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, and othersiclr, "Laion-5b: An open large-scale dataset for training next generation image-text models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 25 278–25 294, 2022.

[54] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[55] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[56] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers," *Advances in Neural Information Processing Systems*, vol. 33, pp. 5776–5788, 2020.

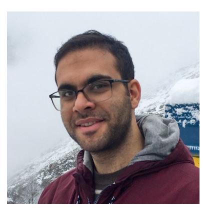
[57] Z. Li, X. Zhang, Y. Zhang, D. Long, P. Xie, and M. Zhang, "Towards general text embeddings with multi-stage contrastive learning," *arXiv preprint arXiv:2308.03281*, 2023.

[58] H. Husain, H.-H. Wu, T. Gazit, M. Allamanis, and M. Brockschmidt, "Codesearchnet challenge: Evaluating the state of semantic code search," *arXiv preprint arXiv:1909.09436*, 2019.

[59] Z. Ji, H. Wang, J. Han, and Y. Pang, "Sman: Stacked multimodal attention network for cross-modal image–text retrieval," *IEEE transactions on cybernetics*, vol. 52, no. 2, pp. 1086–1097, 2020.

[60] F. Zheng, X. Wang, L. Wang, X. Zhang, H. Zhu, L. Wang, and H. Zhang, "A fine-grained semantic alignment method specific to aggregate multi-scale information for cross-modal remote sensing image retrieval," *Sensors*, vol. 23, no. 20, p. 8437, 2023.

Alex Falcon Alex is a postdoctoral researcher at the University of Udine. He earned his PhD in Computer Science, Mathematics, and Physics through a joint program at Fondazione Bruno Kessler and the University of Udine. His research focuses on the automatic analysis of multimedia data, particularly its connection with natural language. He serves on the Technical Program Committee for major conferences like ACM MM and ECCV and has published extensively in multimedia and computer vision.



Ali Abdari Ali studied computer science at Kharazmi University. His research areas include image processing and machine learning. During his master's degree, he focused his research on action recognition in video, dedicating his thesis to utilizing compressed domain elements for action recognition in compressed videos. Now, he is a Ph.D. student at the University of Naples Federico II and University of Udine, Italy. His current research primarily focuses on 3D complex scenario retrieval.

Giuseppe Serra Giuseppe Serra has been an Associate Professor at the University of Udine since November 2019. He earned his Ph.D. in Computer Engineering, Multimedia, and Telecommunications from the University of Florence in 2010. Previously, he was an Assistant Professor at the University of Modena and Reggio Emilia and a visiting researcher at Carnegie Mellon University and Telecom ParisTech. His research focuses on Machine Learning and Deep Learning. He serves on the editorial boards of IEEE THMS and ACM TOMM and regularly reviews for top conferences and journals. He has published 70 papers in leading venues, and he leads the Artificial Intelligence Laboratory.