AdOCTeRA: Adaptive Optimization Constraints for improved Text-guided Retrieval of Apartments

Ali Abdari*
abdari.ali@spes.uniud.it
University of Naples Federico II
Naples, Italy
University of Udine
Udine, Italy

Alex Falcon
falcon.alex@spes.uniud.it
University of Udine
Udine, Italy

Giuseppe Serra
giuseppe.serra@uniud.it
University of Udine
Udine, Italy

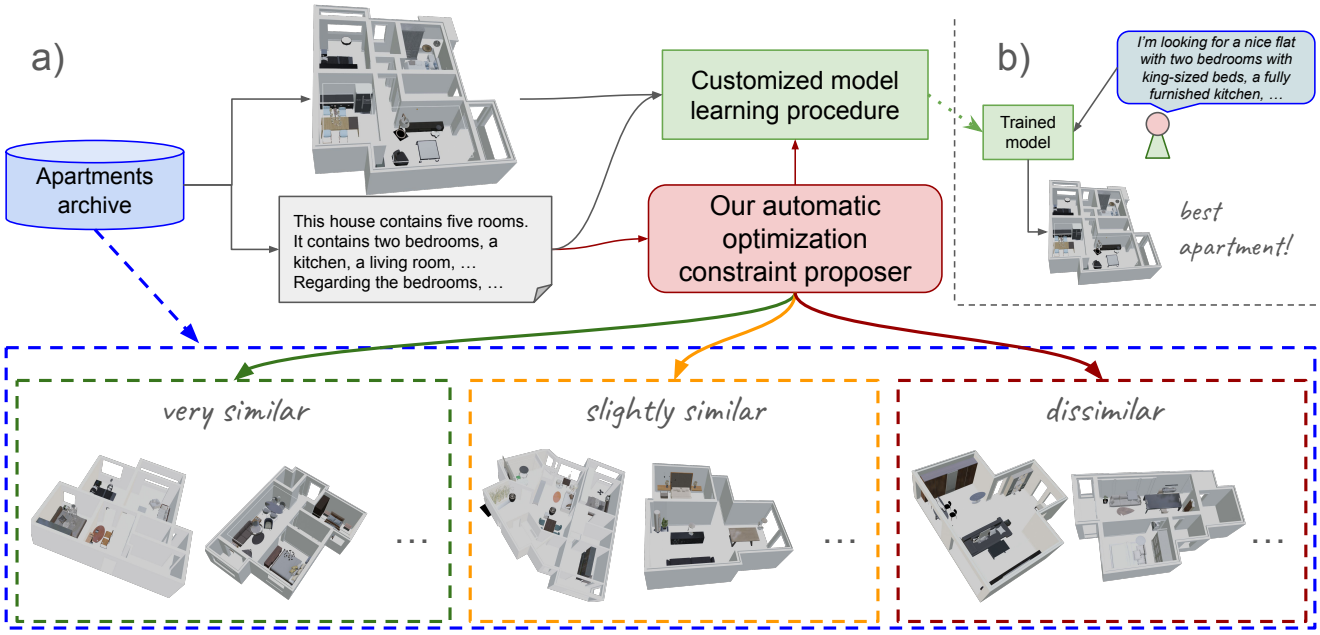

Figure 1: (a) High level view of the proposed method. It includes an automatic optimization constraint proposer, which automatically separates the apartments under analysis into three classes of similarity, useful for suggesting customized constraints for the learning procedure. (b) Given a user query, the system recommends the most fitting apartment.

ABSTRACT

Nowadays, it is common for workers to relocate to new countries while seeking better job opportunities, or to live as digital nomads. While doing so, they face the problem of finding a new place to call home, requiring them to trust online advertisements or to physically visit the apartment. Recently, the research community investigated the possibility of performing the search on the Metaverse, hence reducing time and costs related to traveling and limiting carbon emissions. The methods available are based on state-of-the-art cross-modal retrieval techniques, which learn a joint embedding space by mapping apartment-descriptions pairs close. However, these methodologies push all the other pairs far away in the embedding space. In this paper, we identify this decision as a limitation, since different apartments are likely to share many aspects. To overcome it, we propose AdOCTeRA, which automatically separates the apartments into three classes – very similar, slightly similar, and dissimilar – and proposes adaptive optimization constraints for each of them. We validate our methodology on a large dataset of more than 6000 apartments, obtaining considerable relative improvements over the previous state-of-the-art (+3.8% R@5 and +7.3% R@10), and consistent improvements over the baseline across all the experiments. The source code is available at https://github.com/aliabdari/AdOCTeRA.

*Corresponding author.

ICMR '24, June 10–14, 2024, Phuket, Thailand

ACM ISBN 979-8-4007-0619-6/24/06
https://doi.org/10.1145/3652583.3658039

CCS CONCEPTS

• **Information systems** → **Multimedia and multimodal retrieval**.

KEYWORDS

Text-Apartment Retrieval, Metaverse, Triplet loss, Custom loss function

ACM Reference Format:
Ali Abdari, Alex Falcon, and Giuseppe Serra. 2024. AdOCTeRA: Adaptive Optimization Constraints for improved Text-guided Retrieval of Apartments. In *Proceedings of the 2024 International Conference on Multimedia Retrieval (ICMR '24), June 10–14, 2024, Phuket, Thailand*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3652583.3658039

1 INTRODUCTION

It is becoming more and more common for workers, especially those involved in tech jobs, to frequently relocate from one nation to another while working remotely. These workers, known as "digital nomads", are as popular as ever, with about 17 million (11%) US workers now describing themselves as one[1], a number projected to grow to more than 60 million by 2030[2]. Not only tech workers, more traditional jobs are also starting to pursue this new type of living, such as lawyers and accountants. However, finding a new place to call home can be tiresome, while requiring the workers to either trust online advertisements or travel across countries to visit the advertised locations. This leads to undesired expenses, hours of traveling, and increased carbon emissions polluting the environment. To support all of them in their search from the comfort of their current location, a recent research field emerged, called text-to-apartment recommendation [1].

In the text-to-apartment recommendation scenario, the users interact with a system by querying it with a free-form description of their desires, and the system builds a ranked recommendation list of the most suitable apartments for their needs (Figure 1.b). To obtain a system for this task, in our previous work [1], we collected a dataset of virtual apartments, each paired to a textual description, and proposed an apartment recommendation system built on top of state-of-the-art cross-modal retrieval approaches, including CLIP [29]. Using deep learning techniques, it is possible to automatically learn two functions which map the input (apartment or description, respectively) into a joint apartment-description space. To achieve this goal, contrastive loss functions [15, 32] represent a key component. Contrastive loss functions guide the training by quantifying the distance between representations in the embedding space. Notably, the embedding space can capture relations between samples coming from a single modality [7, 32] or from multiple modalities, e.g., image-text [29], and video-text [27]. In these methodologies, it is common to consider the paired inputs, such as an apartment and its own description, relevant to each other, and completely irrelevant to any other example in the dataset. However, two apartments may be organized in a similar way and share most of the furniture, hence the training procedure should not treat them as completely irrelevant to each other since their similarities make them suitable for the same user query.

Therefore, in this paper, we propose an alternative learning methodology emphasizing this important aspect. The proposed method consists of a module which automatically creates learning constraints which are adaptive and dependent on the apartments under analysis. By doing so, we obtain a customized training procedure which implements more flexible constraints, allowing the model to grasp multiple levels of similarities between the examples. We hypothesize that a model trained with our procedure learns to distinguish between apartments which are very similar, partially similar, and dissimilar, as shown in Figure 1.a, and that these distinctions are fundamental to achieve better generalization. We verify this hypothesis on a large public dataset of more than 6000 apartments, obtaining considerable relative improvements of 3.8% and 7.3% in R@5 and R@10, respectively, demonstrating the effectiveness of the proposed approach. These results and the design choices made in our methodology are further corroborated by an extensive experimental setting.

In summary, the main contributions of this study can be described as follows:

- We identify an important shortcoming of previous approaches in text-to-apartment recommendation affecting the ability of models to properly separate apartments similar to each other from those partially similar or entirely dissimilar. We introduce a method, called AdOCTeRA, to overcome it by means of a module which proposes adaptive learning constraints at training time.
- We verify the effectiveness of the proposed approach on a large scale dataset of more than 6000 virtual apartments, obtaining considerable improvements (relative improvement of 3.8% and 7.3% in text-to-apartment R@5 and R@10) on previous state-of-the-art results.

The rest of the paper is organized as follows. Related works are highlighted and contextualized in Section 2. The proposed methodology is thoroughly described in Section 3. Section 4 explains the experimental setting and the research questions identified and discussed in this work, along with those which remain open. Finally, Section 5 draws the conclusion and plans the future work.

2 RELATED WORK

2.1 Text-based ranking of complex 3d scenes

Retrieving and organizing information based on textual queries has been a challenging research problem for many years. Specifically, the research community focused on using text to rank 2d images [9, 11, 20, 29] and videos [13, 23, 24, 27]. These efforts led to multiple advancements benefiting many other cross-modal applications, such as captioning [14, 37] and question answering [19, 26]. However, only recently the research community started to investigate the problem of ranking complex 3d scenes based on a textual query, drawing inspiration from the many advancements of cross-modal retrieval technologies [1–3, 38]. In fact, prior to these works, the retrieval of 3d scenes was guided by using several formats of visual information, e.g. images or sketches [4, 5, 39]. Single 3d objects, on the other hand, were retrieved both by using 2d or 3d artifacts [21, 25, 28] or, very recently, by text [22, 31]. However, ranking complex scenes is much more challenging than working on single objects, as the former includes tens or hundreds of objects, each of which may affect the relevance of the scene to the user query.

[1] https://www.mbopartners.com/state-of-independence/digital-nomads/
[2] https://www.wysetc.org/2023/01/growth-and-developments-in-the-digital-nomad-market-since-covid-19/

2.2 Contrastive loss functions

The recent approaches available in the literature to solve cross-modal tasks rely on learning to map the data under analysis into a joint embedding space. Contrastive loss functions, which represent a key component to achieve this objective, have been around for several years [6, 15] although they recently became more popular thanks to the impact obtained by SimCLR [7], MoCo [17], and CLIP [29]. By optimizing a contrastive loss function, models are incentivized to learn discriminative representations where paired examples in the dataset (e.g., an apartment and its description) are clustered together and unpaired ones are separated. This is done by working on two examples (paired or unpaired) at a time [15], three [32], four [8], or even more [35], looking to model increasingly complex inter- and intra-class relations. A frequent ingredient in these losses is the use of a margin hyperparameter, which constrains the desired distance between the examples in the embedding space. Although this margin is typically fixed and decided empirically, variable and adaptive solutions were also proposed. Zhang et al. proposed to start from a small margin and increase it monotonically during training to capture varying inter-class relations [40]. Hu et al. used the distance between paired and unpaired examples as the loss [18]. Semedo and Magalhaes linked the margin both to the epoch number and to the paired-unpaired distance [34]. Instead, He et al. proposed to define the margin in terms of a combination of similarity scores computed by frozen, pretrained models and trained models, giving more weight to the former in early training stages and more to the latter in later stages [16]. Falcon et al. proposed to define the margin in terms of a relevance score computed through part-of-speech tagging [12]. Differently from previous works, we introduce a method to automatically separate the examples into three similarity classes – very similar, slightly similar, and dissimilar – and vary the margin based on the selected class.

3 PROPOSED METHODOLOGY

An overview of the proposed methodology, AdOCTeRA, is shown in Figure 2. It is made of various components, including the apartment and textual representation modules, used to compute the text-vision representations; our Adaptive Optimization Constraints Proposer module; and finally the customized contrastive learning framework. The modules are thoroughly described in the following subsections.

3.1 Modeling the apartments and the descriptions

The apartment and textual representation modules are based on the state-of-the-art cross-modal retrieval approach CLIP [29]. To model the apartment, we learn a function f by using a Vision Transformer [10], followed by a one-dimensional convolutional neural network and a shallow MLP. To model the descriptions, we learn a function g by extracting for each of the sentences a representation by using a 12-layer Transformer [36], followed by a bidirectional GRU. This is done because the descriptions of the apartment can become very long, as they need to describe each room, and the furniture present in each of them. Note that both Transformers are jointly pretrained and frozen. A similar approach was followed in our previous work [1].

3.2 Adaptive Optimization Constraints Proposer

As shown in Figure 2, two main submodules compose our Adaptive Optimization Constraints Proposer: a textual-based apartments similarity function S, and the adaptive constraints proposer M. Both are described in the following subsections.

3.2.1 Textual-based similarity function S. The first component introduced in our methodology is a textual-based apartments similarity function, $S(\cdot,\cdot)$. Let x_1 and x_2 be the description of two apartments. To capture their similarity, we use a distilled version of RoBERTa trained for sentence similarity [30], obtaining a value representing the cosine similarity of x_1 and x_2. While its codomain is [0, 1], the distribution of the similarity values is data-dependent and may not span across that range completely, requiring a definition of the thresholds based on the dataset. To avoid this, a further normalization step is added to map the output values of the RoBERTa-based model S' such that the minimum similarity score is 0 and the maximum is 1. To do so, we apply a minmax normalization as follows:

$$S(x_1, x_2) = \frac{S'(x_1, x_2) - min}{max - min} \tag{1}$$

where *min* and *max* are the minimum and maximum values of S' computed on all the possible pairs from the training dataset.

3.2.2 The constraints proposer M_{τ_L,τ_U}. Once the similarity scores are computed, a new learning constraint can be proposed. A popular method frequently used in the literature is the triplet loss [32], which is formulated as follows:

$$L_T(x_a, x_p, x_n) = max(0, \Delta + s(r_n, r_a) - s(r_p, r_a)) \tag{2}$$

where Δ is a fixed real value, called margin, and r_* represents the representation extracted using f (respectively, g) for the apartment (resp., description). In particular, x_a, x_p, and x_n are respectively the anchor, the positive, and the negative elements of the triplet loss. The anchor and the positive are paired samples from the dataset (e.g., an apartment and its description), whereas the negative is another sample in the batch not related to them. However, this means that the same margin is enforced in all the constraints, hence all the negatives are considered equally, neglecting that different apartments may be almost identical or completely dissimilar.

Differently, we create adaptive constraints by establishing a relation between the margin in the contrastive loss function and the apartments/descriptions under analysis. In particular, as shown in Figure 1, we consider three similarity classes—namely, "very similar", "slightly similar", and "dissimilar"—in which the training samples are placed based on the similarity between the sample and the anchor using Eq. 1. We introduce two thresholds to characterize the classes, meaning that samples with a similarity greater than the "upper" threshold τ_U belong to the "very similar" class, those with a similarity smaller than the "lower" threshold τ_L belong to the "dissimilar" class, and finally the other samples fall into the "slightly similar" class. For each class, a different value for the margin will be proposed and used to define the optimization constraint. Formally, we define a function $M_{\tau_L,\tau_U}(\cdot,\cdot)$ which is parameterized by two threshold values, τ_L and τ_U, and returns a value for the margin, as follows:

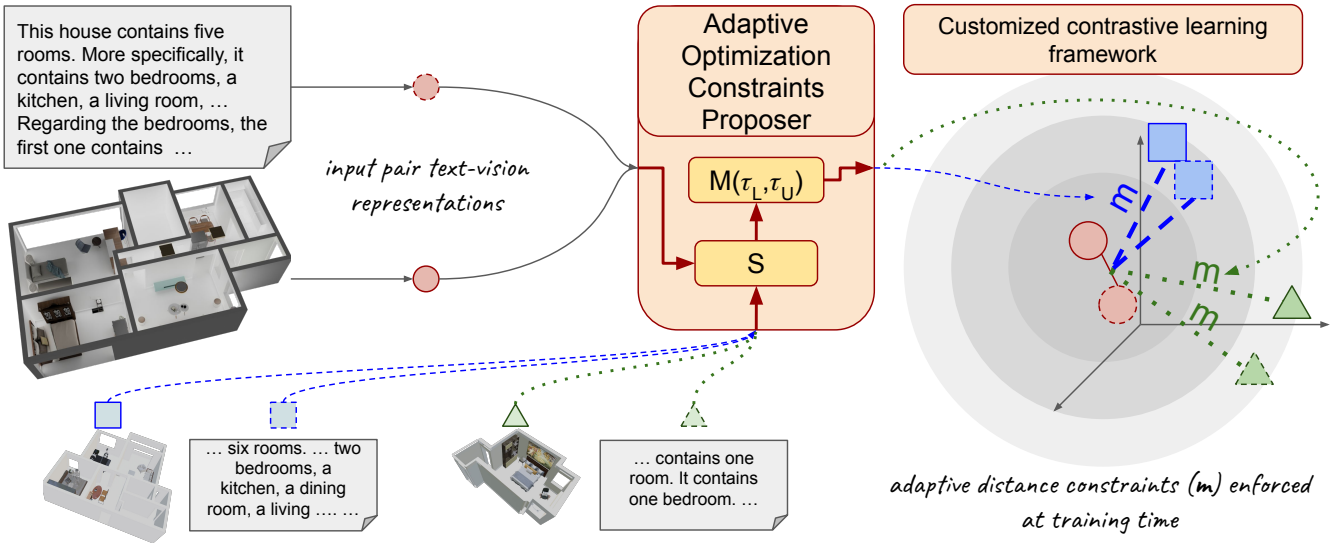

Figure 2: Overview of the proposed methodology. The apartment and textual representation modules, used to compute the input representations, are not shown here for simplicity. Details in Section 3.

$$M_{\tau_L,\tau_U}(x_1,x_2) = \begin{cases} m_1, & \text{if } S(x_1,x_2) > \tau_U \\ m_2, & \text{if } \tau_U > S(x_1,x_2) > \tau_L \\ m_3, & \text{if } \tau_L > S(x_1,x_2) \end{cases} \quad (3)$$

where m_1, m_2, and m_3 represent the three values for the margin used in the three similarity classes.

3.3 Contrastive learning framework

After defining M, each of the optimization constraints is formalized as follows, following a similar structure as the one used in the triplet loss:

$$L_{\tau_L,\tau_U}(x_a,x_p,x_n) = max(0, M_{\tau_L,\tau_U}(x_a,x_n) + s(r_n,r_a) - s(r_p,r_a)) \quad (4)$$

By doing so, we obtain a customized learning framework, in which every constraint is personalized for the apartments under analysis. Finally, the overall learning procedure introduces a constraint via Eq. 4 for each pair of samples from the dataset and one negative sample used for training, as in the following definition:

$$L_{AD;\tau_L,\tau_U} = \frac{1}{|B|}\sum_{A_a,D_p,D_n \in B} L_{\tau_L,\tau_U}(A_a,D_p,D_n) \quad (5)$$

$$L_{DA;\tau_L,\tau_U} = \frac{1}{|B|}\sum_{D_a,A_p,A_n \in B} L_{\tau_L,\tau_U}(D_a,A_p,A_n) \quad (6)$$

$$\mathcal{L}_{\tau_L,\tau_U} = L_{AD;\tau_L,\tau_U} + L_{DA;\tau_L,\tau_U} \quad (7)$$

where B represents the batch of samples randomly chosen from the full dataset to perform training, A and D are the sets of apartments and descriptions which provide the sampling pool for the anchors (A_a and D_a), positives (A_p and D_p), and negatives (A_n and D_n).

4 EXPERIMENTAL RESULTS

4.1 Dataset, baseline method, and evaluation metrics

The dataset under analysis [1] consists of more than 6000 apartments. Each apartment is paired to a textual paragraph describing the number and type of rooms, and the furniture in each of them. Hence, the paragraphs can be quite long and detailed, containing on average 16 sentences and 319 words. Each apartment can be accessed as a 3d scene or as a set of pre-extracted images. An example is shown in Figure 3.

The baseline method used in our methodology is taken from the one adopted in our previous paper [1], called CNV. CNV uses the same visual and textual representation modules detailed in our methodology (Sec. 3.1), but uses a standard triplet loss function to learn the joint embedding space. In our implementation, we use the same hyperparameters chosen by the previous authors to have more comparable results. Apart from CNV, other three methods introduced in [1] are considered for state-of-the-art comparison, including two simple baselines comparing pooled descriptors either without performing any learning (NLB) or by learning a MLP (AFN), and a method performing multitask learning on top of CNV (FaRMaRE).

We follow the same split used in our previous work, resulting in 4256, 912, and 913 apartments for train, validation, and test sets. We select the best model on the validation set and use it to assess the performance on the test set, including standard metrics commonly used in cross-modal retrieval scenarios. The recall rates, R@k, with k set to 1, 5, and 10, quantify how frequently the groundtruth is found within the top k elements of the ranked list. The sum of recalls (Rsum) is also reported. The median rank measures the position of the groundtruth in the ranked list. While in the first experiments, we only report the text-to-apartment retrieval performance for simplicity, the comparison to state-of-the-art methods

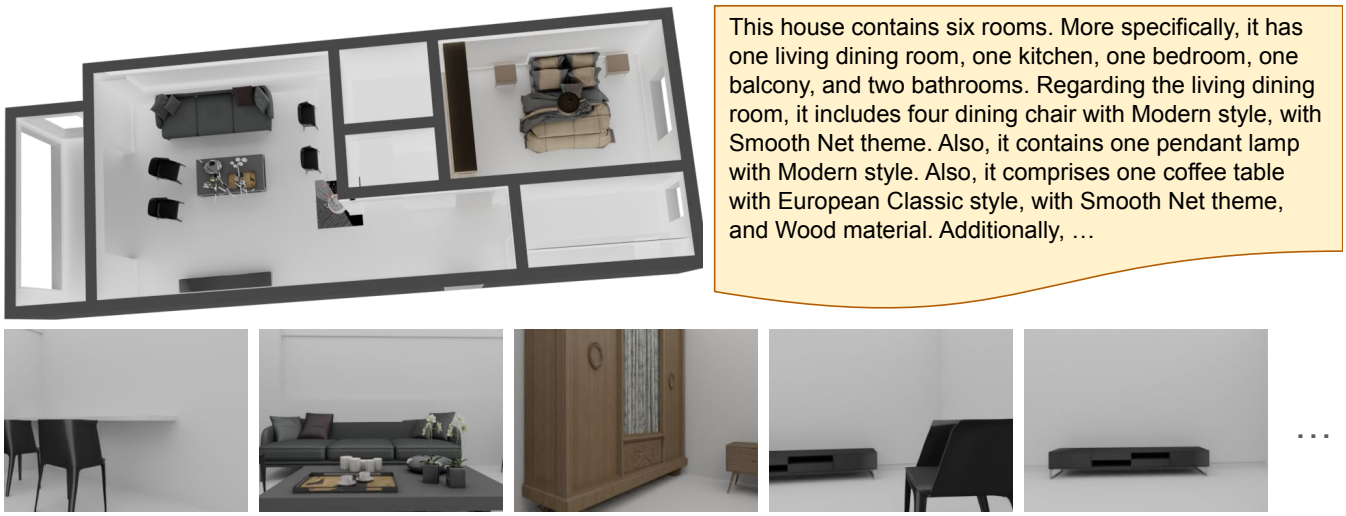

Figure 3: An example taken from the dataset under analysis. It includes the 3d scene (shown here from above), a few examples of the pre-extracted images, and a few sentences from the full description.

also reports the apartment-to-text retrieval performance for comprehensiveness. Notably, all the experiments are repeated three times and the average performance is reported.

4.2 Implementation details

In the proposed method, τ_L and τ_U are set to 35% and 75%, whereas the margins m_1, m_2, and m_3 are set to 0.25, 0.30, and 0.35, respectively. The Vision Transformer used to implement f consists of a ViT-B-32 [10], and it is jointly pretrained with the 12-layer Transformer (learning function g) via CLIP on the LAION-2B dataset, a subset of the bigger 5-billions images dataset [33]. To implement S', we use the 'paraphrase-distilroberta-base-v2' model from the SentenceTransformers environment.

We use PyTorch 2.2.0 for the implementation and run all the experiments on a machine using an RTX A5000 GPU, 32 GB of RAM, and an Intel Core i7-9700K. The batch size is 64 and the training can last for 50 epochs, while an early stopping mechanism has been considered to stop the training procedure if the loss does not decrease by at least .0001 on the validation set for 25 epochs. The Adam optimizer has been used, and the learning rate starts from .008 and is decayed by a factor of 25% after 27 epochs.

4.3 How impactful are the thresholds τ_L and τ_U?

The two thresholds introduced in our methodology represent an important hyperparameter, as they decide how to separate the apartments into the three classes under analysis, that is "very similar", "slightly similar", and "dissimilar". We investigate the importance of these thresholds both in a qualitative and quantitative way. First, in Figure 4 we report the frequency of the similarity values computed with S during one epoch of training, with random batches of size 64. It can be seen that most of the apartments are quite similar to each other, hence the thresholds should also try to separate them accordingly while capturing fine grained details which discriminate them. To make a precise decision, we fix the margins m_1, m_2, and m_3 to 0.25, 0.30, 0.35 respectively and empirically evaluate different values for both thresholds, starting by using [25%, 35%, 45%, 55%] for τ_L and then using [50%, 75%, 90%] for τ_U after fixing τ_L. The results are presented in Table 1.

Varying τ_L has a considerable effect on the performance (e.g. R@10 varies from 50.6 to 53.0), capturing from about 1% of the training samples, when $\tau_L = 25\%$ to about 17% when $\tau_L = 55\%$ (Fig. 4). While the two extremes ($\tau_L = 25\%$ and $\tau_L = 55\%$) lead to small improvements (+1.0% R@5 and +0.7% R@10, +1.3% R@5 and +0.5% R@10, respectively), the intermediate values lead to much better results. In particular, +3.7% R@5 and +4.1% R@10 are obtained on top of the baseline when $\tau_L = 35\%$. This suggests that identifying the apartments which are very different from the anchor (i.e., their descriptions have a S-similarity lower than τ_L) and treating them differently than those which are "slightly similar" or "very similar" during training has a positive effect, compared to treating all the apartments in the same way (i.e., enforcing the same distance, represented by the fixed margin Δ) as done by the baseline.

Looking at the upper threshold, both 50% and 90% do not lead to improved performance compared to using $\tau_U = 75\%$. A small value for τ_U (e.g. 50%) leads to worse performance, possibly because too many apartments are identified as very similar (almost 90%, see Fig. 4). On the other hand, $\tau_U = 90\%$ is too strict, and very few additional apartments are identified as very similar (about 4%). While it leads to better results than the baseline (e.g. median rank goes from 11 to 8.7), the performance is not as good as when $\tau_U = 75\%$.

4.4 What happens if the margins are varied?

After fixing the thresholds to $\tau_L = 35\%$ and $\tau_U = 75\%$, this experiment investigates the effect of varying the margins. In particular, we explore different combinations of margins by keeping m1 fixed, increasing m2 by 0.03, 0.05, .10, and .15, and by increasing m3 with

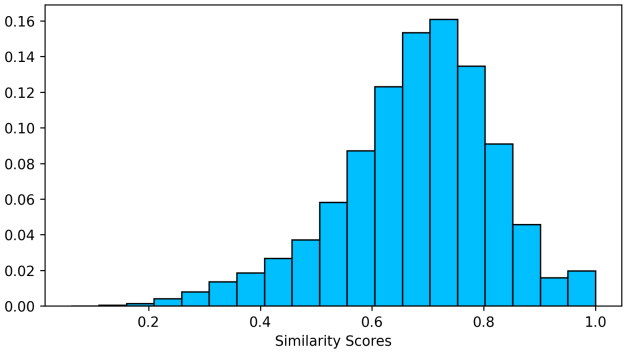

Figure 4: Distribution of similarity values computed with S during one epoch of training. Details in Sec. 4.3.

		Text-Apartment			
		R1	R5	R10	MedR
baseline		23.7	40.6	48.9	11
τ_L	τ_U				
25%	75%	22.9	41.6	50.6	10.0
35%	75%	24.5	**44.3**	**53.0**	**8.3**
45%	75%	**25.0**	43.9	52.2	9.0
55%	75%	23.7	41.9	50.5	10.0
35%	50%	22.3	40.7	49.6	10.7
35%	75%	**24.5**	**44.3**	53.0	**8.3**
35%	90%	23.0	42.9	**53.1**	8.7

Table 1: Performance observed varying the thresholds τ_L and τ_U used during training to separate the samples in different similarity classes. Discussion in Sec. 4.3.

either the same increments as m2 or greater than that (e.g. when increasing m2 by .05, m3 is increased by .10 with respect to m2). The results, measured with the median rank, are reported in Table 2.

Overall, it can be seen that almost all the different combinations achieve a lower median rank (as low as 8.3) compared to the baseline (11). Note that the baseline follows the standard triplet loss, using a fixed value for the margin Δ which is used in every optimization constraint (refer to Eq. 2). Differently, our methodology adapts the margin to the samples under analysis, as described in Section 3. The results in Table 2 suggest that having different margins to separate the apartments in multiple similarity classes is indeed useful for capturing fine grained details. In particular, we find the best result using $m_1 = 0.25$, $m_2 = 0.30$, $m_3 = 0.35$, $\tau_L = 25\%$, and $\tau_U = 75\%$, obtaining a median rank of 8.3 (-2.7 compared to the baseline, a relative improvement of 24.5%).

4.5 How impactful is the definition of S?

In the methodology, the similarity of the descriptions of the apartments is captured by a function S, implemented with recent NLP techniques. In this experiment, we investigate a different definition for such an important function. In particular, we define it in terms of

			Text-Apartment (MedR)					
τ_L			25%	35%	45%	35%		
τ_U			75%			50%	75%	90%
baseline ($\Delta = 0.25$)			11					
m_1	m_2	m_3						
0.25	0.28	0.31	10.0	8.7	8.7	9.0	8.7	10.0
0.25	**0.30**	**0.35**	10.0	**8.3**	9.0	10.7	**8.3**	8.7
0.25	0.30	0.40	9.0	10.3	11.7	9.0	10.3	10.3
0.25	0.35	0.45	9.0	9.7	9.3	11.7	9.7	9.0
0.25	0.35	0.50	10.0	9.3	8.7	9.7	9.3	8.7
0.25	0.40	0.55	9.3	8.7	11.0	8.3	8.7	10.3
0.25	0.40	0.60	9.3	10.7	9.3	10.7	10.7	10.3

Table 2: Performance observed varying the margins m_1, m_2, and m_3 used in the optimization constraints during training. The baseline follows standard triplet loss with a fixed value (0.25) for the margin Δ. Discussion in Sec. 4.3.

	Text-Apartment			
	R1	R5	R10	MedR
baseline	23.7	40.6	48.9	11
S_2	23.4	42.2	51.4	9.7
S (ours)	**24.5**	**44.3**	**53.0**	**8.3**

Table 3: Performance observed varying the definition of the similarity function S. Discussion in Sec. 4.5.

Intersection-over-Union (IoU) of the rooms of the two apartments under analysis as follows:

$$S_2(x_1, x_2) = \frac{|R(x_1) \cap R(x_2)|}{|R(x_1)| + |R(x_2)| - |R(x_1) \cap R(x_2)|} \tag{8}$$

where $R(\cdot)$ extracts a list of the type of rooms within the apartment. For instance, if the description contains "In this apartment, there are three bedrooms, one kitchen, one dining room, one bathroom, and one living room" then, $R(\cdot)$ is the list containing "(Bedrooms, 3)", "(Kitchen, 1)", "(Dining room, 1)", "(Bathroom, 1)", and "(Living room, 1)". The results are reported in Table 3.

Two results are outlined. First, the proposed approach (S) achieves better performance than the alternative function analyzed in this experiment (S_2), obtaining for instance 53.0% R@10 compared to 51.4%. Second, even though S_2 achieves lower results than S, it still leads to better results compared to the baseline, obtaining 51.4% R@10 compared to 48.9%. This result further highlights the effectiveness of the proposed method in capturing finer-grained details of the apartments and their descriptions.

4.6 How does AdOCTeRA compare to the state-of-the-art?

In this analysis, we compare the results obtained by the proposed methodology to those tested in our previous paper [1]. In particular, NLB is a simple baseline which extracts CLIP features for both the visual and textual data and compares their similarity to obtain a ranking. AFN learns a simple function through an MLP after pooling the visual descriptors obtained for the apartment. CNV is

the baseline used in our method. Finally, FaRMaRE uses a multitask learning setting, combining ranking and classification objectives to learn better descriptors for the apartments.

As can be seen in Table 4, FaRMaRE obtains slightly better R@1 in both text-to-apartment (24.8 compared to 24.5) and apartment-to-text scenarios (25.7 compared to 25.0). However, all the other metrics show a significant improvement compared to it, obtaining for instance +3.9% R@10 (53.0 compared to 49.1) and 2.7 lower median rank.

4.7 Limitations and future work

An important limitation of this work consists of the usage of textual-only information to quantify the similarity of the two apartments. In fact, spatial information may also be important to capture finer grained details of the furniture and their placement within the apartment. We performed some early experimentation with cross-modal models, like CLIP, as the key component of the similarity function S, but we observed unreliable similarity values. While we did not report results in this regard, the effect on the performance is considerable, as can also be observed in the retrieval results obtained by the NLB baseline (Sec. 4.6). Notably, this problem may be related to the large number of tokens in the descriptions of the apartments, which lead to noisy representations when pooled together. Therefore, further research is required to better understand how to implement a reliable cross-modal similarity function.

As for the future work, we identify two main directions. First, in this work, we divided the available scenes into three categories, each with its own margin and related optimization constraint. However, in future work we also plan to investigate the effect of incorporating fewer classes, leading to a coarser and more inclusive approach; and, vice versa, more classes, aiming at capturing finer grained differences. Second, while considerable improvements were obtained by using RoBERTa as the base model for our similarity function, newer methods were developed over the years, further pushing the potential of models intended for sentence similarity. Therefore, their integration could prove very effective. Nonetheless, further research should also be done in this regard, as these models are often limited in the number of tokens they can effectively process and understand, which may affect the effectiveness of longer descriptions.

5 CONCLUSIONS

The ability of finding apartments fitting the user interests is gaining interest as it raises the possibility of visiting apartments virtually, in the Metaverse, without needing to physically move. This is both comfortable, as it avoids hours of traveling, and a green solution, reducing the carbon emissions otherwise inevitable. Previous works on this problem adapted state-of-the-art solutions typically used in cross-modal retrieval approaches, such as CLIP [29]. However, different apartments may share lots of characteristics, hence limiting the effectiveness of popular solutions based on contrastive learning, in which paired (cross-modal) examples in the dataset are pulled closer in the embedding space whereas unpaired ones are pushed far away. To overcome this limitation, we proposed a methodology, AdOCTeRA, which automatically separates training samples into three separate classes – very similar, slightly similar, and dissimilar – and proposes adaptive optimization constraints. By doing so, AdOCTeRA promotes a more structured organization of the embedding space, and the experimental results demonstrate the effectiveness of its design. In fact, on a large dataset of more than 6000 apartments, it obtains large relative improvements on the previous state-of-the-art, e.g., +3.8% R@5 and +7.3% R@10. Moreover, the extensive experimental setting provided evidence of its robustness to changes in the main hyperparameters and consistent improvements over the baseline.

ACKNOWLEDGMENTS

This work was supported by MUR Progetti di Ricerca di Rilevante Interesse Nazionale (PRIN) 2022 (project code 2022YTE579) and by the Department Strategic Plan (PSD) of the University of Udine – Interdepartmental Project on Artificial Intelligence (2020-25).

REFERENCES

[1] Ali Abdari, Alex Falcon, and Giuseppe Serra. 2023. 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*. 4293–4303.

[2] Ali Abdari, Alex Falcon, and Giuseppe Serra. 2023. Metaverse Retrieval: Finding the Best Metaverse Environment via Language. In *Proceedings of the 1st International Workshop on Deep Multimodal Learning for Information Retrieval*. 1–9.

[3] Ali Abdari, Alex Falcon, and Giuseppe Serra. 2024. A Language-based solution to enable Metaverse Retrieval. In *International Conference on Multimedia Modeling*. Springer, 477–488.

[4] Hameed Abdul-Rashid, Juefei Yuan, Bo Li, Yijuan Lu, Song Bai, Xiang Bai, Ngoc-Minh Bui, Minh N Do, Heyu Zhou, Yang Zhou, et al. 2018. SHREC'18 track: 2D image-based 3D scene retrieval. *Training* 700 (2018), 70.

[5] Hameed Abdul-Rashid, Juefei Yuan, Bo Li, Yijuan Lu, Tobias Schreck, Ngoc-Minh Bui, Trong-Le Do, Mike Holenderski, Dmitri Jarnikov, Khiem T Le, et al. 2019. Shrec'19 track: Extended 2D scene image-based 3D scene retrieval. *Training (per class)* 700 (2019), 70.

[6] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature Verification Using a "Siamese" Time Delay Neural Network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*. 737–744.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[8] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. 2017. Beyond triplet loss: a deep quadruplet network for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 403–412.

[9] Sanghyuk Chun, Seong Joon Oh, Rafael Sampaio De Rezende, Yannis Kalantidis, and Diane Larlus. 2021. Probabilistic embeddings for cross-modal retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8415–8424.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

[11] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2017. Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612* (2017).

[12] Alex Falcon, Swathikiran Sudhakaran, Giuseppe Serra, Sergio Escalera, and Oswald Lanz. 2022. Relevance-based margin for contrastively-trained video retrieval models. In *Proceedings of the 2022 International Conference on Multimedia Retrieval*. 146–157.

[13] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. 2020. Multi-modal transformer for video retrieval. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*. Springer, 214–229.

[14] Difei Gao, Luowei Zhou, Lei Ji, Linchao Zhu, Yi Yang, and Mike Zheng Shou. 2023. MIST: Multi-modal Iterative Spatial-Temporal Transformer for Long-form Video Question Answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14773–14783.

[15] Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on*

Method	Text-Apartment				Apartment-Text				Rsum
	R1	R5	R10	MedR	R1	R5	R10	MedR	
NLB [1]	0.1	0.6	1.3	413	0.6	1.9	3.7	339	8.2
AFN [1]	10.6	25.5	34.2	29	8.9	23.2	31.3	31	133.7
CNV [1]	23.7	40.6	48.9	11	23.1	40.7	48.9	12	225.9
FaRMaRE [1]	**24.8**	42.6	49.1	11	**25.7**	41.4	49.1	11	232.7
AdOCTeRA (**ours**)	24.5	**44.3**	**53.0**	**8.3**	25.0	**42.9**	**52.0**	**9.0**	**241.8**

Table 4: Comparison with state-of-the-art on the apartments dataset from [1]. Details in Section 4.6.

computer vision and pattern recognition (CVPR'06), Vol. 2. IEEE, 1735–1742.

[16] Feng He, Qi Wang, Zhifan Feng, Wenbin Jiang, Yajuan Lü, Yong Zhu, and Xiao Tan. 2021. Improving Video Retrieval by Adaptive Margin. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1359–1368.

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9729–9738.

[18] Sixing Hu, Mengdan Feng, Rang MH Nguyen, and Gim Hee Lee. 2018. Cvm-net: Cross-view matching network for image-based ground-to-aerial geo-localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7258–7267.

[19] Xiaowei Hu, Zhe Gan, Jianfeng Wang, Zhengyuan Yang, Zicheng Liu, Yumao Lu, and Lijuan Wang. 2022. Scaling up vision-language pre-training for image captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 17980–17989.

[20] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*. PMLR, 4904–4916.

[21] Longlong Jing, Elahe Vahdani, Jiaxing Tan, and Yingli Tian. 2021. Cross-modal center loss for 3d cross-modal retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3142–3151.

[22] Trung-Nghia Le, Tam V Nguyen, Minh-Quan Le, Trong-Thuan Nguyen, Viet-Tham Huynh, Trong-Le Do, Khanh-Duy Le, Mai-Khiem Tran, Nhat Hoang-Xuan, Thang-Long Nguyen-Ho, et al. 2023. TextANIMAR: text-based 3D animal fine-grained retrieval. *Computers & Graphics* 116 (2023), 162–172.

[23] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. 2021. Less is more: Clipbert for video-and-language learning via sparse sampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 7331–7341.

[24] Xirong Li, Chaoxi Xu, Gang Yang, Zhineng Chen, and Jianfeng Dong. 2019. W2vv++ fully deep learning for ad-hoc video search. In *Proceedings of the 27th ACM international conference on multimedia*. 1786–1794.

[25] Dongyun Lin, Yiqun Li, Yi Cheng, Shitala Prasad, Tin Lay Nwe, Sheng Dong, and Aiyuan Guo. 2022. Multi-view 3D object retrieval leveraging the aggregation of view and instance attentive features. *Knowledge-Based Systems* 247 (2022), 108754.

[26] Kevin Lin, Linjie Li, Chung-Ching Lin, Faisal Ahmed, Zhe Gan, Zicheng Liu, Yumao Lu, and Lijuan Wang. 2022. Swinbert: End-to-end transformers with sparse attention for video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 17949–17958.

[27] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. 2022. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing* 508 (2022), 293–304.

[28] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 652–660.

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

[30] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. http://arxiv.org/abs/1908.10084

[31] Yue Ruan, Han-Hung Lee, Yiming Zhang, Ke Zhang, and Angel X Chang. 2024. TriCoLo: Trimodal Contrastive Loss for Text To Shape Retrieval. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 5815–5825.

[32] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 815–823.

[33] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. 2022. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems* 35 (2022), 25278–25294.

[34] David Semedo and João Magalhães. 2019. Cross-Modal Subspace Learning with Scheduled Adaptive Margin Constraints. In *Proceedings of the 27th ACM International Conference on Multimedia*. 75–83.

[35] Kihyuk Sohn. 2016. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems* 29 (2016).

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*.

[37] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. 2021. Just ask: Learning to answer questions from millions of narrated videos. In *Proceedings of the IEEE/CVF international conference on computer vision*. 1686–1697.

[38] Fuyang Yu, Zhen Wang, Dongyuan Li, Peide Zhu, Xiaohui Liang, Xiaochuan Wang, and Manabu Okumura. 2024. Towards Cross-Modal Point Cloud Retrieval for Indoor Scenes. In *International Conference on Multimedia Modeling*. Springer, 89–102.

[39] Juefei Yuan, Hameed Abdul-Rashid, Bo Li, and Yijuan Lu. 2019. Sketch/image-based 3D scene retrieval: Benchmark, algorithm, evaluation. In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE, 264–269.

[40] Yingying Zhang, Qiaoyong Zhong, Liang Ma, Di Xie, and Shiliang Pu. 2019. Learning incremental triplet margin for person re-identification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 9243–9250.