

BiListing: Modality Alignment for Listings

Guillaume Guy
guillaume@airbnb.com
Airbnb
San Francisco, CA, USA

Chun How Tan
chunhow.tan@airbnb.com
Airbnb
San Francisco, CA, USA

Mihajlo Grbovic
mihajlo.grbovic@airbnb.com
Airbnb
San Francisco, CA, USA

Han Zhao
han.zhao@airbnb.com
Airbnb
San Francisco, CA, USA



Figure 1: BiListing embeddings alignment

Abstract

Airbnb is a leader in offering travel accommodations. Airbnb has historically relied on structured data to understand, rank, and recommend listings to guests due to the limited capabilities and associated complexity arising from extracting meaningful information from text and images. With the rise of representation learning, leveraging rich information from text and photos has become easier. A popular approach has been to create embeddings for text documents and images to enable use cases of computing similarities between listings or using embeddings as features in an ML model.

However, an Airbnb listing has diverse unstructured data: multiple images, various unstructured text documents such as title,

description, and reviews, making this approach challenging. Specifically, it is a non-trivial task to combine multiple embeddings of different pieces of information, i.e. each image, each review, etc., to reach a single meaningful listing representation, especially if some of the embeddings lie in different spaces. Faced with such a problem, practitioners often resort to unprincipled approaches of averaging embeddings to produce a single one. However, this often results in an inaccurate representation due to loss of information in the averaging process.

This paper proposes BiListing, for Bimodal Listing, an approach to align text and photos of a listing by leveraging large-language models and pretrained language-image models. The BiListing approach has several favorable characteristics: capturing unstructured data into a single embedding vector per listing and modality, enabling zero-shot capability to search inventory efficiently in user-friendly semantics, overcoming the cold start problem, and enabling listing-to-listing search along a single modality, or both.

We conducted offline and online tests to leverage the BiListing embeddings in the Airbnb search ranking model, and successfully deployed it in production, achieved 0.425% of NDCB gain, and drove tens of millions in incremental revenue.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CIKM '25, November 10–14, 2025, Seoul, Republic of Korea

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-2040-6/2025/11...\$15.00
<https://doi.org/10.1145/3746252.3761577>

Keywords

CLIP, Large Language Model, Contrastive Learning, Multimodality alignment, Airbnb , Airbnb listing, Pretrained embedding

ACM Reference Format:

Guillaume Guy, Mihajlo Grbovic, Chun How Tan, and Han Zhao. 2025. BiListing: Modality Alignment for Listings. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25), November 10–14, 2025, Seoul, Republic of Korea*. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3746252.3761577>

1 Introduction

In this section we introduce the problem, historical challenges and limitations of the existing solutions and our main contribution. In section 2, we describe related work in the area of item embeddings as well as multimodal representations. Section 3 outlines our approach in detail. In section 4 we present the results, both quantitatively and qualitatively. Finally, in section 5, we describe practical considerations for successfully applying BiListing embeddings as features in the Airbnb search ranking model and demonstrate how multimodal embeddings can significantly improve guest search experience backed by a strong lift in bookings on Airbnb .

1.1 Historical Context

To generate listing representations, Airbnb has historically relied on user logs, i.e. co-clicked and co-wishlisted listings leading to bookings [7], and less on unstructured data. However, user research shows that a guest spends a third of their time inspecting the images alone. The main challenge is to combine all the diverse listing content, i.e. images of different aspects and spaces of a home, title, description, and reviews, into a single representation.

For example, a listing may have 60 images, some of the backyard with lush garden and a pool, and some of the living room with Scandinavian furniture. Now consider that it also has a title and description which talk about the location, how to get there and finally reviews that talk about guest experiences. These are all very different aspects of the same listing that need to be meaningfully combined into a single representation. Averaging embeddings is inadequate because some aspects are more dominant than others.

Recent advancements in contrastive learning allowed us to think about the problem as an alignment one, but there were still many practical challenges that needed to be solved which led to the work in this paper.

1.2 CLIP and its limitations

With large-scale pretraining using image-text pairs [19] (see Figure 2) CLIP demonstrated that with orders of magnitude more web-sourced data [21], it can provide semantic understanding of images, leading to a wide array of use cases such as image search using free text queries (see Figure 3).

CLIP is trained using two encoders, one for captions and one for images in a 1:1 relationship. The output of both are then projected into a joint latent space, with the objective of bringing entities belonging to the same pair closer in the latent space.

Adapting CLIP to Airbnb’s listings has several challenges:

- CLIP only applies to a single photo:text pair whereas a listing has multiple photos and pieces of text. Therefore, free text

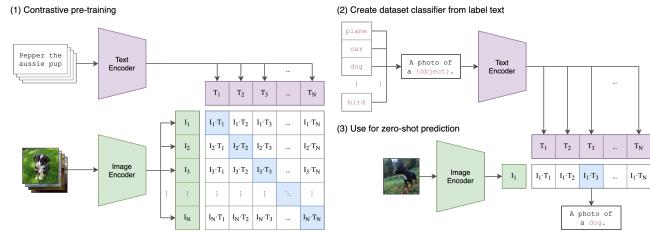


Figure 2: Contrastive learning diagram [19]



Figure 3: Illustration of an image result using from query: "A stunning steel-and-glass home in the Yucca Valley"

listing searches such as "Home in the woods with a hot tub, sauna, and large screen TV" that cover multiple listing aspects that are potentially featured in different photos can lead to poor results.

- CLIP text encoder context is only 77 due to the inherent nature of captions which are usually single sentences, with front-loaded content. Listing descriptions alone are much longer and less front-loaded.
- Pretrained text encoders are semantically tied to photo captioning with formats such as "a photo of {}" (e.g. "a photo of a dog") which is an unnatural formulation for listing search application where queries would be more phrased like "A 2 bedroom villa in {} featuring {}"

1.3 Our contribution

In this paper, we propose BiListings, an approach to align text and photos of a listing by leveraging large-language models and pre-trained language-image models. Specifically, we propose adapting the Contrastive Learning Framework [19] or CLIP to incorporate multiple aspects of the same listing, i.e. images of different parts of the listing and text attributes such as title, description and reviews that capture different aspects of the listing.

This paper is the first of its kind to expand the image-pair paradigm to more complex multimodal entities, and does so in a way that is easily reproducible by other practitioners in related applications where multiple aspects of the same item, that are possibly of

different nature, need to be combined into a single item representation.

Furthermore, we present detailed steps to implement and reproduce our approach. We believe that intuitive and easily reproducible publications are important given the reproducibility crisis with some studies claiming that only a fraction of Machine Learning papers were reproducible[2]

2 Related Work

OpenAI formulated the original CLIP problem in 2021 [19]. Since, multiple improvements have been made. Zhang et al. [25] proposed another alignment process with coarse components to enhance the length of the CLIP context, but with added complexity.

In 2022, Flamingo [1] introduced a multimodal model capable of processing interleaved vision and text inputs for tasks such as visual question answering. However, its architecture poses three challenges for industrial-scale listing representation: (1) it is optimized for interleaved vision-language tasks rather than per-entity embedding use cases like retrieval and ranking; (2) it processes only a limited number of images—typically 5 during training—whereas listings on our platform may contain 64+ images; and (3) it relies on a large decoder-style architecture, making it more resource-intensive than BERT-style encoders, making Flamingo unsuitable for our latency- and throughput-constrained production use case. Similarly, BLIP [14] only focuses on single image entities.

Video retrieval provides another interesting perspective, as videos are made up of frames, similarly to listings. Portillo-Quintero et al. [18] experimented with mean-pooling of CLIP embeddings across video frames, showing strong video retrieval performance. Luo et al. [15] went further and introduced a self-attention mechanism across video frames to learn temporal dependency, showing improved performance.

Furthermore, Faysse et al. [5] proposed to see a document as a collection of patches, from which we can extract embeddings. While enabling multimodal representation, this approach still suffers from not having a unified embedding representation.

Finally, our work addresses the limitations of the approach presented in Grbovic et al. [7]. In that work the authors leverage user action sequence, such as clicks, wishlists, and bookings, to train high-quality listing embeddings. However, the approach suffers from a cold-start problem for listings that do not appear in many sessions. Moreover, the resulting embeddings can only be used to calculate similarities to other embeddings and for recommendation applications and are not searchable with natural language queries. In our work, we present an approach that is effectively able to use a large share of the listing content information from different modalities to create an embedding that is then both searchable with natural language queries and also usable for similarity-based recommendation applications.

3 Approach

3.1 Objective

We propose to formulate the Listing understanding as a modality alignment problem. As hosts upload their listing on Airbnb, the text, as well as the structured data they input provide details of the listings. See example in Table 1

Inputs	
Textual Inputs	Visual Inputs
Title: The Graham Residence on 20 Acres	
Description: Recently remodeled with gorgeous limestone floors and tiled the bathroom. The 1250-square-foot steel-and-glass home features two bedrooms and one bathroom. A generous living room has ample seating options and stunning views out of multiple sliding glass doors. Outside are two large decks with comfortable furniture and an outdoor dining table. A Jacuzzi-brand hot tub sits away from the house on a rock outcrop offering views in all directions. Other outdoor amenities include a propane [...]	
	

Table 1: side-by-side comparison of the multimodal inputs on a listing

The text and photos are not randomly associated. In fact, hosts strive to give the salient details of their listings, featuring amenities, views, location, and vibe that transpire from the photos. In this example, the host highlights the recent renovation, with the materials (limestone floors and tiled bathroom), the sliding windows, and the expectations of quiet and wide surroundings (20 acres).

With this insight, we can take advantage of Airbnb’s large database of listings as training data without additional labels and adapt the CLIP image-text framework to train multimodal embeddings of a listing.

3.2 Architecture

We propose to reuse the 2 CLIP text-vision encoder tower approach in our BiListing model, as depicted in Figure 4. Compared to the standard CLIP model, we added three novel components to enable it to work at the listing level:

- In the vision tower, a trainable projection layer, using a Transformer architecture named "PhotoSet transformer", that consolidates multiple photo embeddings (up to 64¹) into a single projection
- An aggressive embedding compression logic using Optimal Product Quantization (OPQ) to optimize bandwidth consumption
- The generation of Visual Profiles, which summarize the listing’s text description grounded with CV indicators into a short, dense, visual description of the listing

Zooming into the vision tower, our experiments show that a transformer encoder with positional encoding performs best. With this architecture, a dense embedding of an image can be interpreted

¹We chose 64 as 98 percent of Airbnb’s listings have 64 or less photos

as a token and all the photos make up a sentence. Within such a comparison, the EOT token is simply the embedding output of the last photo. In practice, positional encoding captures the fact that hosts tend to add the most important photos first, and photo of the surrounding areas last.

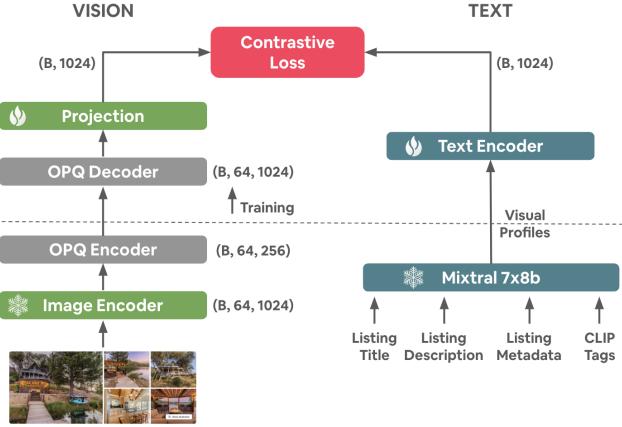


Figure 4: BiListing architecture using CLIP-style 2 tower architecture with vision (left) and text (right)

The pseudo-code associated with the proposed architecture is formulated in Algorithm 1.

3.3 Visual Text Profiles

Airbnb collects a large number of disparate textual information (see 2): listing title, description, location but also Airbnb-generated signals (e.g. CLIP implicit signals, detected amenities) and structured data which all need to be aggregated into single, semantically correct, and natural prompt. Moreover, the prompt must contain only visual elements to facilitate the alignment with photos. For instance, Air Conditioning, or nearby POIs, are not visible features and should be excluded.

To this end, we used a Large Language Model to generate summarized, dense text descriptions of listings capturing the visual salient points for guests, using the words of the hosts. We denote the output "visual profiles", as they represent a consolidated summary of the visual aspect of the listings.

Specifically, we used Mixtral 8x7b [10] in a zero-shot setting with a step-by-step recipe prompt for its open-weight availability and good balance inference cost vs. quality. We found that prefilling the response with "A N bedroom, M bathroom" produces the best outcome.

As an illustration, the following output was generated for the listing from Table 1: "A stunning 2 bedroom 1250-square-foot recently-remodeled steel-and-glass home with gorgeous limestone floors and tiled bathroom featuring a generous living room with ample seating options and stunning views out of multiple sliding glass doors, a jacuzzi, balcony, and a propane fire pit. Located on 20 acres of desert landscape in Yucca Valley, California, USA, the property is nestled among weathered boulders, ancient junipers, and desert oaks."

Algorithm 1: Algorithm to align of Listings modalities

Input:

- CLIP text encoder T_{enc}
- CLIP photo encoder P_{enc}
- PhotoSet encoder PS_{enc}
- Embedding dimensions (e.g., 1024) d
- Number of photos per listing P
- Batch size B
- Dataset D

Output: PS_{enc}, T_{enc}

```

1 Initialize the CLIP encoders  $T_{enc}, P_{enc}$  from the pretrained
   checkpoints; Freeze  $I_{enc}$  (and  $T_{enc}$  if Coarse Alignment);
2 Initialize the PhotoSet encoder  $PS_{enc}$  weights with normal
   distribution;
3 foreach ( $photos, visual\_prof$ )  $\in D$  do
4    $photo\_ftrs \leftarrow P_{enc}.encode(photos)$ ; /*  $[B, P, d]$  */
5    $PS\_ftrs \leftarrow PS_{enc}.project(photo\_ftrs)$ ; /*  $[B, d]$  */
6    $text\_ftrs \leftarrow T_{enc}.encode(visual\_prof)$ ; /*  $[B, d]$  */
7    $logits \leftarrow PS\_ftrs \cdot text\_ftrs.T$ ; /*  $[B, B]$  */
8    $labels = np.arange(B)$ ; /*  $[B]$  */
9    $loss\_p = cross\_entropy(logits, labels, axis=0)$ ;
10   $loss\_t = cross\_entropy(logits, labels, axis=1)$ ;
11   $loss = \frac{loss\_p+loss\_t}{2}$ ;
12   $T_{enc}.backward()$ ;
13   $PS_{enc}.backward()$ ;
14 return  $T_{enc}, PS_{enc}$  ;

```

Source	Description
Title & Description	Host-provided title and description.
Location	Listing location as Airbnb's searches are very local
CLIP implicit signals	Visual tags using implicit feedback collected from internal users. Example: "A picture of a tiny house along the river"
Visually Detected Amenities	100+ amenities visually through object detection for small amenities (e.g., BBQ) and image classification (e.g., golf simulator).
Visual rating	A model generated rating indicating the visual appeal of the listing.
Structured data	Listing Category

Table 2: Inventory of signals going into a visual profile

3.4 Quantization

To accommodate multiple images per listings and still maintain a large batch size as proven empirically necessary [24], normalized image embeddings are encoded offline through an Optimal Product Quantization (OPQ) technique [6] and decoded as part of the training preprocessing.

Specifically, we use OPQ with settings of 256 centroids and a rotation matrix of output dimension of 1280. Each embedding is therefore reduced to 256 uint8, a 97% reduction in size at a loss of 9.9% of L2 mean error. Our experience shows that this loss is acceptable and keeps high fidelity to the original representation.

4 Experimental Results

In this section, we provide a detailed description of BiListing training procedure, offline evaluations we conducted, a summary of the results, and post training steps that were needed to adapt BiListing embeddings for application usages in listing search and ranking.

For easier reproduction, we described details of each steps we took, practical challenges we faced, and how we overcame them.

A dedicated section on BiListing embeddings path to production in listing search and ranking as features of the ranking ML model will follow afterwards.

4.1 Data Preparation

As noted in previous research [4, 20], data quality plays a significant role in modeling performance. We pay special attention to the data quality of the text-based visual profiles and apply multiple data processing and validation steps to our training and evaluation pipeline. The training and evaluation datasets exclude:

- Listings with less than 5 photos
- Listings with short visual profiles (<50 characters)
- A preliminary model trained on a large listing corpus is used to filter out low alignment listings. We use 0.3 as a threshold.

Finally, the raw visual text profile LLM outputs are lightly edited using predefined regular expressions. For instance, the number of bathrooms is removed, and "0 bedroom" is edited to "studio".

Together, the rules result in a reduction in 35% in the size of the dataset, now in the low tens of millions of listings.

4.2 Training Details

For the training, we extract the text encoder from a pretrained CLIP checkpoint (laion/CLIP-ViT-H-14-laion2B-s32B-b79K) and finetune it using the open_clip library [8] on 8XA100 for 15 hours, an accumulation size of 8, a weight decay of 0.2, a warmup of 10,000 steps and an individual batch size of 512 for a global batch size of 32k. The training uses Adam [13] coupled with a cosine scheduler. The model is trained using Ray TorchTrainer on top of the open_clip library.

The Photoset Transformer hyperparameters is tuned with the results below. The best model had 4 layers and 4 attention heads, and underwent a two-step training process:

- **Coarse Alignment:** Frozen text encoder for 140 epochs with a learning rate of $1e - 4$. The main purpose of this step is to pretrain the PhotoSet encoder aligning on the native CLIP representation
- **Fine Alignment:** The next 80 epochs had a learning rate of $2e - 5$ with the last 5 layers of the text encoder being unfrozen. The LayerNorm running stats in text tower for any locked layers were frozen.

4.3 Evaluation Results

4.3.1 Model Evaluation We set aside a random 1% of all listings as a holdout and evaluate performance metrics on this holdout. We evaluated the following metrics and present the results in Table 3 and Table 4:

- **Mean rank (text to image) (MR T->I),** the average rank position of the correct match in the retrieval set
- **Text to image recall (@ 1,5, 10),** the proportion of correct matches ranked in the top-K.
- **Image to text recall (@ 1,5, 10)**

Experiment	Loss	MR T->I (best)
Baseline		
MLP (baseline)	InfoNCE	26.5
PhotoSet Transformer		
<i>without coarse alignment</i>		
PS Trans (4 layers, 8 heads)	InfoNCE	20.2
PS Trans (8 layers, 8 heads)	InfoNCE	20.5
PS Trans (4 layers, 8 heads)	SigLIP [24]	8.43
<i>with coarse alignment</i>		
PS Trans (4 layers, 4 heads)	SigLIP	6.17

Table 3: Experiment results of different model architecture

The evaluation results reveal several useful improvements, in descending order of impact:

- Switching over to SigLIP as a loss function
- Introducing a coarse alignment step followed by a fine tuning step with lower learning rate
- Replacing the MLP baseline with the Photoset transformer

For the best model, we provide more metrics in Table 4.

Metric	Value
Mean rank, (text-> image)	6.2
Recall @ 1 (text-> image)	58%
Recall @ 5 (text-> image)	85%
Recall @ 10 (text-> image)	91%
Mean rank, (image -> text)	6.0

Table 4: Experiment results of the best model setting on a 10k randomly selected listing sample from the holdout

4.3.2 Model Comparison We perform a performance comparison against the in-production embeddings ("Legacy"), an updated version of Grbovic et al. [7] original work. To perform the comparison, we choose a wide range listing structured fields as targets and evaluate using k-nn probes trained on the training set. We also provide a zoom-in onto a sample of Airbnb Categories, Airbnb's structured property offering.

Attribute	BiListing (vision)	Legacy
Urban/Rural Density	0.64	0.41
Space Type (apt, ...)	0.62	0.46
Capacity (# people)	0.57	0.46
Bedroom count	0.55	0.49
Bathroom count	0.75	0.72
ADR percentile	0.29	0.27
Pro Host (Y/N)	0.655	0.74
Listing Tenure (days)	0.491	0.62
Host is superhost	0.53	0.68
Review count (bucketized)	0.37	0.64
Bookings count (bucketized)	0.26	0.56

Table 5: Listing structured attributes accuracy comparison

Attribute	BiListing (vision)	Legacy
A-Frame	0.87	0.80
Pool	0.88	0.81
Cabin	0.86	0.81
Lakehouse	0.87	0.81
Cave	0.87	0.83
Tropical	0.89	0.85
Beachfront	0.88	0.85

Table 6: Airbnb categories accuracy comparison

BiListing performs favorably on most attributes but not all. Unsurprisingly, BiListing performs best for tasks which most benefit from vision (e.g. Urban vs. Rural).

4.4 Ablation Study

To understand the role played by the different components of this complex workflow, we conduct two ablation studies, one on input quality and the other on output quality:

- **Quantization of CLIP embeddings:** The training procedure uses Optimal Product Quantization. We compare the approach against the more common Product Quantization and analyze the magnitude of the errors
- **Dimensionality Reduction of BiListing embeddings:** The integration into the Ranker uses a reduced number of dimensions. We present a study on how the number of dimensions affect the ranker's performance

4.4.1 Quantization of CLIP embeddings We evaluated multiple quantization strategies for the vision tower CLIP embeddings: Product Quantization (PQ) [12] and Optimal Product Quantization (OPQ) [6]. Table 7, indicates that the random rotation introduced in OPQ achieves a significant improvement in the compression quality, with over 54% error reduction at the P50.

	p_5	p_{25}	p_{50}	p_{75}	p_{90}	p_{99}
PQ	18.75	19.87	20.93	22.37	24.21	29.03
OPQ	8.30	8.89	9.49	10.36	11.64	16.46
L2 Error reduction (%)	55.73	55.27	54.68	53.68	51.92	43.31

Table 7: Distribution of loss following OPQ vs. PQ encoding

4.4.2 Dimensionality Reduction of BiListing embeddings Each tower produces an embedding of dimension 1024, too large to use in the production search ranking model as a feature due to various infrastructure requirements, latency requirements and scalability. To address this, we used Principal Component Analysis [11] followed by quantization.

For offline evaluation of the Airbnb's search ranking model, we use Normalized Discounted Cumulative Gain (NDCB) [9] with binary relevance score: whether the listing was booked. Figure 5 shows the relative offline NDCB gain, compared to the final model, when adding different dimensions of the BiListing embeddings as features. We ended up adding just the first 40 PCA dimensions to the production model as we observe diminishing returns with additional dimensions.

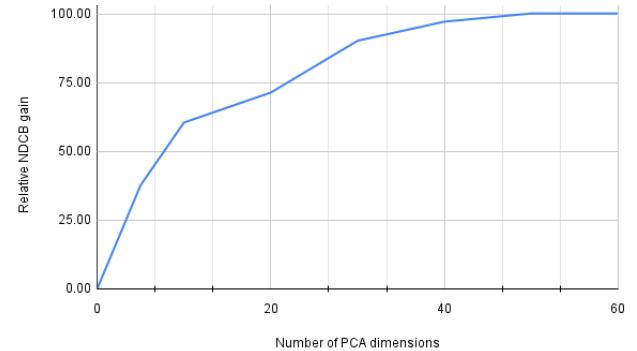


Figure 5: NDCB vs. top n PCA dimensions

Finally, we further investigate whether we can further compress the BiListing embedding through scalar quantization. No statistically significant differences were observed using just 8-bits precision with the full 32-bits float precision.

In conclusion, using the first 40 dimensions of 8-bit-encoded BiListing embeddings proves to be enough.

5 Application

5.1 Integration into Listing Search Ranking

Deployment Highlights

- **Production deployment:** BiListing provides the core listing representation of Airbnb’s core search ranking system.
- **Latency-aware design:** Embeddings are quantized (PCA + byte quantization) to 40 bytes per listing, supporting real-time scoring.
- **Cold start remediation:** Since BiListing uses photos only, its design is more robust to the cold start problem.
- **Integration:** Fully integrated into feature pipelines with automated daily embedding refresh.
- **Business impact:** +0.425% offline NDCG gain; A/B test shows statistically significant uplift in Uncancelled Bookers ($p = 0.025$), resulting in **tens of millions incremental revenue/year**.

We integrated BiListing into the core search ranking model [22, 23] powering Airbnb’s results page, addressing two main challenges: embedding freshness and compactness. A daily pipeline recomputes CLIP embeddings for new photos, applies OPQ quantization, and updates BiListing vectors.

As described in section 3, each listing is encoded into 1024-dimensional vision and text embeddings trained in a shared latent space. For ranking, we use only the vision embedding—cheaper to generate and independent of text availability—arranged as a sequence capturing visitors’ short- and long-term interactions with Airbnb inventory.

Following our quantization ablation (4.4.2), we reduced embeddings from 1024 to 256 dimensions via PCA, observed diminishing returns beyond the 40th component (Figure 5), and applied 8-bit scalar quantization. The final representation is just 40 bytes per listing.

This compact embedding delivered a +0.425% offline NDCG gain—above the +0.3% threshold [22] typically needed for statistically significant booking lift—and, in an online A/B test, increased Uncanceled Bookers ($p = 0.025$), yielding tens of millions in incremental revenue. By transitivity, these results also confirm additional gains over the listing embeddings in [7].

5.2 Internal Listing search

To help users find listings with certain attributes, we deploy an online application using Streamlit for the UI and FAISS [3], on Airbnb’s Sandcastle platform [16]. All active listings have their visual profile (text) and CLIP embeddings (photos) encoded into two sets of embeddings. A third embedding, denoted multimodal is inferred by mean-pooling the first two. Airbnb internal employees can query listings two ways:

- **Free text search.** For instance "A 2 bedroom apartment in La Rochelle with a view on the harbor"
- **Listing-to-Listing.** Using a listing embedding to perform a nearest-neighbor search.

For each query, the user can select which of the 3 modalities to use. We provide a representative sample of (top-1) results in Table 8 for a sample of free text queries:

Query	Image
A stunning treehouse in Japan	
a visually appealing Ryokan in Japan	

Table 8: Free-text search results in Airbnb’s inventory

Due to privacy concerns, we cannot provide listing-to-listing search result samples using a listing input query. However, the results show that the properties are indeed similar in many respects:

- Location (e.g. the Carribbeans)
- Property Type (e.g Villa)
- top amenities (e.g. pool, view of the beach)

6 Conclusion

We proposed BiListing embeddings, extending the standard modality alignment approach through contrastive learning to a more complex entity - Airbnb’s listings. We demonstrate the usefulness of the BiListing embedding in understanding the Airbnb’s listings better through two downstream applications: Airbnb’s search ranking model and as a standalone free text search application for internal Airbnb employees. Our special emphasis was on explaining our training in detail such that it can be easily reproduced by other practitioners who we believe will find it useful for their applications.

We would also like to call out that the illustrative photos in this paper had to comply with Airbnb’s privacy policy and had to be changed. We believe that this change does not change the character of the findings.

7 Future Work

While BiListing demonstrates measurable gains in production quality for complex multi-image entities such as Airbnb’s listings, several avenues remain open for exploration. First, evaluating how techniques for context extension such as Long-CLIP [25] or Tulip [17] could further enhance the model. Second, expanding ablation studies on the choice of large language model for Visual Text Profiles and the role of positional encoding in the PhotoSet Transformer would strengthen the understanding of design trade-offs. Finally, deeper analysis of cold-start performance and integration strategies within the ranking system may yield additional performance gains in practical deployments.

8 AI Usage Disclosure

AI tools were used to assist with grammar checking and minor rephrasing. The authors remain fully responsible for the accuracy, originality, and integrity of all content presented in this work.

References

- [1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. 2022. Flamingo: a Visual Language Model for Few-Shot Learning. arXiv:2204.14198 [cs.CV] <https://arxiv.org/abs/2204.14198>
- [2] Riccardo Albertoni, Sara Colantonio, Piotr Skrzypczyński, and Jerzy Stefanowski. 2023. Reproducibility of machine learning: Terminology, recommendations and open issues. *arXiv preprint arXiv:2302.12691* (2023).
- [3] Matthijs Douze, Alexandre Guizha, Chengqi Deng, Jeff Johnson, Gergely Szilvassy, Pierre-Emmanuel Mazaré, Maria Lomeh, Lucas Hosseini, and Hervé Jégou. 2024. The Faiss library. arXiv:2401.08281 [cs.LG] <https://arxiv.org/abs/2401.08281>
- [4] Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander Toshev, and Vaishal Shankar. 2023. Data Filtering Networks. arXiv:2309.17425 [cs.AI] <https://arxiv.org/abs/2309.17425>
- [5] Manuel Fayolle, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. 2024. ColPali: Efficient Document Retrieval with Vision Language Models. arXiv:2407.01449 [cs.IR] <https://arxiv.org/abs/2407.01449>
- [6] Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. 2013. Optimized Product Quantization for Approximate Nearest Neighbor Search. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*. 2946–2953. <https://doi.org/10.1109/CVPR.2013.379>
- [7] Mihajlo Grbovic. 2018. *Listing Embeddings in Search Ranking*. Retrieved November 27, 2024 from <https://medium.com/airbnb-engineering/listing-embeddings-for-similar-listing-recommendations-and-real-time-personalization-in-search-601172f7603e>
- [8] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. 2021. OpenCLIP. <https://doi.org/10.5281/zendodo.5143773> If you use this software, please cite it as below.
- [9] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.* 20, 4 (Oct. 2002), 422–446. <https://doi.org/10.1145/582415.582418>
- [10] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lampe, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of Experts. arXiv:2401.04088 [cs.LG] <https://arxiv.org/abs/2401.04088>
- [11] I. T. Jolliffe. 2002. *Principal Component Analysis*. Springer-Verlag, New York. <https://doi.org/10.1007/b98835>
- [12] Herve Jégou, Matthijs Douze, and Cordelia Schmid. 2011. Product Quantization for Nearest Neighbor Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 1 (2011), 117–128. <https://doi.org/10.1109/TPAMI.2010.57>
- [13] Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG] <https://arxiv.org/abs/1412.6980>
- [14] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. arXiv:2201.12086 [cs.CV] <https://arxiv.org/abs/2201.12086>
- [15] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. 2021. CLIP4Clip: An Empirical Study of CLIP for End to End Video Clip Retrieval. arXiv:2104.08860 [cs.CV] <https://arxiv.org/abs/2104.08860>
- [16] Daniel Miller. 2024. *Sandcastle: data/AI apps for everyone*. Retrieved September 24, 2024 from <https://medium.com/airbnb-engineering/sandcastle-data-ai-apps-for-everyone-439f3b78b223>
- [17] Ivona Najdenkoska, Mohammad Mahdi Derakhshani, Yuki M. Asano, Nanne van Noord, Marcel Worring, and Cees G. M. Snoek. 2025. TULIP: Token-length Upgraded CLIP. arXiv:2410.10034 [cs.CV] <https://arxiv.org/abs/2410.10034>
- [18] Jesús Andrés Portillo-Quintero, José Carlos Ortiz-Bayliss, and Hugo Terashima-Marin. 2021. A Straightforward Framework For Video Retrieval Using CLIP. arXiv:2102.12443 [cs.CV] <https://arxiv.org/abs/2102.12443>
- [19] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. arXiv:2103.00020 [cs.CV] <https://arxiv.org/abs/2103.00020>
- [20] Alex Fang Samir Yitzhak Gadre, Gabriel Ilharco. 2023. DataComp: In search of the next generation of multimodal datasets. *arXiv preprint arXiv:2304.14108* (2023).
- [21] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. 2022. LAION-5B: An open large-scale dataset for training next generation image-text models. arXiv:2210.08402 [cs.CV] <https://arxiv.org/abs/2210.08402>
- [22] Chun How Tan, Austin Chan, Malay Haldar, Jie Tang, Xin Liu, Mustafa Abdool, Huiji Gao, Liwei He, and Sanjeev Katariya. 2023. Optimizing Airbnb Search Journey with Multi-task Learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23)*. ACM, 4872–4881. <https://doi.org/10.1145/3580305.3599881>
- [23] Jie Tang, Huiji Gao, Liwei He, and Sanjeev Katariya. 2024. Multi-objective Learning to Rank by Model Distillation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24)*. ACM, 5783–5792. <https://doi.org/10.1145/3637528.3671597>
- [24] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid Loss for Language Image Pre-Training. arXiv:2303.15343 [cs.CV] <https://arxiv.org/abs/2303.15343>
- [25] Beichen Zhang, Pan Zhang, Xiaoyi Dong, Yuhang Zang, and Jiaqi Wang. 2024. Long-CLIP: Unlocking the Long-Text Capability of CLIP. arXiv:2403.15378 [cs.CV] <https://arxiv.org/abs/2403.15378>